DECEMBER 1986

LIDS-R-1624

PROCEEDINGS OF THE

9TH MIT/ONR WORKSHOP ON C³ SYSTEMS

HELD AT

NAVAL POSTGRADUATE SCHOOL
AND
HILTON INN RESORT HOTEL
MONTEREY, CALIFORNIA
JUNE 2 THROUGH JUNE 5, 1986

EDITED BY

MICHAEL ATHANS
ALEXANDER H. LEVIS

SPONSORED BY

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
LABORATORY FOR INFORMATION AND DECISION SYSTEMS
CAMBRIDGE, MASSACHUSETTS

WITH SUPPORT FROM

OFFICE OF NAVAL RESEARCH
CONTRACT ONR/N00014-77-C-0532(NRO41-519)

AND IN COOPERATION WITH
IEEE CONTROL SYSTEMS SOCIETY
TECHNICAL COMMITTEE ON C³

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE *(and Subtitle)* 9th MIT/ONR WORKSHOP ON C^3 SYSTEMS		5. TYPE OF REPORT & PERIOD COVERED INTERIM 6. PERFORMING ORG. REPORT NUMBER LIDS-R-1624
7. AUTHOR(s) Michael Athans Alexander H. Levis		8. CONTRACT OR GRANT NUMBER(s) ONR-N00014-77-C-0532
9. PERFORMING ORGANIZATION NAME AND ADDRESS Laboratory for Information and Decision Systems Massachusetts Institute of Technology Cambridge, Massachusetts 02139		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS NR-041-519
11. CONTROLLING OFFICE NAME AND ADDRESS Office of Naval Research Arlington, Virginia 22217-5000		12. REPORT DATE December 1986 13. NUMBER OF PAGES x + 259
14. MONITORING AGENCY NAME & ADDRESS *(if different from Controlling Office)*		15. SECURITY CLASS. *(of this report)* UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

Approved for public release, Distribution unlimited

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

COMMAND, CONTROL, COMMUNICATIONS

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

This report contains printed manuscripts of papers presented at the Workshop by several authors.

FOREWORD

The Ninth MIT/ONR Workshop on C3 systems was held from June 2 to 5, 1986 at the Naval Postgraduate School and the Hilton Hotel in Monterey, California. These Proceedings constitute the written record of the research presented at the Workshop.

Attendance at the workshop this year increased to 106 registrants, excluding students from the Naval Postgraduate School. A list of the registrants and their affiliations can be found at the end of this volume. A total of 71 papers were presented at the Workshop, a large increase from prior years. This increase in papers necessitated the conduct of parallel competing sessions every afternoon. In spite of the large number of papers, there was ample time for discussion of the research results being presented and to squeeze in the presentations of some latecomers. However, only 39 papers were submitted for publication in these Proceedings; once more some authors experienced difficulties in obtaining the necessary approvals for submitting a written version of their presentations.

This workshop is the last one organized by MIT on behalf of the Office of Naval Research. When the MIT/ONR workshops started in 1977, there was little cohesiveness in basic research directions in the theory of Command and Control and related disciplines. The MIT/ONR workshop became a key and unique forum for exchanging ideas, for presenting unclassified research findings, and in relating the theoretical state-of-the-art to the pressing problems faced by the military. Its objective was to foster interdisciplinary research, to allow researchers from different disciplines to learn about Command and Control, and to participate in basic research. At present, basic research in C3 systems is thriving, and we are seeing the emergence of several focused and applied R&D programs whose origins can be traced to the concepts presented in earlier workshops. As the applications of existing theory are increasing, we are uncovering even more challenging unsolved basic research problems in this fascinating area of multidisciplinary research. More important, there is a solid "core group" of several government, industrial, and academic researchers who consider C2 theory as a vibrant and challenging area for future investigations.

For these reasons, we felt that the time had come for the C3 community to organize a different forum for exchanging research ideas and arrive at novel ideas for paper presentation and publication. In short, we felt that after nine years, the MIT/ONR C3 Workshop had fully met its original objectives and that a change in organization and format was a desired and healthy evolution.

In this spirit, it is our present understanding that a new C3 Conference will take place at the National Defense University in Washington, DC in June 1987. The C3 Conference is sponsored by the Joint Directors of Laboratories with help from the Naval Postgraduate School.

The editors sincerely thank the authors and the participants for their contributions to the 1986 C3 Workshop. Special thanks are due to Mr. J. Randolph Simpson of ONR for his help and support; to Professor Michael Sovereign of the Naval Postgraduate School for his help in hosting the workshop; and to Ms. Lisa M. Babine of MIT for her superior handling of the administrative aspects of the Workshop and these Proceedings. The workshop organizers benefited from the support of ONR contract N00014-77-C-0532 (NR 041-519). Finally, we wish the sponsors of the new C3 Conference success in their important undertaking.

Michael Athans
Alexander H. Levis

Massachusetts Institute of Technology
Cambridge, Massachusetts
November 3, 1986

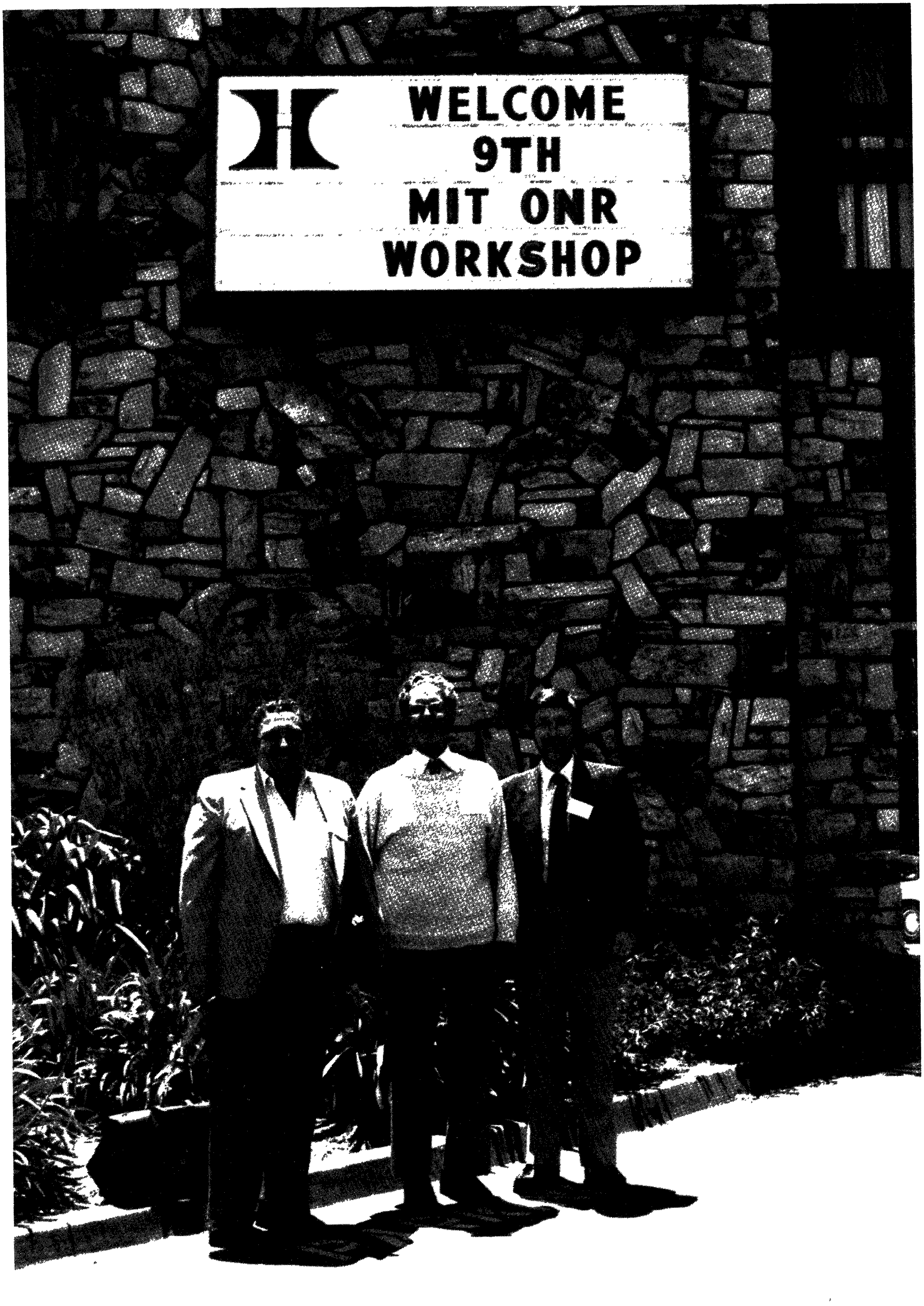

WELCOME
9TH
MIT ONR
WORKSHOP

TABLE OF CONTENTS

COMMAND-AND-CONTROL (C2) THEORY:
A CHALLENGE TO CONTROL SCIENCE.

by

Michael Athans
Professor of Systems Science and Engineering
Department of Electrical Engineering and Computer Science
Laboratory for Information and Decision Systems
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
Cambridge, MA 02139

and

Chairman
ALPHATECH INC.
111 Middlesex Turnpike
Burlington, MA 01803

0. SUMMARY

The basic premise of this position paper is that the field of military Command-and- Control (C2) systems offers challenging basic research opportunities to researchers in the control sciences and systems engineering disciplines. In point of fact, the analysis and design of complex, survivable, and responsive C2 systems requires novel advances in the area of *distributed dynamic decision-making under uncertainty.* Advances are also needed in systems engineering tools for describing, decomposing, and analyzing such systems. As a consequence, control scientists and engineers are uniquely qualified to extend their technologies to meet the multidisciplinary challenges posed by C2 systems and to advance the state of the art in the development of a relevant C2 theory.

The author strongly believes that the methodological, theoretical, algorithmic, and architectural questions which arise in the context of military C2 systems are generic and quite similar to those needed to improve the reliable performance of many other civilian C2 systems, such as air traffic control, automated transportation systems, manufacturing systems, nuclear reactor complexes etc. All such military and civilian C2 systems are characterized by a high degree of complexity, a generic distribution of the decision-making process among several decision-making "agents", the need for reliable operation in the presence of multiple failures, and the inevitable interaction of humans with computer-based decision support systems and decision aids; also, they require the development of novel organizational forms and system architectures which provide for the harmonious interface of the mission objectives associated with the C2 process and the physical hardware, such as sensors, communications devices, computer hardware and software, and effectors -- weapons or machines -- which implement the overall Command, Control, and Communications (C3) system whose purpose is to support the global C2 decision process.

Military C2 systems provide one particular focus for the development of a whole new class of control/estimation/decision technologies - technologies which share the intellectual roots of current research in the control sciences, but which can grow and blossom into methods applicable to a very large variety of civilian complex systems. In addition, they exemplify the kind of growing complexity that systems scientists and engineers must continoually face and find ways of managing.

In the remainder of this paper we shall concentrate upon military C2 processes and C3 systems, since they provide the most stringent performance requirements and because they exhibit the greatest clear-cut need for quantification of their measures of performance (MOP's) and measures of effectiveness (MOE's), and the requirement for novel distributed architectures and organizational forms. The discussion will undoubtedly reflect the personal bias of the author who has studied and researched military C3 systems over the past decade, in the sense that the objectives of a military C2 system are easier to pin down, and the need for survivable/reliable operation with minimal communications is transparent. However, it is the strong personal conviction of the author that any technological advances in the state of the art in military C2 systems are readily transferable to civilian C2 systems.

1. TWO MILITARY C2 SYSTEMS

1.1 Introduction.

In this section we overview two different Battle Management C3 (BM/C3)systems. One relates to the defense of naval Battle Groups, while the second addresses issues related to the Strategic Defense Initiative. The former involves a significant component of tactical human decision making, while the second is envisioned to act in an automatic tactical mode. The author has studied both of these in some detail. Many other BM/C3 systems involving Army, Air Force, and Marine operations involve similar issues. Our objective is to set the stage for the types of issues which are important in C2 systems, so that later on we can isolate certain generic questions common to them. These in turn will define the broad opportunities in which basic research in the control sciences and system theory can extend its applicability.

1.2 The Defense of Naval Battle Groups.

A naval Battle Group (BG) is defined as consisting of at least one carrier (CV) together with several escorting platforms (ships, submarines, and aircraft). The CV's and their platforms contain a wide variety of sensors and weapon systems which allow the BG to carry out defensive and offensive missions as prescribed by higher authority.

The defense of the BG assets is clearly of paramount importance. The threat to the BG is multiwarfare in character. The BG threat consists of enemy submarines, which can launch long range missiles and/or short range torpedoes, surface ships which can launch missiles and/or cannon projectiles, and aircraft that also launch missiles and/or conventional bombs. As a consequence, the defense of the BG involves antisubmarine warfare (ASW), antisurface warfare (ASUW), and antiair warfare (AAW); electronic warfare (EW) permeates the BG operations as well. The enemy platforms must be detected using information from organic BG sensors perhaps "fused" together with information gathered by other national assets; they must also be identified, tracked and engaged (hopefully) before they launch their offensive weapons against the BG assets.

The BG defense involves several layers. Obviously, enemy submarines, ships, and aircraft must be engaged before they can launch their long range missiles; this is often called the "outer battle". "Area defense" against aircraft and missiles is provided by missile-shooting platforms (the newest one being the Aegis class cruisers). "Terminal defense" involves individual platform weapons, such as rapid-fire guns and/or short-range missiles, and different countermeasures (jamming, decoys etc) that are designed to confuse incoming weapons.

The vulnerability of the BG platforms to enemy weapons, especially nuclear ones, forces wide geographical dispersal of its platforms. Also, long-range submarine detection requires certain platforms to operate at the fringes of the BG formation. Thus, it is not unusal for a multicarrier BG to have its platforms spread-out over hundreds of miles. The large geographical dispersal of the platforms makes it difficult to communicate with each other, using line-of-sight communication frequencies, while escaping enemy detection of the communication signals that help localize the BG location and denying the enemy the detection of certain unique electromagnetic emissions that may reveal the identity of certain platforms. Hence, survivability considerations must be traded-off with the need to communicate so as to coordinate the BG defensive operations.

At present the U.S. Navy also operates under a *distributed* C2 doctrine, the so-called Composite Warfare Commander (CWC) doctrine which reflects the complexities of Naval warfare and survivability. The senior admiral in charge of the BG (CWC), under the CWC doctrine, can delegate command and control authority and responsibility to three senior warfare commanders: the ASW commander (ASWC), the ASUW commander(ASUWC), and the AAW commander (AAWC) who are specialists in their respective warfare areas. It is not unusal for these subordinate commanders to be located in different platforms so as to improve survivability and to have direct access to unique sensor data and/or weapon systems. The CWC assigns control of specific platforms (submarines, ships, aircraft, helicopters etc) to each subordinate warfare commander, a resource-allocation problem, so that each one can defend the BG assets from the specific threat in his assigned domain.

Although the CWC doctrine appears to be reasonable at first glance, it requires intensive coordination, and hence reliable communications, among the CWC and his warfare commanders due to several reasons. The first reason is that an enemy submarine (or surface ship) that has survived prosecution by the ASWC (or the ASUWC) will launch its missiles and these missiles become the AAWC's problem. Thus, the AAWC must position his assets in such a way so as to be able to engage surviving submarine and surface ship launched missiles. The second reason relates to the fact that most naval platforms have sensor and weapon resources that are useful in several warfare areas; thus, a destroyer under the control of the ASWC may still be a very valuable platform for the AAWC. The third reason relates to BG electronic warfare (EW); the assets for EW are spread among most platforms, and the superior coordination of the EW assets, at the global BG level, remains an unsolved problem. The Navy, aware of this problem, has assigned an EW coordinator - not a commander - to advise the CWC in EW related matters.

1.3 Battle Management C3 in the Strategic Defense Initiative.

The Strategic Defense Initiative (SDI) offers extraordinary challenges in the Battle Management C3 (BM/C3) area. Long-term SDI system architectures envision a multilayered defense system against ICBM's and SLBM's. Potentially enemy weapons are engaged in the boost, post-boost, early midcourse, late midcourse, high endoatmospheric, and low endoatmospheric phases by a variety of orbiting and ground-based weapon systems. Different sensors reside in diverse satellites in different orbits, as well as in airborne and ground-based nodes. Orbiting weapons may include X-ray lasers, chemical lasers, fighting mirrors to direct ground-based free electron lasers, electromagnetic launched weapons, orbiting kinetic-kill vehicles etc. Ground based weapons may include free electron lasers, and kinetic-kill vehicles such as long-range and short-range missiles.

Clearly the direction of the weapon systems must rely upon the BM/C3 functions of detection, tracking, discrimination (i.e. weapon or decoy?) and damage assesment information provided by both orbiting and ground-based sensor systems. This multi-sensor information must be fused and mapped into the weapon-to-target assignment and engagement control functions. The distribution of the BM/C3 decision processes is dictated not only by orbital mechanics, but even more by severe survivability requirements, so that the SDI system can survive significant enemy attacks by ASAT weapons.

Leaving socio-political considerations aside, the SDI has been criticized in terms of the feasibility of its huge BM/C3 software requirements, since the tactical system will have to operate in an automated mode simply because there is no time for humans to evaluate the huge amounts of sensor information and to arrive at superior weapon engagement strategies in the short time available (about 30 minutes). The critics (many of whom are computer scientists) are addressing in the author's opinion the wrong problem. The challenge is rather a control-theoretic one: *how to properly design the distributed algorithms that implement the diverse BM/C3 functions, so that a prescribed degree of reliability and survivability is maintained.* Although we do not have all the theories as yet, it is the author's belief that many available results in large-scale estimation, optimization, and control are directly relevant and applicable to the SDI BM/C3 problem. On the basis of available results one could argue that, given sufficient research, control theorists and engineers can develop the required survivable and reliable distributed architectures and algorithms which will implement the SDI BM/C3 estimation, optimization, resource allocation, and control algorithms.

2. SOME GENERIC ISSUES IN MILITARY C2 SYSTEMS.

2.1 Introduction.

Altough each military BM/C3 problem has its own unique set of mission requirements and physical assets, nonetheless all C2 systems have a great degree of commonality. *It is precisely this generic commonality that offers the hope that the development of a relevant C2 theory will have a significant impact upon the analysis and design of military C2 systems.* A little thought should convince the reader that the command-and-control of several complex civilian systems also involves similar generic issues.

In this section we discuss what are the major high-level problems in military C2 systems. We focus, in particular, to issues related to organizational forms and distributed decision architectures. These are precisely the areas that offer the most fertile ground for basic and applied research by control scientists and engineers; these will motivate the more detailed listing of relevant interdisciplinary basic research areas in the sequel. We remark that any analysis tools that help quantify the expected performance of existing C2 organizations, as well as of synthesis methodologies that help in designing new superior BM/C3 architectures are desparately needed.

2.2 The Impact of Geography.

A military C2 system is a multi-agent organization. The decision agents are both human decision-makers and computer-based algorithms. The decision agents are geographically dispersed due to environmental and survivability reasons. Geographical dispersion is dictated by the environment, the nature of sensors, and the physics and speed of the weapons. Thus, both geography and vulnerability contribute to the distributed architecture of C2 organizations. Such geographically motivated decompositions define, for example, the multiple defense tiers in the BG defense and in the SDI scenarios. Each defense tier can be further decomposed into sectors, although protocols for hand-over coordination and need for low-level communications present thorny issues.

Geographical distances interact with the speed of the weapons, the range of the sensors, and the tempo of the military operations in the definition of defense tiers, defense sectors etc. It is important to realize that any technological developments that impact sensor ranges, weapons speeds, etc must be reflected into a reorganization of the C2 system in order to maintain superior performance. This may necessitate doctrinal revisions as well as changes in the architecture of the BM/C3 system.

2.3 Functional Decompositions and Distributed BM/C3 Architectures.

Another key element that contributes to the way the C2 process is organized has little to do with geography. The C2 process can be decomposed into a set of generally accepted *C2 functions* that must be executed (sometimes serially and sometimes in parallel and, in general, in an asynchronous manner) to ensure mission success. This list of functions related to defensive Battle Managment C3 is as follows:

(1).*Threat Detection*, based on data from several sensors.

(2).*Target Tracking*, based on data from several sensors. This function may involve 2-dimensional tracking by individual sensors and fusion into 3-dimensional tracks. Sensor cueing, scheduling and control is an integral part of this function.

(3).*Discrimination*, which results in the resolution of true threats from decoys often requiring the fusion of data from several (active or passive) sensors. Sensor cueing, scheduling and control is also an integral part of this function.

(4).*Identification*, the process by which further identity information of threats is established.

(5).*Battle Planning*, the process by which decisions are made on how to deal with the identified threat, based on (1) to (4) above, including contigency planning.

(6).*Weapon-to-Target Assignment*, the set of decisions which lead to the assignment of one or more weapons to engage each threat, including the assignment of any necessary sensor, communication, and other resources required for each and every one-on-one engagement.

(7).*Engagement Control*, the process by which the decisions in (5) and (6) are executed in real time.

(8).*Damage Assessment*, the process by which one identifies and/or verifies the outcome of the engagement, i.e. whether a particular target has been killed.

The above list of BM/C3 functions have to be executed at a global level, at a defense tier level, at a sector level etc. The so-called *BM/C3 architecture* reflects how these functions are implemented by the sensor, computer, communications, and weapon hardware and where the *BM/C3 algorithms*, which realize these functions and are executed by human commanders and/or computers, are located. It is obvious that the vulnerability of the humans and hardware that implement the BM/C3 functions, i.e. *the vulnerability of each and every BM/C3 function*, is a very strong driver to the physical distribution of the decision agents; this leads to the problem of first analyzing candidate *distributed BM/C3 architectures* and later on the design of C2 organizations which implement the distributed BM/C3 architectures in a superior manner. Ideally, the survivability of each function to enemy attacks and to environmental phenomena calls for some redundancy; exact replication should be avoided if at all possible.

2.4 The Impact of Complexity.

The decomposition of the C2 decision processes is also influenced by the complexity of the warfare problem. This is, in general, true when simultaneous engagements involving heterogeneous sensor and/or weapon systems can take place, and human commanders make a large part of the decisions. For example, the BM/C3 decision process for the defense of a Battle Group falls in this category. Different commanders are trained to be "specialists" in different warfare areas, although they may have to share, and compete for, many common resources. No commander alone can deal with the inherent complexity of the global engagement; this leads to a decomposition of the decision process along distinct "expertise" dimensions.

In such C2 organizations team training is essential so as to achieve superior coordination and to make best utilization of scarce common resources. Indeed, it has been observed that in well-trained teams the decisions of individual commanders are different than those that the same commander would make if he were to operate in isolation (see Sections 3.4 and 3.5 for additional discussion).

At an abstract level, one can model the decomposition of the C2 process along specialist dimensions as yet an alternative way of decomposing the generic BM/C3 functions.

2.5 Discussion.

At present, all analysis and synthesis studies related to distributed BM/C3 architectures are carried out in an ad-hoc manner; it is self evident that the development of quantitative methodologies, theories, and algorithms relevant to the distributed BM/C3 architecture problem would be welcomed by the defense community. *It is interesting to note here that the C2 community does not, in general, appreciate the intimate relationship of distributed decision-making algorithms that execute the BM/C3 functions, their tactical communications requirements, and their intimate relation to distributed BM/C3 architectures.*

It is generally acknowledged that *centralized* BM/C3 hierarchical architectures are very vulnerable, introduce possibly unacceptable time-delays, yet are efficient in resource-utilization. At the opposite extreme, it is also realized that *autonomous* architectures (those that operate in a purely decentralized mode with no tactical coordination whatsoever) are more survivable, require minimal time-delays, but are most inefficient in the use of scarce resources. Obviously, *distributed BM/C3 architectures* are the answer, somewhere between centralized and autonomous ones. The difficulty is that there are an infinite number of ways that one can think of designing distributed BM/C3 architectures, and no general guidelines are available on how to even get started!

In short, superior BM/C3 architectures must be distributed in both a geographical and a functional sense, taking advantage in an integrated manner of the impact of geography, functional decomposition, mission objectives, problem complexity and the survivability of the BM/C3 functions. Current C2 technology approaches all of the above problems in a completely intuitive and qualitative way. As a consequence, there does not exist even a systematic methodology that can be used to understand in a precise manner the complex cause-and-effect relationships inherent in a C2 process and to describe them using a minimal set of primitives, measures of performance, and measures of effectiveness. Clearly control scientists and engineers can have an impact in this key area.

There is no fundamental reason whatsoever which inhibits the emergence of a quantitative methodology that addresses, in a relevant manner, the challenging BM/C3 architectural problems. Indeed, one can argue that the strong relationship of the nature of the distributed algorithms that implement the BM/C3 functions, which make strong use of control-theoretic concepts (variants of hypothesis testing, Kalman filtering, mathematical programming, stochastic optimization, etc), and of distributed BM/C3 architectures provides a natural starting point in the quest for a C2 theory. *Engineers and scientists trained in control theory and operations research, and related normative disciplines, are uniquely qualified to develop the needed basic research for distributed BM/C3 systems.*

3. A BASIC RESEARCH AGENDA.

3.1 Introduction.

In this section we outline some basic research directions that appear to be relevant in the quest for a C2 theory. Needless to say, there is no claim that the suggested research directions are all inclusive. However, it should become self-evident that these research directions are in the spirit of the evolving research traditionally associated with control science and engineering. The control science field broadened its research horizons into decision-oriented problems more than fifteen years ago when we started studying "large-scale systems". What we call C2 theory requires advances in control/estimation/decision technologies along particular dimensions to support our basic understanding of the BM/C3 processes and their reliable and effective implementations.

3.2 Understanding a Complex C2 System.

Before we can even analyze, never mind design, a C2 system we must first understand it. In order to understand it, a common representation language and a hierarchy of models must be developed which are useful in the sense that the key variables, transformations of these variables, and measures of performance become transparent.

Such C2 representation tools are not currently available. Block and functional-flow diagrams are used to indicate interconnection of physical devices, but these are not sufficient to capture the information flow, the sequence of events, the essential precedence relationships, and the time delays that are so crucial.

Some very recent attempts, which show some promise, are based on extensions of the *Petri Net* methodology originally developed to model digital computer operations. The extension of the Petri Net methodology to model BM/C3 systems requires the assignment of attributes to the Petri Net tokens, stochastic decision rules, time delays, and nonconcurrent events. Such extended Petri Net methodologies appear useful because they can help isolate the truly independent variables (analogous to a minimal state-space realization), keep track of the information and physical variables that must be present before a particular decision can be executed, account for stochastic time-delays associated with the implementation of the decision process, and capture probabilistic outcomes of the decisions. Such extended Petri Net methodologies also blend well with finite-state representations. Further, they can be used to study the C2 process in terms of its basic generic BM/C3 functions (see Section 2.3), allowing for a certain freedom to the C2 analyst in controlling the level of aggregation and the degree of detail appropriate for the questions posed. They also resemble the discrete-event dynamic models being used to describe manufacturing systems.

3.3 Modeling C2 Systems.

At a very detailed level the state variables underlying any C2 system are both continuous and discrete; hence, so-called *hybrid* state-spaces must be studied. The dynamic evolution of the state variables can be modeled in discrete-time; however, there does not exist a fixed time interval (such as sampling time) that governs the evolution of the state variables. Rather, we deal with *event-driven* dynamical systems and state-variable transitions occur at stochastic times that are, in general, asynchronous. Hence, modeling methodologies that are "tied" to a time- synchronization model are not apt to be either relevant or useful. Therefore, *problem-driven research related to hybrid, event-driven, asynchronous stochastic dynamic systems is of interest, especially when the state transition probabilities also depend on the values of not only certain exogenous variables, but also on a subset of the state variables.*

The difficulty with a hybrid state approach to modeling complex C2 systems is the huge dimension of the underlying state space. Although such fine detail may be necessary to construct an event-driven microscopic Monte-Carlo simulation (which may require well over 100,000 lines of FORTRAN code), such large-scale microscopic simulations (several of which have been constructed for specific military problems over the years) are time-consuming, expensive, and not well suited for analysis, design, and evaluation of alternate distributed BM/C3 architectures.Also, "what if" questions are costly to answer using these huge simulations. Indeed, a major shortcoming of many of the existing large scale simulation models is that the C2 process is not modeled in a way that tradeoffs associated with different BM/C3 architectures can be carried out. *This brings up the research need for systematic aggregation methodologies that result in higher level models that, hopefully, approximate the microscopic interactions and are more suitable and amenable to analysis and design.*

Few aggregated models exist, and even these have some significant limitations. The so-called *Lanchester Equations* of combat, a set of nonlinear differential equations that model mutual attrition of opposing forces using different types of weapons, have been widely used by the military community. However, it is very difficult to incorporate in the Lanchester-type equations the impact of different distributed C2 organizational forms. *The development of high-level models that capture not only attrition, but explicitly incorporate decision variables that relate the impact of alternative C2 organizations would be highly desirable, because one could then, in principle, analyze, synthesize, and optimize the BM/C3 architectures.* Ideally, these aggregate models should have their roots in the microscopic hybrid-state models so that their predictions can be checked by detailed Monte-Carlo simulations. It would be very useful to develop aggregation methodologies for this class of systems, with transparent advantages and shortcomings.

Another set of important modeling-oriented questions relate to the evaluation of aggregate measures of performance (MOP's) and measures of effectiveness (MOE's). In optimal control jargon, MOP's involve functions of output variables that have a specific meaning and are important to a military decision maker; think of combinations of different MOP's as defining the integrand of the cost-functional in a dynamic optimal control problem. Similarly, think of MOE's as corresponding to a particular cost functional which integrates over time a weighted combination of the MOP's. *One of the most important MOP's in any C2 system relates to the time delays associated with the execution of the generic BM/C3 functions,* such as detection, tracking, discrimination etc (see Section 2.3). The reason is that the performance of a BM/C3 system is like a race against time between the moving physical entities (targets, sensors, weapons) and the information variables. In a well designed BM/C3 system the information variables must win the race; the detection function must be completed before either the tracking and/or the discrimination functions can commence, and targets must have been sorted, identified, and tagged before we can wisely commit weapons against them. Delays in execution of any of these functions may degrade the kill probability, result in inefficient use of battle space, cause weapons to be assigned to decoys rather than threatening targets and/or assign too many weapons against the same target, and perhaps allow many targets to leak through a particular defense zone.

There exist significant and challenging opportunities in developing large-scale models that quantify delays for a given BM/C3 architectures; the availability of these models would allow the C2 analyst to pinpoint bottlenecks which would point the way for modification of the BM/C3 architecture. Delays arise from a wide variety of phenomena, e.g. signal processing of sensor data, other computational delays, communications delays in fusing information, and decision delays associated with human or algorithmic decision-making. It appears that significant extensions to the available theory associated with queueing networks are necessary in order to faithfully describe the elemental and global time-delays associated with a particular BM/C3 architecture.

To appreciate the relevance of queueing theory think of a target as being a "customer" in a service queue; a C2 node must service the target in the sense of performing a BM/C3 function (e.g. detection, tracking, engagement, etc). In a proactive BM/C3 system a specific C2 node may be assigned to perform the appropriate function. Since the target is moving, there is only a finite time-window of opportunity to service this target; otherwise, the target will leave (leak) that particular C2 node. Thus, we have to deal with *queues with reneging.* Although some theoretical results are available in this class of queueing network problems, additional research is required to arrive at an expanded set of theoretical results, together with efficient computational algorithms, to faithfully model the delays in a BM/C3 system.

Another important research area deals with the development of efficient computational algorithms that capture transient effects in queueing networks. In most military scenarios transients are very important. At present, such transient phenomena can only be handled by microscopic simulations, and these are difficult to interface with classical queueing network models. Any theoretical developments that help simplify the interface of static and transient delay models are very relevant and useful. If we develop theories and algorithms that allow the C2 analyst to evaluate easily both steady-state and transient delays, then one would also be able to use such queueing network models to study the vulnerability of the BM/C3 system to enemy countermeasures (jamming, node destruction) at least from a delay viewpoint.

3.4 Modeling Human Decision Makers in C2 Organizations.

In present C2 systems, almost all of the BM/C3 functions discussed in Section 2.3 are executed by trained human commanders; there are very few computer-based decision aids in use today. Since a C2 system involves the integration of humans with physical assets (sensors, communication links, weapons, etc), it is self evident that in order to analyze the performance of a C2 system one needs some high-level mathematical models that abstract the decision-making process of trained military commanders. In the absence of such models one can only rely upon very very expensive field exercises and war games; these are valuable and necessary, but their cost precludes the answering of too many "what if" questions. In particular, *current military expertise does not necessarily carry over without significant training to situations in which technological advances yield new sensor and/or weapon systems.* To put it another way, technological breakthroughs in sensors and/or weapons may require a drastic reorganization of the BM/C3 architecture in order to realize the benefits of these "high tech" hardware. It is not obvious that even a top-notch commander, trained under an older doctrine and within a different C2 organization,will perform at his best, say, in a war game that incorporates the novel "high-tech" devices.

A most pressing research topic is the development of "normative/descriptive" models of human decision-makers operating in a geographically dispersed and distributed tactical BM/C3 architecture environment. The term "normative/descriptive" is used here to stress that the mathematical models of human decision makers should be based on nonclassical optimization based formulations, which explicitly include constraints that reflect human cognitive limitations, the impact of workload, and the protocols associated with the C2 organization.

Distributed detection, estimation, optimization, and organizational design problems with communications constraints (topics which we shall discuss more in Section 3.5) result in *normative/ prescriptive* solutions; they define superior ways that a team of "agents" should map their nonclassical information patterns into decisions, thus providing a prescription for the optimal team behavior. *Such normative/ prescriptive models are very useful for providing paradigms and help to design experiments by cognitive psychologists which can pinpoint in what precise sense trained human decision makers, and the organization as a whole, deviate from the predictions of normative/prescriptive models and solutions.* Experimental results should then provide "empirical/descriptive" models of individual and organizational decision making. The next challenge is to blend the outcomes of the normative and of the empirical research, using the insight provided by the empirical/descriptive models to introduce additional constraints in the original normative formulation. The new "hybrid" solution, termed normative/descriptive, should yield far better mathematical models of team human decision making and of the performance of the organization as a whole; note that these "hybrid" mathematical models can be used for predictive purposes in subsequent BM/C3 modeling and analysis studies.

Control scientists and engineers can assume a leadership position in this fascinating research area. First, the development of normative/prescriptive models, theories, and algorithms for distributed decision making is a subject of research that has received attention (not enough!) by researchers in the large-scale systems area. Second, control scientists have pioneered the development of mathematical models that can adequately predict the behavior of a human operator in carrying out a well-defined task, so that we do have a reasonable past success record in this area. Third, although there exist many basic research results by cognitive psychologists in modeling the "bounded rationality" of human decision makers, there are no results, at present, in the psychology community that address the types of problems inherent in the distributed tactical decision making environment which is typical in military BM/C3 problems. Therefore, the solutions of pure normative/prescriptive distributed decision problems, and the (often) counter-intuitive nature of the results, will be very valuable in the proper definition of the experimental designs to be carried out by cognitive psychologists. Some research efforts that use the tools of normative sciences (control theory, information theory, mathematical programming, etc) have shown very promising initial results in this area.

3.5 Distributed Situation Assessment.

The generic BM/C3 functions of target detection, tracking, discrimination, and identification (see Section 2.3) serve the purpose of providing a global picture of *situation assessment function* in the C2 process. Knowing in a timely and accurate way the identity and attributes of each and every target, as well as its current location and velocity, is essential in order to construct a list of possible alternative actions and decide on what seems to be the best one.

From a technology point of view, the situation assessment function falls squarely in the domain of modern control theory. Most of the basic research findings in optimal estimation theory, developed during the past twenty five years, have been applied to the situation assessment function with a great deal of success. It is perhaps surprising that there is still a great deal of basic research that remains to be done in order to implement the situation assessment function succesfully in complex BM/C3 systems.

The relevant research directions can be appreciated from the fact that in order to obtain a clear picture of the threat one must "fuse" information from several, possibly heterogenous sensors, which are geographically distributed, each obtaining data from a multiplicity of targets. Thus, *any relevant research in this area must address the generic problems associated with multiple sensors and multiple targets, including the fact that accurate estimation of both continuous (e.g. position) and discrete (e.g. identity) state variables is required.* Hence, the overall problem formulation must include a "hybrid" state space (see also the discussion in Section 3.3).

In multi-target problems we have the generic complexity that even when we are using a single sensor we do not have information over time regarding the matching of sensor measurements and targets. This phenomenon brings up the issue of *data association* which must be performed by the algorithm in addition to its classical detection and tracking function. Technically, this involves setting up a (rapidly growing over time) hypothesis-testing problem that necessitates judicious pruning of the resulting decision tree. The next class of problems often goes under the name of *multisensor correlation,* which requires the exchange of information among two or more sensors in order to improve the hybrid state estimate for a particular target, and this must be done for several targets at each and every instant of time. Since each sensor has a different hybrid state estimate trajectory for each target, the consolidation of information in the *multisensor fusion* problem requires the solution of another large-scale hypothesis testing problem. It should also be noted that identity information is often provided by specialized sensors (e.g. passive ESM receivers, active discrimination sensors) which more often than not have poor location accuracy. In short, it is highly nontrivial to design a superior BM/C3 architecture and associated algorithms that result in an accurate and timely implementation of the situation assessment function in a dense multi-target multi-sensor environment. It should be noted that the presense of multiple hypothesis testing algorithms in such BM/C3 decision structures can be exploited by digital computers with special parallel processing architectures.

The above discussion suggests that the hybrid state estimation problem and the associated large scale hypothesis testing algorithms are only a part of the research challenge. The multisensor fusion problem requires significant tactical communications among the sensors, and these communications are vulnerable to enemy intercepts and/or jamming. It is clear that some communication is necessary to arrive at a superior situation assessment; what is not clear is what is the minimally acceptable exchange of information. Perhaps, each sensor node should have the intelligence to transmit information only when it is clear that this communication is cost-effective. Conversly, each sensor should only transmit information only when requested; the intelligent sensor that requests information should be sure that the received information is worth the cost. It should be self evident that such information transmission options will have a significant impact of the architecture of the situation assessment function in the BM/C3 system. It should be noted that present BM/C3 architectures are notorious for trying to communicate everything to everybody.

The research problems become even more complicated and challenging if we assume that one or more sensor nodes can be destroyed, with some probability, by the enemy. In that case the algorithms that implement the situation assessment function must be distributed so as to improve the survivability of the situation assesment BM/C3 function. *At present we do not have a general theory, accompanied by algorithms, that addresses this class of problems. The development of such a theory will have a significant impact in BM/C3 problems, and will definitely impact the design of superior BM/C3 architectures which also exploit parallel processing in digital computers.* The theory promises to be highly nontrivial because it will require the solution of distributed team-decision problems, with nonoverlapping information patterns including incomplete "models of the world". To make matters worse, there is strong theoretical evidence that the underlying optimization problems are NP-complete; hence, we may have to be satisfied with suboptimal solution algorithms, accompanied however by guaranteed performance bounds.

In spite of their complexity, a very small number of distributed hypothesis testing and estimation problems have been solved during the past few years. Obviously these algorithms are valuable in their own right in automated situation assessment systems. However, the nature of their normative/prescriptive solutions has also provided valuable qualitative and quantitative insight into the decision rules of the completely rational "decision agents" operating in a distributed team decision setting. Indeed, one can see in certain team solutions that the decision rules (mapping of local information into team decisions) of the same decision agent are *very different* than those that would have been employed if the same decision agent was operating in isolation under identical environmental conditions. Another set of valuable insights relate to the fact that in order for a team of decision makers to reach decision-consensus, based on different local information, tentative individual decisions must be communicated to each other with a quantifiable minimum communication frequency. *These findings reinforce the claim in Section 3.4 that the normative/prescriptive solution of distributed decision problems can have some impact in experiments carried out by cognitive psychologists, since from a purely mathematical point of view a perfectly rational decision maker uses different decision rules depending on whether he/she makes a decision in isolation or as a member of a team; such a change in the behavioral pattern, if observed, should not be attributed to the "bounded rationality" of the decision maker.Also, the nature of normative/prescriptive results can flag the monitoring of key observation and decision variables in the human team experiments.*

3.6 Distributed Battle Engagement.

Following the situation assesment function, the BM/C3 system must execute a sequence of real time decisions to implement its defense objectives against the threat. The Battle Planning, Weapon-to-Target Assignment, and Engagement Control functions (see Section 2.3) are the BM functions that implement the battle engagement.

Complex multiple weapon-target engagements have benefited somewhat from available theoretical results in mathematical programming and optimal control theory. However, these studies have been very problem specific and, more often than not, the problem formulations, algorithms and tradeoff studies are classified. To the best of the author's knowledge, there are no unclassified studies that pose these battle engagement problems in a generic setting, exploit the available state-of-the-art, and isolate the advantages and disadvantages of present solution methodologies so as to point out specific basic research directions for future work.

In this class of problems we are concerned with planning and executing several engagements of M weapons against N targets. The problem complexity is related to the different options available to the defense. The more options available to the defense, the harder the problem and the greater the potential payoffs associated with near-optimal decisions. Residual threat uncertainty also contributes to the complexity of the defense decisions.

Generic studies of battle engagement issues should include one or more of the following three types of defense weapons. The potential effectiveness of each weapon can be quantified by its idealized one-on-one kill probability.

(1)*One-on-many weapons*. Such weapons have the potential to kill several targets all at once. The X-ray laser, which focuses the X-ray energy of a nuclear explosion along several beams, is an example of such a weapon. Since such weapons can be very effective to the defense, their *commitment threshold* must be carefuly selected. The presence of such weapons within one or more defense tiers can force the offense to adopt a different offense strategy than simply a saturation attack.

(2)*One-on-one reusable weapons*. Such weapons can engage one target at a time, and must be sequenced over a subset of targets until they run out of resources. Different types of laser weapons (including orbiting mirrors) and machine-gun type weapons fall in this category. Note that such weapons must employ some sort of *target_sequencing algorithm* to decide the order in which they should engage several targets. Optimal target sequencing algorithms must take into account the different times to lock-on to a target, to service it, and to slew it against another target.

(3)*One-on-one non-reusable weapons*. These represent classic one-on-one engagements. Missile interceptors fall in this class. Often, such weapons require additional guidance, and perhaps target designation/illumination, resources to hit their assigned target.

The physical characteristics of the defense weapons interact with the physical attributes of the targets, geometry, speed etc. Typically, a particular target has a finite time-window during which it can be successfully prosecuted by a particular weapon. The decision to commit a particular weapon to that target must obviously take into account this time-varying target vulnerability. Relative target/weapon speed characteristics may require that the weapon-to-target pairing decision be made long before the target vulnerability window. Other strong temporal effects arise when the succesful intercept requires that the target be illuminated by a laser or radar for designation and terminal homing purposes.

One of the neglected areas of research relates to the coupling of the situation assessment and battle engagement functions. Almost all present studies assume that the targets have been localized and identified before weapons are assigned to them. It is very important to capture the residual uncertainty of the threat situation assessment into the very formulation of the weapon-to-target assignment problem. This is particularly important in the midcourse phase of SDI defense, where we must distinguish several thousand re-entry vehicles from many more thousands of decoys. The optimal use of battlespace may necessitate to tentatively assign missiles against targets that have not been fully discriminated. As time goes on, the discrimination function will improve the probability that a particular target is a real threat or a decoy, and there may be ways to divert a missile tentatively assigned to a decoy to engage a real thereat. We note that current practice enforces what is known to control scientists as the "certainty equivalence" principle in stochastic control theory. The class of open research problems that we have discussed employ what is often called the "open loop feedback optimal" policy of stochastic control. Indeed these problems can become very complex if we assume that an active discrimination resource must be scheduled, as a function of time, over a set of targets. Then we must study the simultaneous optimization of the discriminator dynamic schedule and of the weapon-to-target assignment function.

It should be evident from the above discussion that, as a rule, M-on-N engagements have a highly dynamic flavor. Also, stochastic effects are dominant, since kill probabilities are nonunity. The decision variables are both discrete-valued (how many weapons should we assign to a particular target? which weapon should be assigned to what target?) and continuous-valued (when should we launch a particular interceptor? where should we intercept the target?). There may exist specific problem variants that require optimal use of battlespace; in this vein, the possibility of *salvage fusing,* which means that a target with a nuclear warhead explodes when intercepted, results in very challenging decision-dependent state-space constraints in the SDI scenario. Extreme care must be exercised to ensure that the planned intercept trajectories avoid the nuclear fireballs that follow salvage fusing. As a consequence, *stochastic dynamic optimization problems, with mixed integer programming overtones,* are present in most complex battle engagement formulations. In principle, such optimization problems can be formulated as stochastic dynamic programming problems with a high degree of combinatorial complexity. In order to obtain computable solutions one must exploit the structure of these problems, perhaps decompose them into a subset of simpler problems, and then develop algorithms that take advantage of the problem-specific information.

The study of complex M-on-N battle engagements are hard enough even when posed at a centralized level. *The scientific study of distributed battle engagement decision architectures is just beginning.* Presumably, over and above the problems associated with the existence of non-classical information patterns at each decision node, one must study the tradeoffs associated with the vulnerability improvement of the battle engagement decision function vis-a-vis possible misuse of scarce weapon resources (multiple targeting of the same target, targeting of the wrong target, etc). Any new results that provide quantitative insight in this class of problems will be very valuable indeed.

Relevant research in this area must explicitly recognize that the underlying optimization problems are almost surely NP-complete. Thus, there are numerous opportunities for designing novel near-optimal solution algorithms with guaranteed worst-case and expected performance bounds. The distributed version of the battle engagement problems provides fertile new research directions that blend architectural issues, decision-theoretic issues, and communication interfaces that carry the necessary coordinating information.

4. CONCLUDING REMARKS.

In this position paper we discussed, in an informal way, the nature of the research topics that we feel are essential ingredients of a general C2 theory. We need a variety of results that help us understand, analyze, compare, and synthesize BM/C3 systems. We stressed the need for incorporating vulnerability/reliability requirements which dictate the study of distributed decision-oriented BM/C3 architectures.

The roots of this research agenda are in the traditional discipline of control science and operations research. We need a hierarchy of modeling tools which help us understand and model, at different levels of detail and aggregation, complex BM/C3 systems. We need novel theoretical and algorithmic advances in the distributed versions of multiple hypothesis testing, estimation, and optimization problems, stressing nonclassical information patterns, costly communications, and explosive combinatorial complexity. The modeling and analysis of BM/C3 systems with interacting human decision makers poses special challenges in the development of "normative/descriptive" models of human decision makers operating in a team tactical environment. Finally, we need the development of methodologies that help integrate performance and vulnerability of BM/C3 functions and define superior distributed C2 organizations and BM/C3 architectures.

Control scientists and engineers have already pioneered the development of specific C2 quantitative models and analytical tools that improved past practices. These recent accomplishments, although modest, have had a significant impact on the way the military C2 community is thinking. Thus, the present climate is very favorable for basic research in this area, with good opportunities for transitioning basic research into advanced development. The control community is ideally qualified, from a technological point of view, to advance the state of the art in C2 theory.

The challenge relates to the way the basic research is conducted. It is the author's opinion that the major advances in C2 theory will be made by researchers who invest a great deal of effort and energy in understanding and appreciating the complexities and subtleties of military BM/C3 systems. We have stressed that although optimization problems abound, the research issues at a basic level have a very nontrivial combinatorial flavor. Hence, it is the intimate familiarity with specific pragmatic issues that will provide the essential guidance to the researcher on the development of near-optimal algorithms that solve inherently NP-complete problems. It is highly unlikely that the needed research breakthroughs can be *solely* based upon abstract extensions of current theory. It is also evident that collaborative research between control scientists, cognitive psychologists, computer scientists, and communications engineers is required to address the many important dimensions of BM/C3 systems.

ACKNOWLEDGMENTS.

The author has benefited over the past decade from numerous discussions on research priorities and new directions in the area of C2 systems with several colleagues. Special thanks are due to A. H. Levis, J. N. Tsitsiklis, N. R. Sandell, D. A. Castanon, R. R. Tenney, J. G. Wohl, J. J. Shaw, L. C. Kramer, J. R. Lawson, S. Brodsky, C. Holland, J. R. Simpson, W. R. Vaughn, W. Meyers, D. L. Kleinman, M. Melich, J. Offut, J. Wozencraft, P. Girard and W. B. Davenport.

The financial support of the Office of Naval Research under contracts ONR/N00014-77-C-0532, ONR/N00014-84-K-0519 and ONR/N00014-85-K-0782 is also greatfully acknoweledged.

HUMAN PERFORMANCE AND MATHEMATICAL MODELING IN C^2: WHERE IS THE DISCONNECT?

Martin A. Tolcott

Decision Science Consortium, Inc.
7700 Leesburg Pike, Suite 421
Falls Church, Virginia 22043

I. Introduction

Research in command and control is becoming increasingly multidisciplinary, if not quite interdisciplinary. One of the more significant changes during the past ten years has been a recognition by the C^2 community of the critical role of human decision making and information processing. This has led to increasing participation by psychologists in C^2 research programs. The hope has been that with greater understanding and quantification of human performance, mathematical models of C^2 systems could now be enriched to include human functions, and that such enriched models would greatly improve the system design process. This hope has not yet been realized.

An examination of recent C^2 research, much of it described at previous MIT/ONR Workshops, suggests why. With a few exceptions, the work on human performance and the work on mathematical modeling differ in many important respects. These differences are hindering the effective interaction between the two disciplines, preventing them from making a significant combined contribution to C^2 system design.

This paper is an attempt to describe some of these differences, in the hope that making them explicit will promote discussion and strengthen coordination between the two major groups involved, mathematicians and psychologists. The goal is not to suggest that one group is right and the other wrong. It is rather to help all concerned achieve a better understanding of the factors that seem to be preventing a more rapid integration of the two lines of research. I will be guilty of generalizing from examples; there are several notable exceptions, and I will call attention to them as illustrative of moves in the right direction.

II. Theoretical Approaches

The first important difference between the two groups is in their theoretical approaches or starting points. Those mathematicians who have been dominant in C^2 research have tended to start from the viewpoint of control theory, while the psychologists have tended to approach the problem from the point of view of decision theory. This theoretical difference has resulted in significant differences in how they characterize the system with which they are dealing (Figure 1). Control theorists emphasize the input/output characteristics of systems, while decision theorists tend to stress the specific cognitive or information processing functions that are carried out within the system, and the uncertainties characterizing them. The block diagram presented by Jin and Levis [1] (Figure 2)

Theoretical Approaches

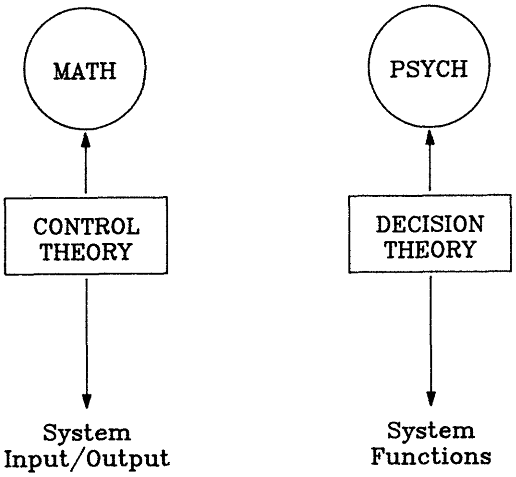
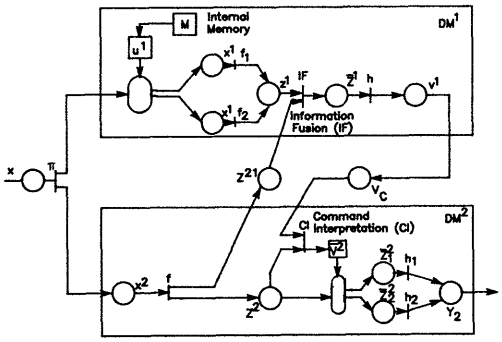


Figure 1

Input/Output

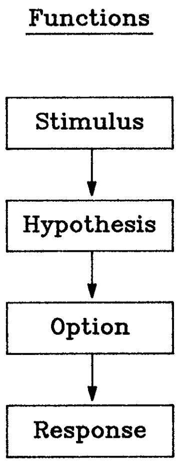


(Adapted from Jin and Levis, 1985)

Figure 2

typifies the initial characterization of a C^2 system based on control theory. In contrast, models developed by psychologists are more likely to be built around some variant of the SHOR (Stimulus-Hypothesis-Option-Response) model as developed by Wohl [2] (Figure 3), or more recently elaborated by Cohen, Thompson, & Chinnis [3] (Figure 4). As shown in Figure 4, it is possible to decompose the functions of situation assessment and choice (or option selection), both of which are critical in command and control, into lower level functions that make explicit the role of uncertainty. These low level functions include tasks such as inference and assessment that are prominent in decision theory and that deal specifically with the handling of uncertainty.

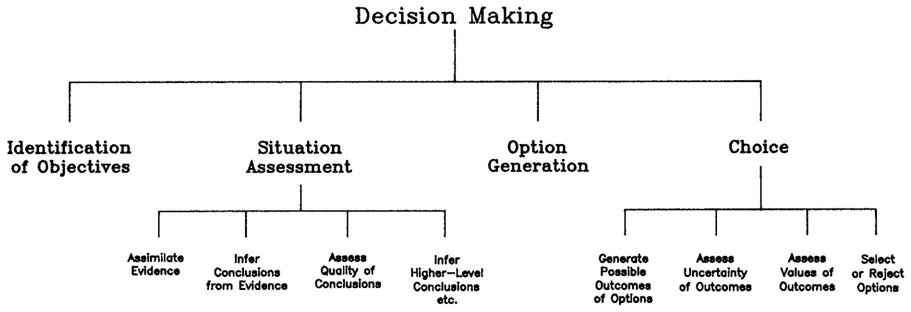
Can these two approaches be integrated? The answer depends on what is meant by "integrated." Mathematical modelers have begun to introduce elements of uncertainty into the tasks they posit for humans in C^2 systems, but their models do not incorporate specific measures of how humans perform in dealing with uncertainty. It may be the case that these functions are at a level of detail unnecessary in C^2 systems models.



(Wohl, 1981)

Figure 3

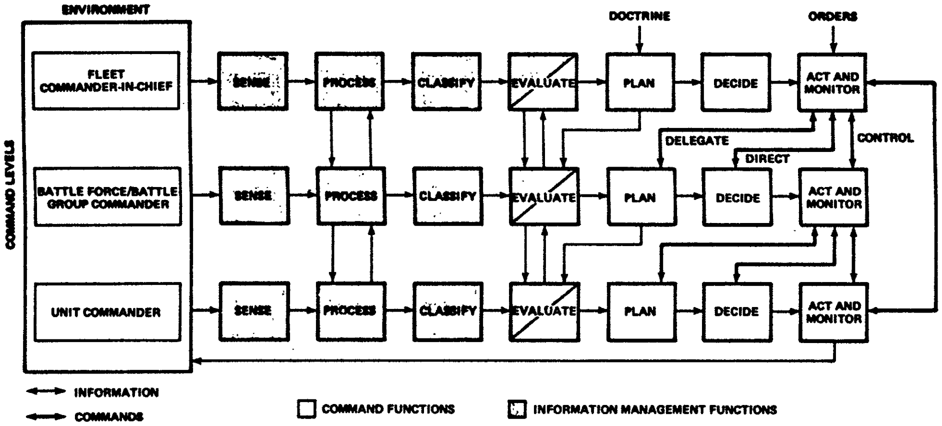
More Detailed Version of SHOR Model


Decision Making
Identification of Objectives
Situation Assessment
Option Generation
Choice
Assimilate Evidence
Infer Conclusions from Evidence
Assess Quality of Conclusions
Infer Higher–Level Conclusions etc.
Generate Possible Outcomes of Options
Assess Uncertainty of Outcomes
Assess Values of Outcomes
Select or Reject Options

(Cohen, Thompson, and Chinnis, 1985)

Figure 4

A Model With Some of Each



(Hamill and Stewart, 1985)

Figure 5

However, two recent efforts suggest that at least some degree of integration is possible. Figure 5 presents a model proposed by Hamill and Stewart [4] which incorporates information processing and decision nodes (such as classify and evaluate) into an input/output format encompassing several command levels in a C^2 hierarchy; it remains to be seen whether they will be able to quantify the functions within those nodes.

Secondly, Goodman [5] and Goodman and Nguyen [6] (among others) are working on ways to combine *linguistic information* (of the type used by humans in characterizing certain types of evidence) with *numerically expressed evidence*, to provide an enriched quantitative model of logical inference in situation assessment. Human-in-the-loop experiments are needed to determine the extent to which human expressions of uncertainty can be converted to mathematical expressions, at levels relevant for C^2 modeling, but these approaches appear promising.

III. Performance Measures

Largely as a result of their differing theoretical approaches, the two lines of investigation tend to utilize different dependent measures of performance (see Figure 6). Consistent with the approach based on input/output theory, mathematical models have relied heavily on information throughput rate as their performance measure. In many early models, errors were converted into delays, on the assumption that incorrect responses could be corrected given enough time. On the other hand, behavioral studies have always recorded not only response speed but also accuracy, and psychologists have been interested in understanding the effects of situational and individual variables not only on accuracy in general but on the types of errors produced. In research on decision behavior, psychologists have tended to use so-called normative models (e.g., Bayesian inference, or maximum expected utility) as criteria, and have interpreted behavioral deviations from these models as evidence of errors or biases in judgment.

Measurement of accuracy in relatively unstructured decision tasks (which are the most interesting in C^2 systems) is admittedly difficult. Most often there is no single correct response, and normative models usually fail to capture the tradeoffs and compromises that are essential parts of the C^2 decision making process. On the other hand, errors are often not con-

Performance Measures

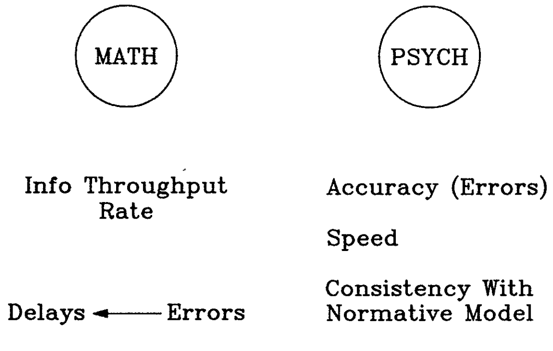


Figure 6

vertible to delays, and knowledge about types of errors is important for the improvement of C^2 systems.

In recent work, Andreadakis and Levis [7] have incorporated measures of accuracy into their C^2 models, by employing a concept of timeliness defined as the ability to respond accurately within a time window of opportunity. This is a step in the right direction, although it does not deal explicitly with the various *types* of errors that are made under different conditions. Figure 7 gives a large set of measures identified by Serfaty and Kleinman [8] as dependent variables that should be considered in behavioral experiments. The authors have been studying team performance in a task similar to air defense. It is obvious that the appropriate measures are specific to the type of task being studied, but their list is rich enough to serve as a menu in designing experiments relevant to C^2. The challenge is still to find ways of incorporating a variety of measures into C^2 models.

A Step Toward Coordination

Task Volume Measures

Number of tasks performed
Number of tasks left unperformed
Number of tasks successfully performed
Number of tasks unsuccessfully performed
Number of tasks acted on with partial information

Reward Measures

Team score
Subject score
Number of assists
Rate at which points are lost

Reaction Time Measures

Initial reaction time
Attack reaction time
Information—attack onset
Safety margin
Average time taken to do a task
Time to resolve unknown tasks

Resource Measures

Over/under assignment of resources
Amount of resources requested/transferred
Resource utilization rate
Frequency of conflicts
Frequency of conflict resolution

Communication Measures

Number of communications per subject
Rate of communications
Communication cost (usage)
Resource/message

(Serfaty and Kleinman, 1985)

Figure 7

IV. Treatment of Variability

Mathematicians and psychologists concerned with C^2 systems tend to deal differently with the concept of performance variability. It is an exaggeration to say that mathematicians try to ignore variability and psychologists tend to emphasize it, but in practice this often turns out to be the case.

The difference emerges early in the conceptualization of a problem. For the mathematician the first step is to posit a system that is mathematically tractable, even if this requires some degree of oversimplification. Then the key features of the system are identified, its inputs and outputs are modeled, (assuming perfect human performance), then the variables that represent the features of interest are parameterized and predictions made of their effects on the dependent outputs. Experiments may be conducted to validate the predictions, but usually these experiments consist of computer simulation runs. The models tend to be at a macro (or system) level, small differences tend to be ignored, even if statistically significant, and variance, if acknowledged, is expressed in terms of probability density functions. The emphasis is on prediction.

The psychologist's approach is first to identify those variables that are likely to affect human performance, and tentatively predict the effects of those variables. An experiment is then devised that permits controlled variation of those factors (the independent variables), even if this requires some degree of oversimplification. Experiments are run with humans to test the predictions, and the effects of the variables are modeled. The models tend to be at the micro (or human) level, and statistically significant differences are emphasized even if they are practically very small. The emphasis is on describing and explaining the variations in performance. The differences between the two approaches are summarized in Figure 8.

To the extent that human performance is significantly affected by C^2 features that can be controlled by system designers, the findings of psychologists' experiments can be valuable, although they would need to be verified in more realistic (i.e., not oversimplified) experimental settings. But a prior step should be to incorporate these findings into the predictive models of the mathematicians in order to determine their significance. The question that should be asked is: if human performance could be improved by the amount suggested by the experiments, how much difference would it make to the overall performance of the system? Unfortunately it is seldom clear how to express "overall

Treatment of Variance

MATH	PSYCH
Acknowledge Variance	Seek Variance
Emphasize Prediction	Emphasize Explanation
Models are Macro (System Level)	Models are Micro (Human Level)
Statistical Significance of Small Differences Ignored	Statistically Significant Small Differences Not Ignored

Figure 8

performance of the system." As discussed earlier, information throughput rate is too narrow in that, as generally used, it fails to incorporate concepts of uncertainty. Recent work by Wohl, Entin and Eterno [9] has taken an important step in relating concepts of uncertainty to human performance variability. Starting with the control theory concepts of *measurement* uncertainty (e.g., sensor reliability) and *process* uncertainty (e.g., control predictability), they have argued that the decision theoretical functions of hypothesis evaluation and option selection must be based on a careful assessment of those uncertainties. Their experimental work, using an ASW context, has suggested that differences in performance among individuals may be due to differences in the way they assess the uncertainties. For example, a commander who thinks his sensor reliability is very high may be more influenced by recent sensor data then one who thinks it is low and tends to dampen out its recent changes. This work has begun to focus on rate of uncertainty reduction as a measure of C^2 performance, and may be pointing the way toward the integration of human judgmental behavior into mathematical models of C^2. Certainly, more work is needed to operationalize this measure, but conceptually it is tempting to regard the processes of hypothesis testing and option evaluation as two aspects of uncertainty reduction, for which *faster* means *better* in C^2 systems.

V. Conclusions

I have described what I regard as some of the major differences between psychologists and mathematicians in their approaches to command and control system problems, differences that I believe are impeding attempts to incorporate human performance measures into mathematical models. I have also pointed to some promising integrative efforts. What else can be done?

First, there should be more deliberate discussions between the two groups about their differences in approach. A show-and-tell meeting with a crowded agenda leaves little time for discussion of issues beyond the immediate paper being discussed. Time should be reserved for discussion of the more fundamental issues raised here.

Second, there should be increased efforts to incorporate the results of psychological experiments into mathematical models of C^2. This is not to assume that such efforts will be successful; they are more likely to be unsuccessful. But the attempts themselves will be fruitful in revealing the causes of failure, and in suggesting what must be changed, the experiments or the models.

Finally, integrative efforts would benefit from closer interaction with C^2 operations experts and system designers, particularly in obtaining guidance about acceptable measures of C^2 performance. How much error or delay can be tolerated under different threat conditions? How much performance variability can be tolerated? Can uncertainty in the system be measured, and would rate of uncertainty reduction be a useful overall measure of merit? Movement toward integration has occurred, but slowly; it should be possible to hasten it.

References

[1] Jin, V. and Levis, A.H. Computation of delays in acyclical distributed decisionmaking organizations. In Athans, M. and Levis, A.H. (Eds.) *Proceedings of the 8th MIT/ONR Workshop on C^3 Systems.* Cambridge, MA: LIDS, MIT, 1985.

[2] Wohl, J.G. Force management decision requirements for Air Force tactical command and control. *IEEE Transactions; Systems, Man and Cybernetics*, 1981, *11*(9), 618-639.

[3] Cohen, M.S., Thompson, B.B. and Chinnis, J.O., Jr. *Design principles for personalized decision aiding: An application to Air Force route planning.* (Technical Report 85-3) Falls Church, VA: Decision Science Consortium, Inc., 1985.

[4] Hamill, B.W. and Stewart R.L. *Acquisition and representation of knowledge for distributed command decision aiding.* Paper presented at Distributed Tactical Decision Making (DTDM) Annual Coordinating Meeting, Newport, RI, 1985.

[5] Goodman, I.R. *A possibilistic approach to modeling C^3 systems.* Paper presented at 9th Workshop on C^3 Systems, Monterey, CA, 1986.

[6] Goodman, I.R. and Nguyen, H.T. *Uncertainty models for knowledge-based systems: A unified approach to the measurement of uncertainty.* Amsterdam: Elsevier Science Publishers, 1985.

[7] Andreadakis, S. and Levis, A.H. *Performance and timeliness in command and control organizations.* Paper presented at 9th Workshop on C^3Systems, Monterey, CA, 1986.

[8] Serfaty, D. and Kleinman, D.L. *An experimental plan for studying distributed tactical decisionmaking.* In Athans, M. and Levis, A.H. (Eds.). Proceedings of the 8th MIT/ONR Workshop on C^3 Systems. Cambridge, MA: LIDS, MIT, 1985.

[9] Wohl, J.G., Entin, E.E. and Eterno, J.J. *Modeling human decision processes in command and control.* (TR-137) Burlington, MA: Alphatech, Inc., 1983.

DEVELOPMENT OF SCHEMA FOR DECISION MAKING

David Noble and Carla Grosz

Engineering Research Associates, Inc.
Vienna, Virginia

ABSTRACT

A recently developed model of human information processing for situation assessment can account for the subjective assessments of a situation based on presented information. This model explains these assessments in terms of memory reference structures called schema. Schema are easily acquired by people trained to recognize situations by the characteristics of their features. Schema are not easily acquired by people trained to evaluate situations by formal computational methods. People trained in this way seem to acquire schema that replace parts of the formal method. These schema allow people to approximate the formal method by relating the observed problem to previously encountered similar instances.

INTRODUCTION

The expert decision making addressed in the article is based on remembering what worked in previous situations. A person using this process might describe his reasons for a decision by saying "I've seen this type of situation before, and the last time I was in this kind of situation I did - - - -. Since this action was successful last time, I will take the same action this time". This kind of decision making typifies many decisions made by experienced people well-trained in a specialty, but it also characterizes many of the ordinary decisions that most people make everyday.

Though this process usually leads to satisfactory decisions, it can lead to some very poor ones as well. It's success depends on a person's ability to recognize that a new situation is the same type as a previously experienced one and that the situation characteristics that made a particular action work in the previous situation are also present in the new situation.

People making decisions this way may feel that their decisions are "intuitive", seemingly made automatically. We assume that such decisions are in fact supported by sophisticated information processing and elaborate memory structures.

This talk describes research that is examining these memory structures and associated information processing. The research overall is concerned with testing whether a particular model (Noble, Boehm-Davis, Grosz, 1986) of this structure and information processing is useful for understanding the relationship between presented information and the assessments and decisions that result from these presentations. The research described here concerns the formation of the memory structures proposed by our model.

This work is funded by ONR as part of a special focus program in distributed decision making.

OVERVIEW OF MODEL

The proposed model is built from memory reference structures called "schema". These structures are characterized by variables, a hierarchy of embedding, and varying levels of abstraction which "attempt to represent knowledge in the kind of flexible way which reflects human tolerance for vagueness, imprecision, and quasi-inconsistencies" (Rumelhart, 1977). As recognition devices their "processing is aimed at the evaluation of their goodness of fit to the data being processed" (Rumelhart, 1980).

Each schema (or network of schema) characterizes a type or class of situation. It contains information that enables a person to evaluate a particular observed situation in terms of that class, and to assess the degree to which the observed situation has properties characteristic of that class.

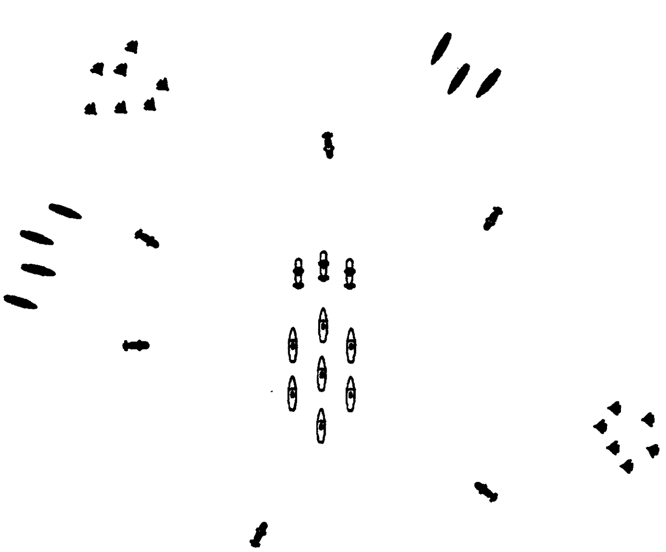
Recent experiments have examined schema for two classes of situations: "all-out attacks" and "barriers". Schema for all-out attacks enable a person to evaluate all-out attacks. A person could use his all-out attack schema to assess the strength of attack represented by observed ships, submarines, and aircraft. Figure 1 is a picture of an "all-out attack" taken from recent experiments testing this schema model. In this figure the Battle Group (white) is being attacked by hostile (black) ships, submarines, and aircraft. The all-out attacks will serve in the following description to illustrate general properties of schema.

The schema used in our model have three layers: a slot layer, a criteria layer, and an action and inference layer. The slot layer defines the situation features relevant to the schema. These features are objects and spatial or temporal relationships among objects. The slot layer defines the physical and functional properties of features used to evaluate to the class of situations represented by their schema. All-out attacks have four features, corresponding to four schema slots. These features are the number of ships, the number of aircraft, the number of submarines, and the number of quadrants surrounding the Battle Group that contain the submarines.

The second layer contains data used for feature assessment. It contains "criteria curves" that allow features to be assessed. These criteria curves relate measurable properties of features to feature strength. In an all-out attack example, a strong feature has characteristics expected in strong attacks, and a weak feature has characteristics expected in weak attacks. The feature "number of ships" would be rated high (strong) in an attack with seven ships, and rated low (weak) in an attack with two ships.

The third layer contains inferences to be made or actions to be taken for situations evaluated by the schema. In the case of all-out attacks, this layer would specify actions to be taken to counter an all-out attack of a particular strength.

Three information processing steps use the data in a schema to assess observed situations. These steps are feature identification, feature assessment, and feature combination. The



OPTIMAL LAUNCH RADIUS

Figure 1. An example of an all–out attack.

first step finds in the situation features needed for situation evaluation. In all-out attacks these features are number of ships, number of submarines, number of aircraft, and number of quadrants around the Battle Group containing submarines. The second step evaluates each feature. In the all-out attacks, the feature "submarine deployment" would be rated very strong if submarines were observed in three of four quandants; it would be rated very weak if submarines were observed in only one quadrant. The third step combines the features. In our model, the assessment of the combined features is the geometric mean of the assessments of the individual features. In the example, this geometric mean is the predicted effectiveness of an all-out attack conducted by the observed hostile platforms.

Recent experiments have shown that this model can be used to predict peoples' assessments of attacks or barriers from measurable properties of presented information. In these experiments subjects were trained to evaluate situations by seeing a sequence of example attacks or barriers described in terms of their features. These subjects easily learned to evaluate attacks, giving assessments that approximated an expert's independent assessment and that were stable over time.

The subjects in our experiments seemed to base their assessments on schema, the general models for situation classification and evaluation, rather than basing them on memory of the specific instances seen in training or on a conscious calculation rule. Several tests for schema were identified during these experiments. Two important tests are accuracy and predictability. Accuracy was evaluated by comparing subjects' evaluations with the evaluations predicted from a training model. Subjects were trained by observing pictures developed from a schema model which was not described to the subjects. Subjects' later evaluations of different test pictures accurately reflected this model. Predictability was evaluated by comparing the geometric means of feature assessments provided by subjects with their assessments of the overall situation. The correlation between the geometric means and overall assessments, averaged over subjects, exceeded .98.

TESTING THE USE OF SCHEMA IN DECISION MAKING

The work described previously validates the utility of our model for predicting situation assessments from measurable properties of depicted situations. In that work subjects were trained by seeing examples explained in terms of features. The work did not test whether people could acquire schema when trained using other methods, nor did it test the link between situation assessment and decision making. The experiments described below tested these issues.

The task

Figure 2 is an example of a situation assessment and decision task in experiments that tested the use of schema in decision making. This diagram depicts a hostile barrier consisting of ships and submarines. The ships are shown explicitly. Since the submarines cannot be localized precisely, they are represented by the cross-hatched submarine patrol areas. The Battle Group is stationed at the juncture of the straight and curved paths. The straight and curved paths represent two possible Battle Group courses.

Subjects were instructed to select a Battle Group action by evaluating the number of hits received by the Battle Group along each path. They are told to take the path where the Battle Group receives the fewest hits provided that this number of hits is six or fewer, and otherwise to stay in place.

The method for calculating hits is complicated. Hits from ships are calculated in a sequence of discrete time steps. At the end of each time step hits from hostile ships are determined from the geographic relationship between the hostile ships and the Battle Group. Specifically, subjects are told that the Battle Group can move from one dot to the next in one hour, and that hostile ships can move in any direction by an amount shown in the figure legend. The Battle Group receives one hit at the end of each hour from each hostile ship within strike range. Strike range is also shown in the figure legend. Total Battle Group hits along each path are the sum of the hits at the end of each time step.

Hits from submarines are calculated differently. Subjects are told (not in these words) to divide the submarine patrol area in half along the patrol area's horizontal center of gravity, and to mark the center of gravity of each of these halves. Since there are four patrol areas, there are eight center of gravity points. The Battle Group received one hit for each center of gravity point within the strike range of the Battle Group path.

This procedure is sometimes difficult for people to learn, and careful picture evaluation by these rules takes about six to eight minutes per picture.

Experiment procedure and hypotheses

The procedure in the experiments to test the initial hypotheses contained three phases: training, path ranking, and barrier quality assessment. During training subjects both computed path quality using the algorithmic procedure described above and also reviewed the results of such calculations. In ranking paths subjects ranked the three alternatives (straight, curved, stay) in order of desirability. In assessing barrier quality subjects rated the overall quality of the barrier and the quality of the barrier along the straight path and along the curved paths. Subjects were given fifteen to thirty seconds to rank each alternative and assess each barrier. These times were much to short for subjects to use the formal algorithmic procedure for computing the number of hits received by the Battle Group along each path. These rankings and evaluations were intended to determine whether subjects would be able to replace the algorithmic rule for calculating path quality with a schema, and whether subjects' rankings of alternative quality (straight, curved, stay) would follow directly from assessments made using this schema.

We assumed that as people learned this task, they would stop using the detailed procedure described above for calculating the number of Battle Group hits, and instead would begin to estimate these hits from the "look" of the hostile forces around

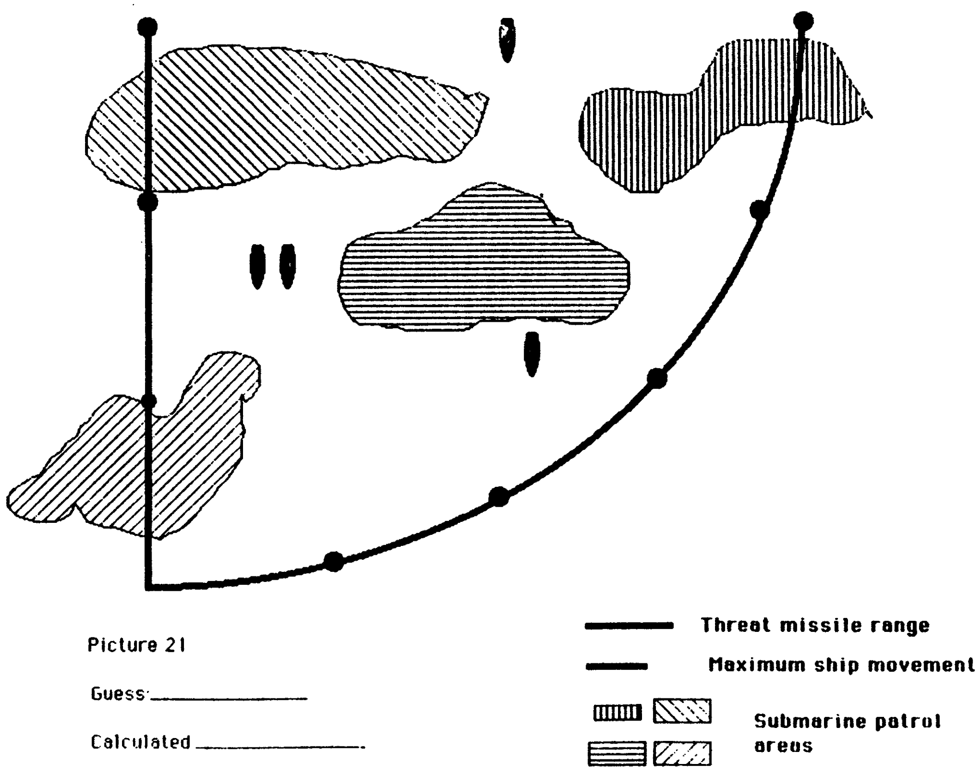


Figure 2. An example of a situation assessment and decision task.

each path. We thought that with sufficient experience people would begin to associate types of barriers with particular outcomes. We hypothesized that schema would be formed for path quality even when people were trained to follow a specific computational procedure. It is not necessary for people to be trained from examples described in terms of features for schema to form.

We also assumed that alternative rankings would follow immediately from the schema-based assessment of barrier quality. We assumed that subjects would equate path desirability for the Battle Group with barrier weakness. Subjects would use their schema to evaluate the quality of the barrier along each path. They would rank the path through the weaker part of the barrier more desirable than the path through the stronger part of the barrier.

Tests for schema formation

Since path rankings did not follow from barrier quality assessments, tests were performed to determine if schema had been formed. If schema were not present, then barrier assessments made at different times might vary considerably. In this case it is possible that the subjects ranked alternatives based on their assessments barrier quality, but made a different assessment later when when asked to rate barrier quality.

The two tests for schema presence performed were assessment accuracy and assessment predictability. Schema accuracy was measured by the fraction of times subjects identified the "book value" best option correctly, as evaluated by the procedure taught in training. Predictability was measured by the correlation between the geometric mean of the feature ratings and the barrier assessment.

In order to test assessment predictability additional data was collected to determine whether schema were formed. Subjects were asked to evaluate two features for the overall barrier and for each part of the barrier. These features were "enemy ships near path" and "path is patroled by enemy submarines".

Neither test indicated that subjects had acquired stable schema. Subjects picked the correct alternative to be best only half the time. Since there were three alternatives, chance is .333. The correlation between geometric means of the feature ratings and subjects' assessments was 0.90 for the straight path and 0.497 for the curved path. These data indicate that the subjects, though performing better than chance, were not performing very well.

Relationship between alternative rankings and barrier evaluations

Table 1 shows the number of times that the part of the barrier evaluated weakest corresponded to the path selected as the most desirable alternative. These data clearly do not substantiate our hypothesis that alternative rankings follow directly from schema rankings.

Subject	# of matches (out of 12)
1	5
2	8
3	10
4	6
5	6
6	7
7	9
8	7
9	8
10	7

Table 1. Number of times each subject ranked the path through the part of the barrier evaluated weaker higher than than the path through the part of the barrier evaluated stronger.

SCHEMA FORMATION

The above data indicated that while schema had not fully formed and had not stablized, schema might be forming. Additional experiments were performed to examine whether schema were forming, and if so, to determine what these embryonic schema look like.

Two different hypotheses for schema formation were considered. The first (Lewis, 1985) proposes a process of feature abstraction and refinement. The second proposes that schema replace pieces of the rule-based procedure taught in training. As schema formation progresses, the schema grow more powerful and replace more of the rules.

Feature abstraction and refinement

The feature abstraction and refinement process is a hypothesis and test sequence in which people associate situation features with situation outcomes. According to this method, people apply a rule that links situation features with different outcomes to specific actions. When this rule works, it is not changed. When the rule does not work, it is refined, often by being made more specific.

Special materials were developed to test this process. Each picture in these training materials contained a special feature indicator that by itself conveyed the best path. The feature indicating the straight path was a ship to the right of the curved path. The feature indicating the curved path is twin ships somewhat to the right of the upper part of the straight path. The feature indicating that "stay" was best was overlapping submarine patrol areas. The test pictures also contained these features, but in these pictures the feature indicators were no longer associated with the best alternative.

It is postulated that if feature abstraction and refinement accounted for schema formation, then our subjects would use the indicator features in making their judgments. Our data showed that such feature indicators were very seldom used. In a few cases subjects consciously noticed these special feature indicators. When they did, they selected the Battle Group alternative associated with the feature indicator. Those subjects who said that they did not notice these indicators did not select alternatives associated with these indicators.

Our data show no support for schema formation by feature abstraction and refinement.

Rule approximation

After training for an hour some of our subjects were able to assess Battle Group hits reasonably well by examining the picture for about twenty seconds. Since the formal calculation procedure takes about six to eight minutes per picture, these subjects must have found a method to quickly approximate the answer. If this method relies on memory reference structures, then these structures may be the precursors of fully developed barrier evaluation schema.

These subjects reported that they estimated Battle Group hits along each path by an "eyeball and count" method. In this method subjects examine each hostile ship and submarine patrol area, estimated the hits from this platform, and added over platforms. It was proposed that this process would be supported by "mental rulers". Mental rulers are a set of memory based reference distances for estimating whether each ship can place a hit on the Battle Group at each dot marking a Battle Group movement step. There is a different mental ruler for each of the movement steps.

Such rulers could be accurate guides for computing Battle Group hits. In fact, the formal process specified in training can be replaced by a much simpler process of counting the number of well-formed circles in figure 3 that contain ships. This process gives the same answer as the ship movement algorithm if no ship is counted as being in more than two circles.

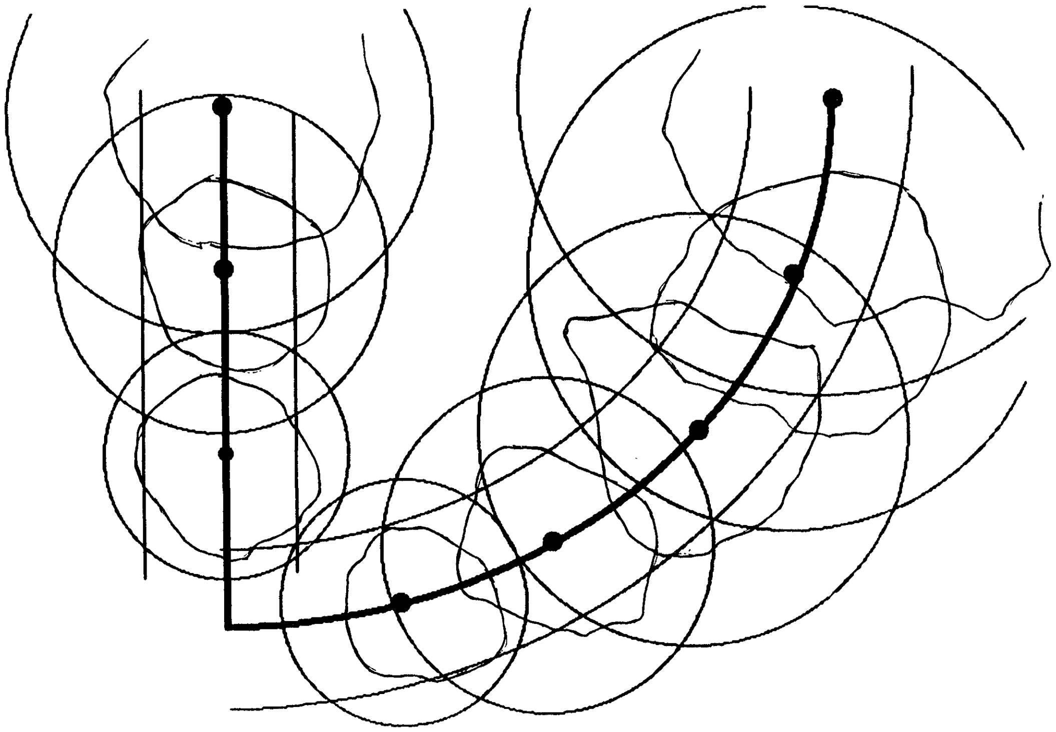


Figure 3. Range circles for barrier evaluation, and subjects approximation to range circle

Two tests for the use of "mental rulers" were performed. In the first, we asked two subjects to reproduce the circles by estimating for each movement step dot the furthest distance a ship can be at the start of the Battle Group movement and still score a hit at that step dot. Figure 3 (poorly formed circles) are the circles produced by one of these subjects. These circles were drawn on charts that did not include the reference scales in the picture legend.

In the second test we prepared slides that showed only a single ship in relation to the straight and curved paths. There were no reference circles, nor were there reference scales in the picture legend. There were five kinds of test slides prepared. Type A showed ships at positions evaluated by subjects four times during training. Type B were type A ships reflected symmetrically across the ship path. Type C were type A ships displaced parallel to the circumference of an appropriate well-formed circle in figure 3. Type D were displaced outward from type A, and type E were displaced outward from type C.

Results from these tests also show mental rulers apparently being formed. Subjects got 65% of type A and B and 45% of type C correct. They seemed to be forming mental rulers by generalizing from specific instances computed in training. They were equally good both for the instances seen in training and for symmetric reflections of these instances. They also were fairly proficient in ships moved parallel to the range circle. As expected, the subjects performed much worse on types D and E, ships moved outward from the range circles. On these they got only 24% and 31% correct.

The subjects also did better on ships related to the straight path than to the curved. They got 56% correct on the straight path, and 43% current on the curved.

CONCLUSIONS

Schema models of situation assessment describe the memory structures and information processing used in situation assessments based on situation recognition and classification. One such model seems to account for subjects' situation evaluations in experiments where subjects are trained to evaluate situations by being shown examples described in terms of features. When trained this way, subjects quickly learn to evaluate situations accurately.

The schema model for situation assessment did not work well when subjects were taught to evaluate situations by a specific algorithm. Data indicate that subjects so trained do not easily acquire the schema that facilitate rapid situation assessment. Experiments investigating schema acquisition in these cases revealed the formation of schema able to replace some of the algorithm steps. These schema seem to be formed by generalizing from specific instances seen in training.

REFERENCES

Lewis, M.W. , and Anderson, J.R. (1985). Discrimination of operator schema in problem solving: Learning from examples. Cognitive Psychology, 17, 26-65.

Noble, D. , Boehm-Davis, D, and Grosz, C. (1986). Schema-based model of information processing for situation assessment. (Technical Report R-621-86) Vienna, VA: Engineering Research Associates.

Rumelhart, D.E. (1980). Schemata: the building blocks of cognition. In R. Spiro, B. Bruce, and W. Brewer (Eds), Theoretical issued in reading comprehension. Hillsdale, NJ: Lawrence Erlbaum Assoc., 33-58.

Rumelhart, D.E., and Ortony, A. (1977). The representation of knowledge in memory. In R.C. Anderson, R.J. Spiro, and U.E. Montague (Eds.) Schooling and the acquistion of knowledge. Hillsdale NJ.: Lawrence Erlbaum Assoc.

Experimental Gaming for Command, Control and Communications

Prof. Michael G. Sovereign
and
CDR Joseph S. Stewart II

Naval Postgraduate School

Introduction

A series of annual experiments in Command, Control and Communications have been held at NPS over the last three years. These experiments have been sponsored by the Defense Communications Agency under the auspices of the Joint Directors of Laboratories C3 Basic research program. These experiments have used a computer-aided wargame and a large number of players to provide data for investigation of C3 issues which are of broad interest to the community. This paper describes those experiments, details the data gathering methods of the most recent experiment, and provides an introduction to the results obtained when the Headquarters Evaluation and Assessment Tool (HEAT) methodology developed by Defense Systems Incorporated (DSI) is applied. DSI has supported the experiments and analyzed the data in each year. A comprehensive summary of their work is [1]. The issues addressed to date include connectivity, centralization and command role. The data for each experiment include thousands of observations gathered through a month of experimentation representing several thousand officer subject hours in realistic battle command situations.

For each of the experiments in the series a consensus was reached by the three participating organizations, DCA, DSI and NPS as to the specific subject which would be investigated. In general the investigations concerned command and control structures and their performance, how these structures might be modified by design, or how they might change during the course of a series of stressful events. A constraint was that the computer laboratory environment would allow the games to be replicated, and that resultant data from a series of iterations would support statistical analysis. During the series of experiments it was found that the team was able to present realistic problems using the wargaming system, that the subjects (who were officer-students) made reasonably effective decisions, and that a series of short gaming events produced data which could be analyzed statistically. In addition, the experiments could be controlled to reduce the effects of learning and to explore minor changes in the command and control system architecture which was being simulated. The wargame (hardware and software) used is the Navy's Interim Battle-Group Tactical Trainer (IBGTT) developed by the Naval Ocean Systems Center and currently in use by the Tactical Training Group, Pacific. Generalization from these results is of course dangerous, but a continuity of results over a considerable scale (one to four carrier groups) and range of scenarios has been shown (Sea of Japan, Persian Gulf and Norwegian Sea).

C2 Laboratory Experiment - Connectivity

The first experiment of the series was an attempt to corroborate Soviet [2] findings which indicated that the command effectiveness as measured by the speed and correctness of its decisions, of a battlefield headquarters is influenced by the command structure. In the NPS Wargaming Analysis and Research Laboratory (WAR lab) a set of military problems were presented to subjects who were organized in increasingly connected command structures ranging from star to fully-connected. This experiment, conducted in November 1983, utilized the Navy's Combined Warfare Commander (CWC) concept to represent the distributed headquarters of a hypothetical battle group in the Sea of Japan. The data collection plan was designed to allow the use of HEAT measures to quantify the activity of the headquarters units regardless of their relationship and ability to communicate with the other headquarters in the command. For example, one measure was time to complete a planning cycle. The results of this experiment are reported by the authors in a paper in the ONR-MIT Theory of C3 Conference series, Ref. [3] and by DSI. DSI could not discount the findings by the earlier Soviet researchers but did indicate that there were differences in speed of action and error rates depending upon the command linkages and communications structure, as shown in Table 1 from the ONR-MIT paper.

Table I

* Star structures are slightly faster than fully-connected structures but not to a statistically significant level.

* (Did not contradict Soviet findings.)

* The fully connected structure was able to reach a decision more often than the star structure but the decision error rate was about the same.

* (Did not contract Soviet findings.)

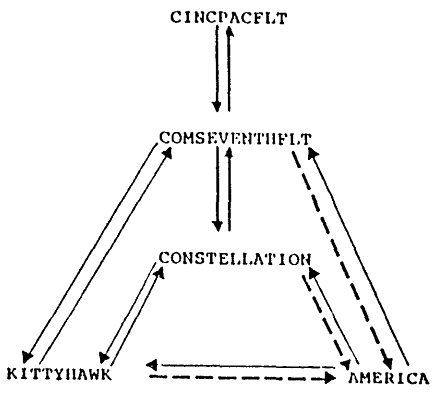
* Fully connected structures were always slower to initiate hostilities mistakenly than were other structures.

* (An independent finding.)

Organizational Responsibility - Centrality

The second in a series of experiments was conducted in October of 1984. The objective was to examine alternatives in the degree of centrality of information and decision making in a multiple-carrier battle force. For this experiment a headquarters was created which represented a battle force of three carrier battle groups. They operated together in an environment which was rich with potential adversaries around the Straits of Hormuz and Persian Gulf. The design allowed the command responsibilities to be varied such that each carrier was responsible for every event in its vicinity, known as geographic or decentralized responsibility, or only for specific types of response over the whole area of conflict, known as functional or centralized. A two-way electronic mail system was added for this experiment to provide controlled communications including jamming, which was an experimental variable. Figure 1 represents the basic organization tested. The relative performance against small, discrete threat problems and against larger, complex threats were hypothesized as shown in Figure 2. Forty-five players were arranged into teams such that each set of three teams experienced six variants of the game according to the experimental design shown as Figure 3. The end game was varied to avoid learning effects. The data that was generated was again analyzed and HEAT scores were assigned where appropriate. The results are shown in Figure 4. This experiment was one of the first attempts to analyze the organizational interaction of three aircraft carriers, simultaneously dealing with the same series of problems, using analytical methods with man in the loop. The effort was therefore significant in its own right and formed a bridge to the experiment of 1985.

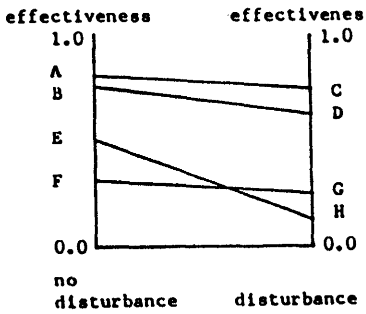
EXPERIMENTAL COMMUNICATIONS AND DISTURBANCES



- TWO-WAY ELECTRONIC MAIL COMMUNICATIONS NETWORK
- DASHED LINES INDICATE A TYPICAL DISTURBANCE

FIGURE 1

C^2 organizational experiments: anticipated findings



A - geographic, discrete
B - functional, discrete
C - geographic, discrete
D - functional, discrete
E - functional, complex
F - geographic, complex
G - geographic, complex
H - functional, complex

Figure 2

Sequence of Scenario, Structure, Disturbance, and End-Game

Group 1	Group 2	Group 3
Iraq functional clear comm attack	USSR geographic clear comm intimidate	Iran & Iraq geographic clear comm provoke
Iran geographic clear comm attack	Iran geographic clear comm provoke	nobody functional clear comm intimidate
Iraq & USSR functional disturbance intimidate	Iraq & USSR functional disturbance attack	Iraq geographic clear comm attack
Iraq & Iran geographic disturbance attack	nobody geographic disturbance intimidate	Iraq & USSR geographic disturbance attack
USSR geographic disturbance provoke	Iraq & Iran functional clear comm attack	Iran functional disturbance attack
nobody functional clear comm provoke	Iraq functional disturbance provoke	USSR functional disturbance provoke

Figure 3

C^2 organizational experiments: observed vs expected

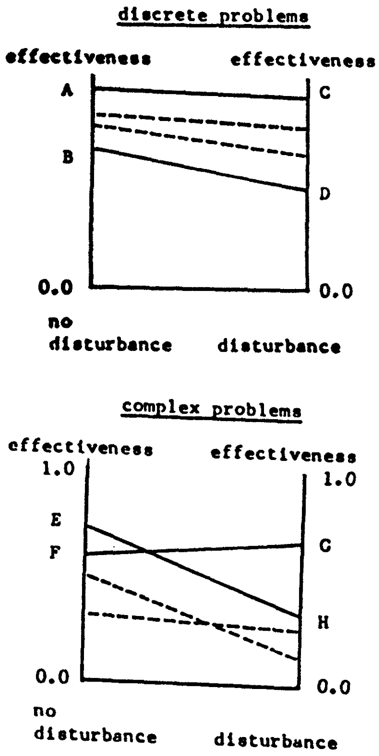


Figure 4

Taking the Fleet Exercises to the Laboratory - Role Experiment

Navy operational fleet commanders are interested in obtaining answers to emerging questions of strategy and tactics such as how to fight a multiple carrier battle force. In the past large-scale exercises have been used to good advantage to seek answers to some of these questions. Now, however, it is frequently the case that the full-scale exercise costs limit the number of these exercises that may be conducted. As a substitute it has been proposed that the large computer-aided wargaming facilities now becoming available on both coasts under fleet control may be used as a substitute for some exercises. To a limited degree it may be possible to create tests for the same questions in the laboratory which has additional advantages. In 1985 a large exercise was run under fleet sponsorship which involved a battle force of three aircraft carriers. The third laboratory experiment in our series was designed after the earlier exercise with the possibility that the results of the two endeavors would be comparable. The WAR Lab simulation provided to the subjects a representation of the same friendly forces and challenged them with the same threat environment as did the exercise which had been conducted both at sea and in-port in a series of Battle Force-Inport Training (BFIT) exercises. Although the laboratory is much less realistic than the exercise, it offers the following advantages discussed below, ease of data extraction and design flexibility so that a range of alternatives can be explored in a variety of environment scenarios. A high comparison of the experiments and the exercises are shown as Table II.

Table II. Comparison of NPS Experiments and Fleet BFITs

	NPS Experiments			Fleet BFITs	
	Connectivity (1983)	Centrality (1984)	Role (1985)	2-85	1-86
Representation of Combat					
Degree of realism	2	2	2	4	4
Degree of control	3	3	3	2	2
Contextual Variables					
HEAT cycle frequency (per hour)	0.67	4.0	4.0	3.1	3.3
Type of warfare	Conventional	Conventional	Conventional	Conventional	Conventional
Friendly units monitored	31	84	154	236	398
Enemy units monitored	55	39	155	77	118
Problem complexity	12	12	12	10	10
Problem subtlety	9	18	4	4	4
Defining Variables					
Echelon	BF/BG	BF/BG	BF/BG	BF/BG	BF/BG
Role type	Objective-specific	Objective-specific	Objective-specific	Objective-specific	Objective-specific
Subordinates	31	84	154	28	67
External nodes	3	3	3	9	7
Internal nodes	4	4	5	5	5
Links per internal node	3.75-5.25	5.25	6.2	7	7
Space distribution	Dissociated	Dissociated	Dissociated	Dissociated	Dissociated
Connectivity	50-100%	100%	100%	100%	100%
Capacity Variables					
Personnel	4	12	13	40	40
Grade of planners	03-05	03-05	03-05	03-07	03-07
Unit experience	0	0	0	0-1	0-2
ADP usage	Monitor, Inform	Monitor, Inform	Monitor, Inform	Monitor	Monitor
External linkage reliability	100%	100%	100%	98+%	98+%
Internal linkage reliability	100%	96%	94%	98+%	98+%
External linkage capacity	Data	Data	Data	Image	Data
Internal linkage capacity	Text	Text	Text	Voice	Voice
External linkage medium	Wire	Wire	Wire	BLOS radio	BLOS radio
Internal linkage medium	Wire	Wire	Wire	BLOS radio	BLOS radio
Dependent Variables					
HEAT cycle time (directives) (min.)	4	--	--	23	11
Understanding correctness (enemy intent)	50%	65%	75%	100%	100%
Understanding completeness (enemy intent)	71%	--	78%	100%	67%
Prediction completeness	--	--	21%	34%	13%

The lessons learned in the 1984 experiment provided guidance for the design of the organization which would be simulated. Although we investigated the geographic and functional organizations in the previous year, it appeared that a hybrid would be the more logical choice for a commander faced with the resources and the problems of operating a three-carrier battle group. The design then included half the sessions run with the strike coordinator being a functional entity and the Allied ASW coordination being run in a functional manner in all sessions. Figures 5 depicts the organizational charts for these designs.

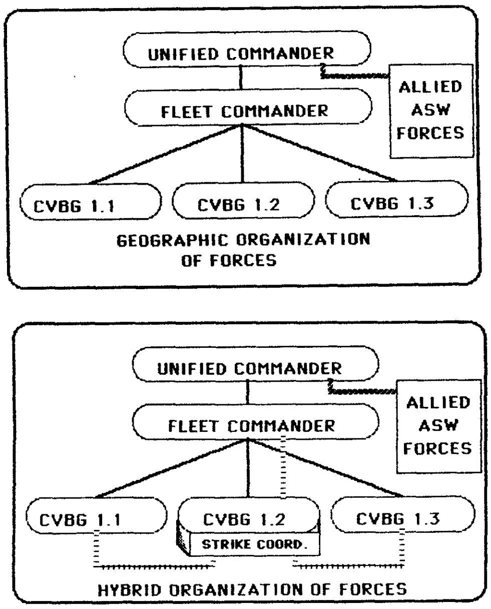


Figure 5

The quantity of data generated by any single game is large. To provide the reader with the scope of material it is necessary to understand the make up of one command position. Figure 6 shows this configuration as a collection of three terminals which interact with the main computer and a geotactical display which shows the force layout in planform. During a game, orders to control combat units are entered at the player position and automatic acknowledgements are received, including attempted actions which cannot be supported at that time. Information is extracted from the game data bank by single keystroke at the status position. Communications are sent and received at the comm position.

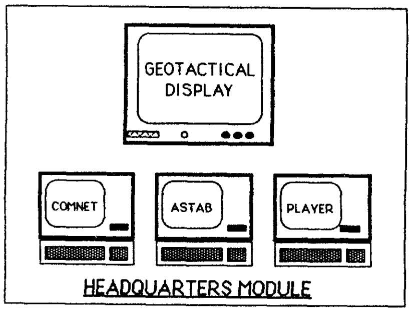


Figure 6

Through modification of the electronic mail software of the previous year, we were able to provide structure to the communications between headquarters such that the intent of the decision makers could be more easily analyzed with pre-formatted reporting requirements. In addition, the software provided copies of each individual message from each headquarters unit and aggregate statistics about rates of activity at each node in the network. Moreover, by the addition of observers at each cell we were able to quantify the activity of each cell to a greater degree than in the past. Each observer was armed with a laptop computer which was preprogrammed to accept single keystroke entries of seven items of interest to the HEAT analysts. The article are shown in Table II. The observers could then rapidly record events that would show up in no other record and these events could be compared with other records. A time-dated series of observations could then be dumped to the host computer.

The complete record of the headquarters activity consists of (1) the player record, which is all the orders entered at this terminal and the computer responses, (2) the commmunications record, which is every message sent by the terminal and an analysis of the terminal players time to generate messages and to respond to incoming messages throughout the game, and (3) the file of observer responses. These constitute a headquarters packet. For each game there were three headquarters packets and, in addition, there is a complete file of umpire orders which includes the moves of the opposition, and a file containing a record of engagements, damage, and the speed at which the game was progressing. Figure 7 depicts orders which might be entered by a player, in this case, the umpire control console operator, and appropriate responses from the software about recent orders.

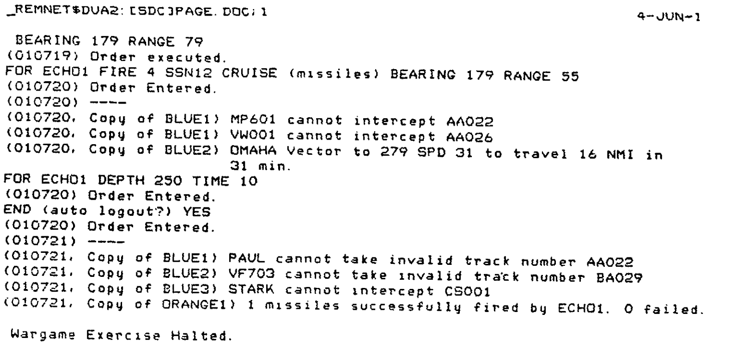
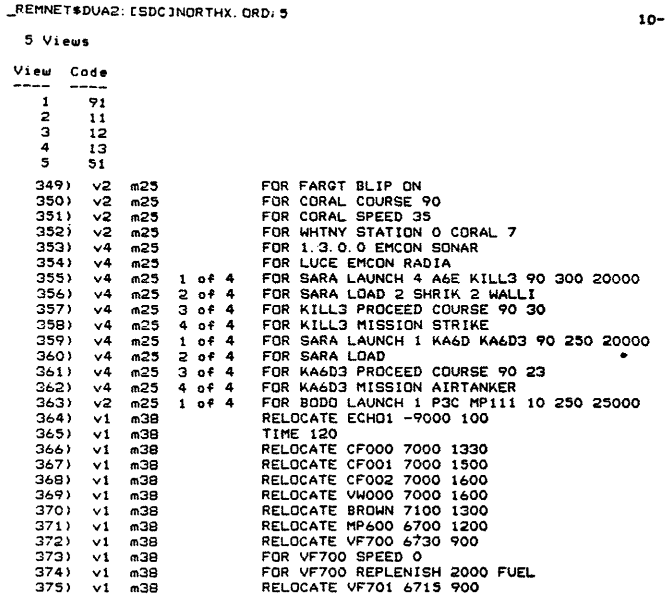

```
_REMNET$DUA2:[SDC]PAGE.DOC;1                                           4-JUN-1

 BEARING 179 RANGE 79
(010719) Order executed.
FOR ECHO1 FIRE 4 SSN12 CRUISE (missiles) BEARING 179 RANGE 55
(010720) Order Entered.
(010720) ----
(010720, Copy of BLUE1) MP601 cannot intercept AA022
(010720, Copy of BLUE1) VW001 cannot intercept AA026
(010720, Copy of BLUE2) OMAHA Vector to 279 SPD 31 to travel 16 NMI in
                        31 min.
FOR ECHO1 DEPTH 250 TIME 10
(010720) Order Entered.
END (auto logout?) YES
(010720) Order Entered.
(010721) ----
(010721, Copy of BLUE1) PAUL cannot take invalid track number AA022
(010721, Copy of BLUE2) VF703 cannot take invalid track number BA029
(010721, Copy of BLUE3) STARK cannot intercept CS001
(010721, Copy of ORANGE1) 1 missiles successfully fired by ECHO1. 0 failed.

 Wargame Exercise Halted.
```

Figure 7.

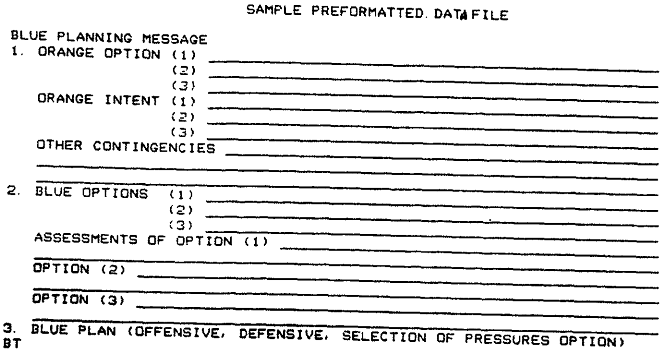
Figure 8 is an aggregated listing of all orders entered by all players from which move and countermove can be traced for postgame analysis. Figure 9 (TOP) is an example of a preformatted message which is presented to a communicator and a message which results from the use of the system (bottom) complete with time tags which have been added by the software. Figure 10 aggregates the results of many messages between available nodes in the category "throughput time", which is the sum of construction time, transmission and reception time increments. Observations made by observers which were colocated with the players in each headquarters cell are shown in Figure 11. Various codes depict significant events as shown in Table III. Ultimately the results of all these messages, game orders and decisions are

Figure 8



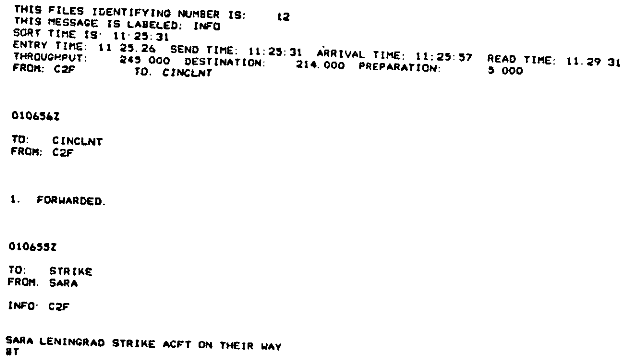
```
_REMNET$DUA2: [SDC]NORTHX.ORD;5                                10-

 5 Views

View  Code
----  ----
   1    91
   2    11
   3    12
   4    13
   5    51
   349)  v2  m25           FOR FARGT BLIP ON
   350)  v2  m25           FOR CORAL COURSE 90
   351)  v2  m25           FOR CORAL SPEED 35
   352)  v2  m25           FOR WHTNY STATION 0 CORAL 7
   353)  v4  m25           FOR 1.3.0.0 EMCON SONAR
   354)  v4  m25           FOR LUCE EMCON RADIA
   355)  v4  m25  1 of 4   FOR SARA LAUNCH 4 A6E KILL3 90 300 20000
   356)  v4  m25  2 of 4   FOR SARA LOAD 2 SHRIK 2 WALLI
   357)  v4  m25  3 of 4   FOR KILL3 PROCEED COURSE 90 30
   358)  v4  m25  4 of 4   FOR KILL3 MISSION STRIKE
   359)  v4  m25  1 of 4   FOR SARA LAUNCH 1 KA6D KA6D3 90 250 20000
   360)  v4  m25  2 of 4   FOR SARA LOAD                           •
   361)  v4  m25  3 of 4   FOR KA6D3 PROCEED COURSE 90 23
   362)  v4  m25  4 of 4   FOR KA6D3 MISSION AIRTANKER
   363)  v2  m25  1 of 4   FOR BODO LAUNCH 1 P3C MP111 10 250 25000
   364)  v1  m38           RELOCATE ECHO1 -9000 100
   365)  v1  m38           TIME 120
   366)  v1  m38           RELOCATE CF000 7000 1330
   367)  v1  m38           RELOCATE CF001 7000 1500
   368)  v1  m38           RELOCATE CF002 7000 1600
   369)  v1  m38           RELOCATE VW000 7000 1600
   370)  v1  m38           RELOCATE BROWN 7100 1300
   371)  v1  m38           RELOCATE MP600 6700 1200
   372)  v1  m38           RELOCATE VF700 6730 900
   373)  v1  m38           FOR VF700 SPEED 0
   374)  v1  m38           FOR VF700 REPLENISH 2000 FUEL
   375)  v1  m38           RELOCATE VF701 6715 900
```

Figure 9



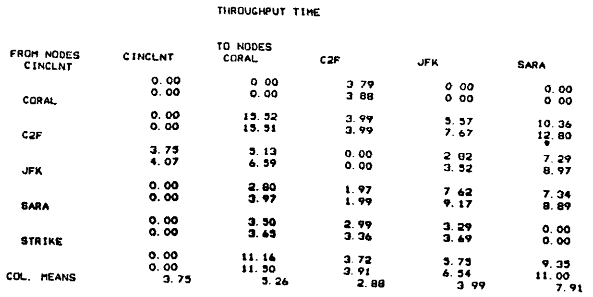
SAMPLE PREFORMATTED. DATAFILE

BLUE PLANNING MESSAGE
1. ORANGE OPTION (1) ____________________
(2) ____________________
(3) ____________________
ORANGE INTENT (1) ____________________
(2) ____________________
(3) ____________________
OTHER CONTINGENCIES ____________________
____
____
2. BLUE OPTIONS (1) ____________________
(2) ____________________
(3) ____________________
ASSESSMENTS OF OPTION (1) ____________________
____
OPTION (2) ____________________
____
OPTION (3) ____________________
____
3. BLUE PLAN (OFFENSIVE, DEFENSIVE, SELECTION OF PRESSURES OPTION)
BT



```
THIS FILES IDENTIFYING NUMBER IS:    12
THIS MESSAGE IS LABELED: INFO
SORT TIME IS: 11:25:31
ENTRY TIME: 11 25.26  SEND TIME: 11:25:31  ARRIVAL TIME: 11:25:57  READ TIME: 11.29 31
THROUGHPUT:     245 000  DESTINATION:    214.000  PREPARATION:       5 000
FROM: C2F         TO. CINCLNT


0106562

TO:    CINCLNT
FROM:  C2F


1.  FORWARDED.


0106552

TO:    STRIKE
FROM.  SARA

INFO:  C2F


SARA LENINGRAD STRIKE ACFT ON THEIR WAY
BT
```

Figure 10



THROUGHPUT TIME

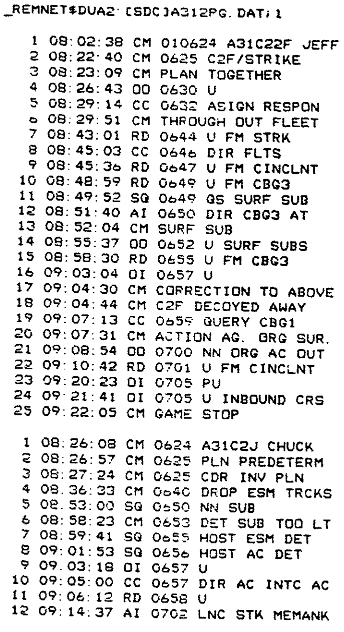
FROM NODES	CINCLNT	TO NODES CORAL	C2F	JFK	SARA
CINCLNT	0.00 0.00	0 00 0.00	3 79 3 88	0 00 0 00	0.00 0 00
CORAL	0.00 0.00	15.52 15.51	3.99 3.99	5.57 7.67	10.36 12.80
C2F	3.75 4.07	5.13 6.59	0.00 0.00	2 82 3.52	7.29 8.97
JFK	0.00 0.00	2.80 3.97	1.97 1.99	7 62 9.17	7.34 8.89
SARA	0.00 0.00	3.50 3.65	2.99 3.36	3.29 3.69	0.00 0.00
STRIKE	0.00 0.00	11.16 11.50	3.72 3.91	5.75 6.54	9.35 11.00
COL. MEANS	3.75	5.26	2.88	3 99	7.91

Table III. Observed Heat Measures

TITLE	DEFINITION	SCALINGS
Received Directive Quality (RDQ)	This measure scores the quality of the directive by whether or not it was understood, and also the action taken by the recipient if the directive was not understood	Not understood, not queried Not understood, queried Understood Incomplete, not queried Incomplete, queried
Surprises Queried (SQ)	This measure scores the action taken by the cell when surprised	Not understood, not queried Queried via status board Queried via talk
Action taken to Influence Orange (AIO)	This measure scores the attempts by BLUE cells to influence ORANGE action	No attempt Attempt
Contingency Coverage (CC)	This measure scores the contingency planning of each cell	Number of contingencies
Orange Options Understood (OOU)	This measure scores the BLUE understanding of the options available to ORANGE	Not understood Understood Partially understood
Orange Intent Understood (OIU)	This measure scores BLUE understanding of ORANGE intentions, or plans	Not understood Understood Partially understood
Blue Predictions (PR)	This measure scores whether or not BLUE cells predicted the outcomes of each alternative action developed	Predictions made/ Not made/Number

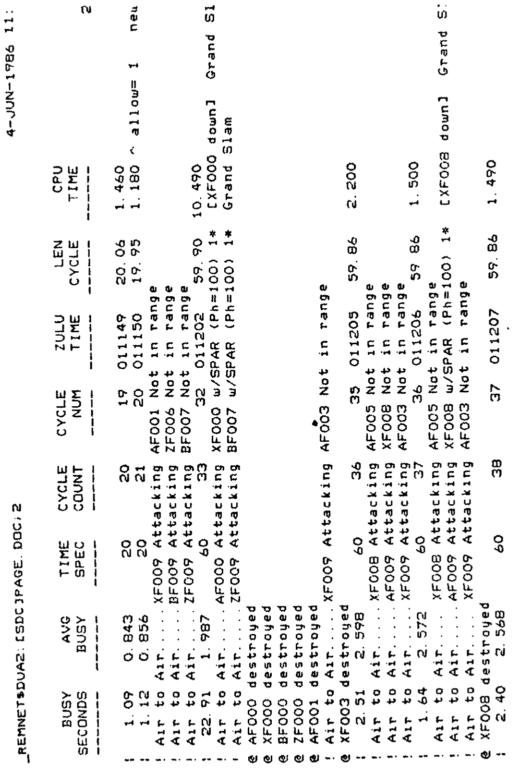
played out by the software and displayed on the game control terminal as shown in Figure 12. By analyzing these results such factors as missed opportunities, weapons expenditure rates, force exchange ratios and the effect of feints or information delays can be determined. The design team settled on 16 runs as providing a realistically sufficient experiment which ultimately produced 64 data packets for analysis using HEAT.

Figure 11



```
_REMNET$DUA2: [SDC]A312PG.DAT;1

  1 08:02:38 CM 010624 A31C22F JEFF
  2 08:22:40 CM 0625 C2F/STRIKE
  3 08:23:09 CM PLAN TOGETHER
  4 08:26:43 OO 0630 U
  5 08:29:14 CC 0632 ASIGN RESPON
  6 08:29:51 CM THROUGH OUT FLEET
  7 08:43:01 RD 0644 U FM STRK
  8 08:45:03 CC 0646 DIR FLTS
  9 08:45:36 RD 0647 U FM CINCLNT
 10 08:48:59 RD 0649 U FM CBG3
 11 08:49:52 SQ 0649 QS SURF SUB
 12 08:51:40 AI 0650 DIR CBG3 AT
 13 08:52:04 CM SURF SUB
 14 08:55:37 OO 0652 U SURF SUBS
 15 08:58:30 RD 0655 U FM CBG3
 16 09:03:04 OI 0657 U
 17 09:04:30 CM CORRECTION TO ABOVE
 18 09:04:44 CM C2F DECOYED AWAY
 19 09:07:13 CC 0659 QUERY CBG1
 20 09:07:31 CM ACTION AG. ORG SUR.
 21 09:08:54 OO 0700 NN ORG AC OUT
 22 09:10:42 RD 0701 U FM CINCLNT
 23 09:20:23 OI 0705 PU
 24 09:21:41 OI 0705 U INBOUND CRS
 25 09:22:05 CM GAME STOP

  1 08:26:08 CM 0624 A31C2J CHUCK
  2 08:26:57 CM 0625 PLN PREDETERM
  3 08:27:24 CM 0625 CDR INV PLN
  4 08.36:33 CM 0640 DROP ESM TRCKS
  5 08.53:00 SQ 0650 NN SUB
  6 08:58:23 CM 0653 DET SUB TOO LT
  7 08:59:41 SQ 0655 HOST ESM DET
  8 09:01:53 SQ 0656 HOST AC DET
  9 09.03:18 OI 0657 U
 10 09:05:00 CC 0657 DIR AC INTC AC
 11 09:06:12 RD 0658 U
 12 09:14:37 AI 0702 LNC STK MEMANK
```

Figure 12



```
_REMNET$DUA2:[SDC]PAGE.DOC;2                                                  4-JUN-1986 11:

   BUSY      AVG     TIME    CYCLE    CYCLE    ZULU     LEN      CPU
 SECONDS    BUSY     SPEC    COUNT     NUM     TIME    CYCLE     TIME                           2
 -------   ------   -----    -----    -----   ------   ------   ------

!   1.09    0.843     20       20        19   011149    20.06    1.460
:   1.12    0.856     20       21        20   011150    19.95    1.180 ^ allow= 1    neu
! Air to Air.....XF009 Attacking AF001 Not in range
! Air to Air.....BF009 Attacking ZF006 Not in range
! Air to Air.....ZF009 Attacking BF007 Not in range
:  22.91    1.987     60       33        32   011202    59.90   10.490
! Air to Air.....AF000 Attacking XF000 w/SPAR (Ph=100) 1*  [XF000 down]  Grand Sl
' Air to Air.....ZF009 Attacking BF007 w/SPAR (Ph=100) 1*  Grand Slam
@ AF000 destroyed
@ XF000 destroyed
@ BF000 destroyed
@ ZF000 destroyed
@ AF001 destroyed
! Air to Air.....XF009 Attacking AF003 Not in range
@ XF003 destroyed
:   2.51    2.598     60       36        35   011205    59.86    2.200
! Air to Air.....XF008 Attacking AF005 Not in range
! Air to Air.....AF009 Attacking XF008 Not in range
! Air to Air.....XF009 Attacking AF003 Not in range
:   1.64    2.572     60       37        36   011206    59 86    1.500
! Air to Air.... XF008 Attacking AF005 Not in range
! Air to Air.....AF009 Attacking XF008 w/SPAR (Ph=100) 1*  [XF008 down]  Grand S:
! Air to Air.... XF009 Attacking AF003 Not in range
@ XF008 destroyed
:   2.40    2.568     60       38        37   011207    59.86    1.490
```

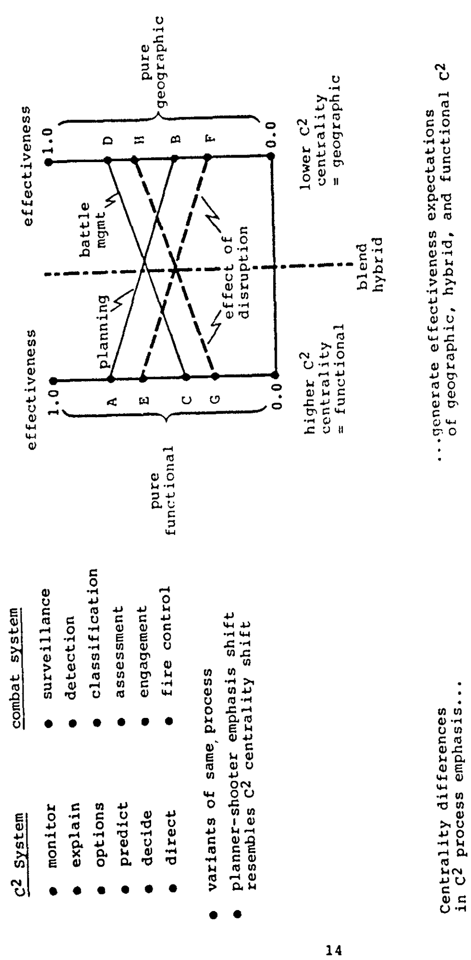

C^2 System
• monitor
• explain
• options
• predict
• decide
• direct
combat system
• surveillance
• detection
• classification
• assessment
• engagement
• fire control
• variants of same process
• planner-shooter emphasis shift resembles C^2 centrality shift
effectiveness
1.0
A
E
C
G
0.0
pure functional
planning
battle mgmt
effect of disruption
effectiveness
1.0
D
H
B
F
0.0
pure geographic
higher C^2 centrality = functional
blend hybrid
lower C^2 centrality = geographic
Centrality differences in C^2 process emphasis...
...generate effectiveness expectations of geographic, hybrid, and functional C^2
14

Figure 13. Logic of C2 Organizational Experiments

Results of the experiments have been analyzed in four forms which are discussed below.

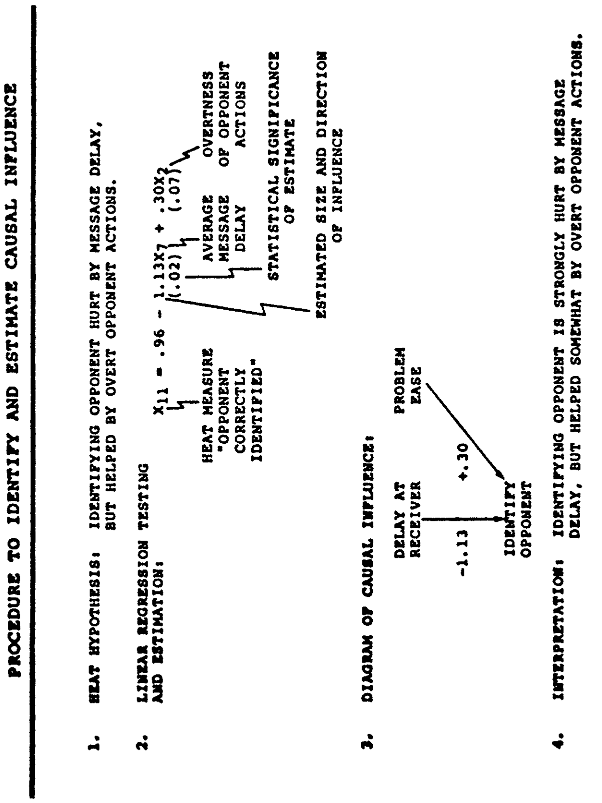
Within the experimental design, a number of replications, for example, 18 in the geographic design shown in Figure 5, are obtained. The hypothesized results of these cases can be expressed by a figure such as Figure 13 which shows the relative overall combat effectiveness as measured by the exchange ratio of enemy losses to friendly losses. For example, the geographic organization in non-complex scenarios should do best. The results from the applications with these properties can be averaged and the relative ranking against other combinations of scenario and organization can be determined.

Crossovers such as shown where the functional organization deteriorates more under jamming, are particularly interesting. Most of the hypotheses made have been confirmed as shown in Table III.

Another type of analysis has been to prepare a matrix of regression relations for each experiment and even across the experiments. In this approach the data for each case is set up as a row in a matrix and statistical relationships are extracted across the cases. For example, Figure 14 shows the relationship between correct identification of the opponent and the dependent variables message delay and overtures of opponent action. The statistics are reasonable and the signs are correct.

DSI has represented the same regression information can be shown as an influence diagram with the coefficients as shown in the bottom of Figure 14. When the entire matrix of regression results is displayed as in Figure 15, a complex set of relationships can be captured which include the interdependencies of scenario, traffic and headquarters performance on the overall combat performance measure, the exchange ratio.

Figure 14



PROCEDURE TO IDENTIFY AND ESTIMATE CAUSAL INFLUENCE

1. HEAT HYPOTHESIS: IDENTIFYING OPPONENT HURT BY MESSAGE DELAY, BUT HELPED BY OVERT OPPONENT ACTIONS.

2. LINEAR REGRESSION TESTING AND ESTIMATION:

$$X_{11} = .96 - \underset{(.02)}{1.13}X_7 + \underset{(.07)}{.30}X_2$$

HEAT MEASURE "OPPONENT CORRECTLY IDENTIFIED"
AVERAGE MESSAGE DELAY
OVERTNESS OF OPPONENT ACTIONS
STATISTICAL SIGNIFICANCE OF ESTIMATE
ESTIMATED SIZE AND DIRECTION OF INFLUENCE

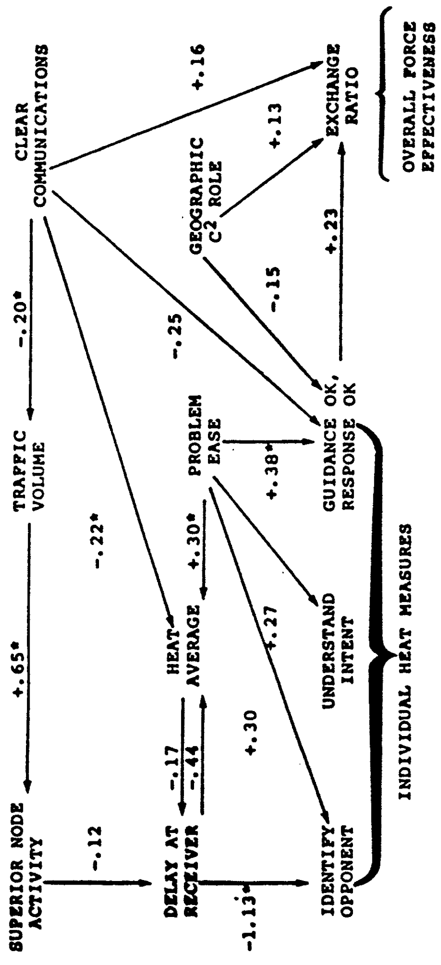
3. DIAGRAM OF CAUSAL INFLUENCE:

DELAY AT RECEIVER
PROBLEM EASE
-1.13
+.30
IDENTIFY OPPONENT

4. INTERPRETATION: IDENTIFYING OPPONENT IS STRONGLY HURT BY MESSAGE DELAY, BUT HELPED SOMEWHAT BY OVERT OPPONENT ACTIONS.

Figure 15

DIAGRAM OF EFFECTS SHOWING ESTIMATED CAUSAL INFLUENCES AMONG MILITARY PROBLEM, C2 TRAFFIC, NETWORK, PROCESS QUALITY, AND OVERALL FORCE EFFECTIVENESS



Conclusion

These large-scale trials show that substantial conclusions can be drawn from realistic decision-making experiments in command, control and communications that are controlled by an experimental design.

References

1. Defense Systems Inc., 1985 C2 Effectiveness Experiments, May 1986, MacLean, Virginia 22102.

2. Durzhonen, V.V., Concept Algorithm, Decision, Moscow 1972.

3. Sovereign, M.G. and Stewart, J.S., Assessing the Organizational Responsibility of Headquarters Under Differing Levels of Stress, 8th Annual ONR-MIT Workshop on C3 Systems, 1985, p. 49.

THE DEFINITION, IMPLEMENTATION, AND CONTROL OF AGENTS IN AN INTERVIEW SYSTEM FOR DISTRIBUTED TACTICAL DECISION MAKING

Jeffrey M. Gilbert and Robert L. Stewart

The Johns Hopkins University
Applied Physics Laboratory

ABSTRACT

In the course of conducting research into the cognitive processes associated with distributed tactical decision making (DTDM), we are developing an experimental environment for interviewing tactical experts and for analyzing the processes involved in distributed problem-solving behavior. This environment, or interview system, is being developed on the basis of a theoretical framework that includes both a taxonomy for DTDM processes and a composite cognitive model of those processes. As a part of this framework agents and agent protocols are used to represent selected elemental entities and interactions of distributed decision making. It is anticipated that this abstraction will assist in decomposing observed DTDM behavior into its component processes and in classifying and analyzing those processes with respect to the theoretical framework.

This paper focuses on the concept of agents and agent protocols as it is employed in our research. It first briefly describes the theoretical framework, the experimental environment and how the protocols/agents concept fits within that structure. Next it discusses the process whereby agents and agent protocols are identified, classified and formally defined. The paper goes on to describe the choices made in implementing agents and protocols on a computer-based subsystem of the interview and analysis system. The paper continues by discussing the top-level architecture and control for the agent subsystem in terms of the primary elements, or 'flavors', that it uses. Finally, it lists directions for current and planned work in developing agents and the agent subsystem as a part of the continued evolution of the entire interview and analysis system.

INTRODUCTION

During the last two years we have been conducting research into the issues underlying distributed tactical decision making (DTDM). The subject of our study has been distributed decision making in the Composite Warfare Command (CWC) paradigm. In particular, each of the experienced decision-making subjects in our study performs the role of one of the three Warfare Area Commanders, namely Anti-Air, Anti-Surface, and Anti-Submarine Warfare Area Commanders (AAWC, ASUWC, and ASWC, respectively). Aspects and variables of the decision-making process for this domain area are then identified and studied for each set of subjects by analyzing the goals, plans, and interactions among the three decision makers.

In conducting our investigation into DTDM processes, we have constructed a theoretical framework on which to base our experimental work [Hamill and Stewart 1986]. This framework comprises both a taxonomy whereby processes may be classified and a composite model for processes that is closely aligned with this taxonomy. Classification within this framework is divided along three major processing dimensions: psychological, computational, and communicative.

Our theoretical model for DTDM processes is a composite whose three main constituents, respectively, fall along the three major processing dimensions of the taxonomy. Along the psychological dimension, the model adopts the human information processor model as espoused by Rasmussen [Rasmussen 1982] [Rasmussen 1983]. Along the computational dimension, it incorporates the Year 2000 C^3 process model developed at APL [Halushynsky and Beam 1984] [Lawson 1984]. Finally, along the communicative dimension, the model employs the Open Systems Interconnect (OSI) model that was developed by the International Standards Organization [Tanenbaum 1981]. Further elaboration on this framework - the cognitive process taxonomy, the composite model, and the justification for choices made in their development - is available elsewhere [Hamill and Stewart 1986] and will not be included here.

A theoretical concept useful in the analysis of the interactions between distributed decision makers is that of protocols. We view a protocol as a generalization encompassing the characterization of human problem-solving activities proposed by [Newell and Simon 1972] as well as communications protocols defined by the OSI process model. The concept of a protocol is in this sense intended to model interactions among decision-making entities that may occur along any of the major processing dimensions and at any given level of abstraction. It is anticipated that modeling interactions in terms of the protocols they embody will permit us to decompose distributed problem-solving and communications activities. In this way we hope to isolate, organize, and classify those processes that are occurring in DTDM.

In this theoretical framework the entities that actually execute according to such protocols are represented by agents. By an agent we mean any person or mechanism that is empowered by a guiding intelligence to act to achieve some goal or result. Agents, then, are capable of (if not authorized for) independent decisions and possibly independent actions. They thus serve in our model as intelligent or quasi-intelligent entities that execute various types of DTDM processes at various levels of processing abstraction.

In this view, agents are intended to represent any type of functional decision-making unit. They may therefore be human, machine, or possibly a combination of both. Furthermore, for any set of agents operating on a given processing problem at a given level of abstraction, the functional information associated with each agent implicitly defines a sequence of knowledge states that will occur and how they will occur. This sequence of knowledge states associated with a an open system of peer-level agents is what we mean by a protocol. Of course, protocols could be explored for any set of agents that require interaction. But since the primary goal of our work is to identify and illuminate those protocols relevant to distributed decision making as well as to identify the knowledge and communication requirements for associated interactions, agents and their protocols become most significant to this research when they involve multiple tactical decision nodes. Hence, our experimental environment, though capable of supporting protocol analysis for arbitrary sets of interacting agents, focuses on those processes involving multiple DTDM nodes.

In support of the entire DTDM interview, test, and analysis process, an experimental environment, or interview system, is being developed in accordance with the theoretical framework. The interview system includes both manual elements and computer-based elements. Its design supports experimentation with multiple decision-making nodes.

The interview system is intended to serve many purposes in the experimental phase of this research. It provides means for identifying the required elements for DTDM scenario and plan generation. Its architecture includes mechanisms for developing and testing the plans, protocols, and agents of subjects experienced in distributed tactical decision making. In operation, it assists in the identification, instantiation and exercise of agents and protocols at various DTDM processing levels, and it helps to identify the knowledge and communication requirements for multi-node agent interaction.

The remainder of this paper will concentrate on one part of the interview system, namely the agent subsystem. It will discuss several aspects of this subsystem and how its development has proceeded in accordance with our theoretical framework. First, the agent definition process will be outlined along with the objectives and rationale for that process. Next, the computer implementation of the agent subsystem and some justifications for choices made in that implementation will be described. In particular, some of the features of object-oriented and rule-based programming will be considered in terms of their relation to the agents and protocols we are trying to model. Finally, the top level of control and the primary 'flavors' of objects used to implement the control architecture of the agent subsystem will be presented.

AGENT DEFINITION

As indicated above, the purpose of the protocols/agents concept in our model is to provide a window into the interactive processes occurring in distributed tactical decision making. Decomposing the problem-solving and planning activities of our subjects in terms of protocols and agents will assist in the analysis of those activities. This decomposition will then help to illuminate knowledge acquisition and representation issues and to identify and focus on those aspects of our subjects' activities that are specifically relevant to multi-node decision making.

The process we have developed to identify and define agents has been designed with these goals in mind and in accordance with the theoretical framework we have constructed. One objective for agent definition is thus to identify and represent specific inter-node interactions at specific abstraction and processing levels. Identification will proceed with an examination of the protocols observed in our subjects' distributed decision-making behavior as they interact in simulated tactical environments. The agents thus identified and the protocols they employ will constitute a procedural knowledgebase and a representation of communication requirements for the observed interactions among DTDM nodes.

Breaking down the agent definition process into its components, the first step is a protocol analysis of observed subject interactions. This analysis attempts to separate and formalize those protocols and classify them according to our taxonomy. For example, consider two decision-making nodes, an AAWC, whose goal may be to optimize AAW coverage, and an ASWC, whose goal may be to optimize ASW detection capacity. Assume that in the course of the scenario in which they are operating a platform that has been performing both AAW and ASW roles for the battle group is lost or is otherwise disabled. Reorientation of the battle group formation may be required by either or both of the warfare area commanders. Both will typically have access to an active plan for the entire force. However, their respective views of this plan, of the means for its execution, and of the situation itself may differ. Furthermore, the primary subgoals faced by the AAWC and the ASWC are themselves different. Thus one could expect that the reorientations which they might recommend for a specific set of conditions may also differ.

Assuming no reorientation has previously been specified for this set of conditions, in order for the reorientation to proceed other than by doctrine, some interaction between AAWC and ASWC will, ostensibly, be necessary. The interaction may be via direct verbal communication between the two warfare area commanders or it may be through other agents of the commanders. The interaction may even be implicit in the orders given by those commanders. It is expected that the interaction will expose protocols at several levels of processing and abstraction. In any case, the goal of the first phase of our analysis is to identify, observe, and classify all such protocols occurring during internode interaction.

Once the protocols for a particular situation have been separated and formalized according to our model, the next phase of the agent definition process is to attempt to create formal definitions for those agents that are required to account for the observed protocols. The formal definition for an agent will include both knowledge requirements and information flow requirements (such as time limits, minimum required data, required types of data, connectivity, etc). It will also incorporate the functional description of the agent needed to represent the observed protocol.

Returning to the above example, if a platform had been assigned primary responsibility for a certain sector of AAW coverage, then one would expect the loss of this platform to cause the AAWC or one of his agents to call for a reassignment of AAW responsibilities and perhaps a reorientation of force assets. This requirement, if it affects the ASWC, will typically be reflected in the interactions between the two area commanders, or between their agents, and will be observed in the in the course of our interviews. For example, a simple but important protocol/agent relevant to the previous hypothetical situation involves recognition, ultimately on the part of the AAWC, that primary AAW coverage has been lost for a sector. This recognition will in turn trigger an AAW goal to compensate somehow for the lost coverage. A similar mechanism will likely operate for the ASWC, recognizing when ASW objectives are not being met and triggering necessary action to meet those objectives.

In phase one of agent definition, our protocol analysis of the interaction between AAWC and ASWC or their agents would thus identify this recognize/trigger protocol. Concurrently, it would be analyzed and classified (perhaps primarily as a rule-based computational process).

In the second phase this protocol would be represented formally in terms of agents whose functions are to monitor critical AAW and ASW conditions and to alert the AAWC and ASWC, respectively, when those conditions fail to be met. In this simple example, agent functions may occur at a rule-based level such that they would be easily implemented with a production system directly on computer. Certainly, this will not always be the case. In particular, as the the level of processing abstraction becomes higher one would expect the computer implementation of agents to become correspondingly more complex, requiring human collaboration or human control.

The third phase of agent definition is to move from the formalized agent description that has been constructed (including agent functionality) to a computer-based implementation of those protocols/agents considered feasible for machine representation, and to identify those agent processes not amenable or appropriate for computer representation. Computer instantiation of DTDM agents is designed to be flexible so that the features of the agents can be directly reflected in that instantiation. Thus agent instantiations may employ rule sets or other formalisms to represent heuristic knowledge. Algorithmic knowledge could be directly coded in procedural form. In any case the architecture (discussed below) for controlling agents is intended to provide appropriate mechanisms for specifying connectivity, permitting interactions, and monitoring those interactions regardless of the internal representation for the agents involved. Hence language requirements and implementation choices are driven by the theoretical and representational needs for protocols and agents rather than vice versa.

The end product of the agent definition process is thus a set of relevant DTDM agents for each node and the protocols they embody as represented in their functionality. To as great an extent as possible this process attempts to formalize agent definitions to a level amenable for computer instantiation. Representation with this level of rigor serves two purposes. First it allows our model to automate certain relevant protocols/agents and thus may permit convenient simulation and testing of the dynamic aspects of DTDM interaction. Just as importantly, it enforces intimate attention to details of the observed behavior that might otherwise be missed.

For all selected agents chosen to be represented in the computer-based interview system, the final phase of definition is validation and testing. In this phase, the computer representations are exercised in the presence of their authorizing subjects and stressed under tactical simulations in order to determine whether they achieve the subjects' requirements. For numerous reasons the agents as implemented may not meet those requirements: The protocols observed during subject interactions may not completely encompass the DTDM needs of the subjects. The subjects themselves may not have foreseen some necessary aspects of the agents they employ. Or the experimenters may have made errors in their formalization and instantiation of the protocols/agents. Thus the agents defined may not function as expected or as required.

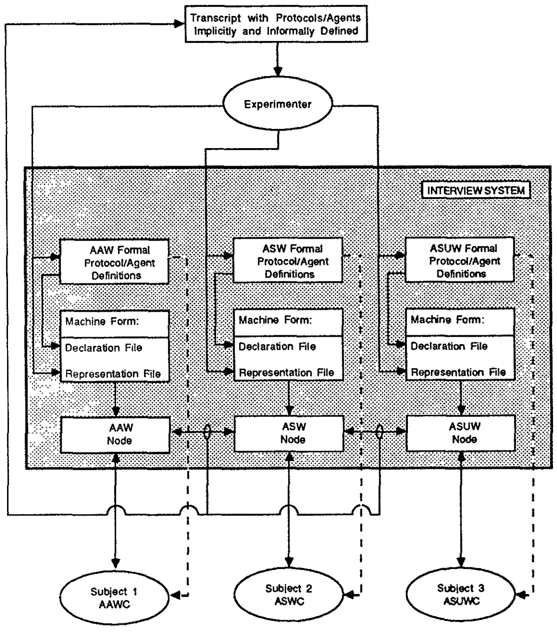
Should it be determined that certain needs are not being met by agents as they have been instantiated, new subject interaction specific to those needs is initiated. Thus begins a new iteration of the agent definition process that concentrates on those requirements that failed to be achieved on the previous iteration and on the protocols/agents that are necessary to meet those needs. Iteration continues in this way, converging on the specific features needed to meet the requirements of all DTDM subjects.

In summary, the definition process is illustrated in diagram 1. As shown, a transcript of the interactions among our subjects implicitly and informally describes the protocols/agents they employ. From this description the experimenters perform a protocol analysis to identify and classify protocols observed and to formalize agent descriptions (including functionality, connectivity, communication requirements, etc). Depending on the type of agent being defined, this formalized description may in part be fed back to the subjects for preliminary confirmation (dotted line).

Next, where possible, the formalized agent description is translated into machine-readable form. The translation produces two files that together define the agent. The Declaration File contains connectivity and communications requirements for the agent as well as administrative data for use by the control mechanisms to assist in agent execution. The Representation File contains any code that is necessary to represent the agent's functionality. This may, as mentioned above, include coding in procedural form, in rule-based form, or in whatever representation is most appropriate for that agent.

Finally, the computer-based agent control subsystem reads the declaration file and representation file for each agent. The agents thus installed are then tested and validated in the presence of the DTDM subjects so that discrepancies between actual and required behavior become

Diagram 1: Agent Definition



apparent. Such discrepancies are then corrected by iterating this process until DTDM requirements for each of the subjects' agents are adequately met.

IMPLEMENTATION OF COMPUTER-BASED ELEMENT OF AGENT SUBSYSTEM

As stated in the preceding section, the choices made with regard to computer implementation of the agent subsystem have been driven by the theoretical and representational requirements for protocols/agents. First of all, agents should be distinct, semi-autonomous decision-making entities. The implementation should reflect this distributed nature of agents. In addition, since according to our framework agents operate quasi-intelligently, their implementation must be capable of incorporating independent procedural information. Similarly, it must be capable of maintaining independent data structures. Heuristic knowledge may well be required of the agents, and hence it is also be important to have forward-chaining rule-based programming capabilities available for their implementation (as this mode of representation often proves a convenient tactic for maintaining such knowledge). Finally, it is essential for the experimenter to have convenient mechanisms for controlling and monitoring inter-agent communications.

Reviewing object-oriented programming one notes that it has several qualities that make it a particularly attractive paradigm through which to model agents. First, objects are logically distinct entities that are capable of semi-autonomous operation. This was mentioned above as a fundamental requirement for agents. In addition, they provide for the encapsulation and maintenance of data structures needed by agents (in the form of object variables) as well as required procedural information to act on those structures (in the form of procedures called methods). Finally, the simplified interfaces and loose coupling between objects that is afforded by the message-passing mechanism equips object-oriented systems with a convenient means for implementing and representing agent interaction. For these reasons an object-oriented programming approach was chosen for implementing protocols/agents.

As also mentioned, in many cases it is desirable to incorporate decision-making heuristics in the computer-based implementation of an agent. Thus, returning to the example protocol discussed above, an agent whose function is to alert the AAWC of a possible loss of AAW coverage will likely incorporate some rule-of-thumb such as 'If a critical AAW resource becomes unavailable, then consider whether battle force reorientation would be appropriate.' In this instance, as in many others, the heuristic is conveniently stated in the form 'if <condition> then <action>.' Commonly, there will be many such rules, each of which represents a particular heuristic that the agent may use in making a decision or taking an action. If the agent has been properly classified with our taxonomy, the entities and actions that occur in the rules (e.g., 'critical AAW resource' and 'battle force reorientation') will tend to fall within a fairly specialized realm - in this case the elements necessary for planning and controlling AAW coverage. Under such circumstances, rule-based programming often provides an appropriate means for representing and implementing such heuristics. In fact, it is for just such situations, implementation of heuristic rules for operation in a focused problem domain, that production systems tend to be most fruitful. This is certainly not to say that every heuristic and every domain of knowledge is amenable to production system representation. However, where heuristic agent functionality *is* conveniently represented in terms of rule sets, a rule-based programming capability becomes a useful tool.

For a number of reasons we chose Maryland's Franz Lisp/Flavors/YAPS as an environment in which to develop the agent subsystem. A primary factor influencing that decision was the fact that this programming environment was the only one we examined that combines object-oriented programming features, by way of a programming system known as Flavors, with a rule-based programming capability, in a system called YAPS. In fact, since YAPS -- which stands for 'Yet Another Production System' -- is written in terms of Flavors and Flavor objects, it also furnishes the user with many of the powerful object-oriented features of the system, thus allowing considerable programming flexibility and power. For example, YAPS rulebases are themselves objects. One may therefore have many independent databases and rule sets operating in the same environment. They may send messages to invoke one anothers' methods. And information detailing the operation of one rulebase may be hidden from other rulebases. In short, every feature provided to an object is available to a rulebase since it is itself just a particular flavor of object. Franz Lisp/Flavors/YAPS equips the experimenter with the major features required to implement protocols/agents and it incorporates those features in a logical and integrated manner that permits both flexibility and programming power (see [Allen, Trigg and Wood 1983] and [Allen 1983] for additional information on Flavors and YAPS, respectively).

AGENT SUBSYSTEM CONTROL AND ARCHITECTURE

This section outlines the elements developed in Flavors to control agent execution and communication in our computer implementation. As has already been stated, Flavors is a powerful system utilizing objects and the object-oriented programming metaphor. A flavor is itself not an object but rather a template for an object. In other words a flavor is essentially a description for a class of objects, i.e., the 'instances' of that flavor. The flavor describes those data structures that are to belong to each of its instances as well as the methods, or procedures, which they will have available. Thus each instance of a flavor has its own copies of the structures. And when the instance receives a message for which the flavor has a method defined, it executes that method using its own copies of the data structures.

Our agent subsystem currently employs the flavors *Control, AgentRB, Comm,* and *Calendar.* Together these flavors and some auxiliary Lisp functions comprise the control and architecture of the computer-based agent subsystem. *Control* and *Comm* have been developed for operation within a single UNIX Franz Lisp process. However, they are currently being extended to operate in multiple processes and multiple machines, an important goal for our computer implementation. In the following the purpose and basic operation of each of the primary flavors is explained.

THE *CONTROL* FLAVOR

Control is a flavor designed to encapsulate the structures and functions which direct agents to execute, to keep track of the agents that are currently known to the system, and to determine which of those agents are ready to execute. This flavor, then, implements the top level of control for agent execution in the subsystem. Only one instance of the *Control* flavor is needed in any one UNIX process and this instance, because it is unique, has been named *Control* as well.

In order to direct agent execution, *Control* maintains two data structures, also called instance variables. The first is bound to the list of all agents currently known to *Control.* The second is bound to the list of all agents that are currently ready to execute.

In addition to these instance variables, *Control* has a number of associated methods which maintain the variables and which perform the top level of agent control. The most important of these directs agents to run. As long as there are agents ready and waiting to be run, it orders each of those agents to execute in turn. In addition, through each pass of this top-level loop, *Control* polls the *Calendar* (discussed below) to see whether there are any previously scheduled actions (events) that are now ready to take place. If so, these actions are also executed before the next iteration of the control cycle.

In order to execute agents properly *Control* places some minimal requirements on any executable agent flavor. First of all, when a given agent is created and initialized, the new agent should declare itself to the system. In addition, it is the responsibility of the agent to determine when and if it is ready to be activated and to inform *Control* accordingly. Likewise, when an agent completes any actions it had to perform, it must inform *Control* that it has depleted. Finally, in order for *Control* to direct agents to act they must accept a message to 'run', interpreted as a directive for them to execute.

Agents may be defined in terms of any flavors that conform with the above constraints. Therefore, the control system provides a great deal of flexibility in defining agent flavors and objects and the experimenter may choose whichever representations and facilities he deems appropriate when implementing a given agent. The following section gives an example of a particular agent flavor that has been developed as one such possible, useful representation using YAPS.

AN AGENT FLAVOR

As stated before, an important facet of YAPS is that it is itself written in terms of Flavors and flavor objects. A consequence of this design is that it is not difficult to create multiple rulebases, each with its own working memory and rule set, all operating within a single Franz Lisp process. Indeed, since the *Control* flavor invokes minimal constraints upon the flavors which may be defined as agents, one may define a rule-based agent flavor incorporating the YAPS production system without much difficulty. For reasons discussed earlier, production systems provide convenient representation for certain forms of heuristic knowledge and are thus are desirable in the implementation of some agents.

In order to define a rule-based agent flavor, it was convenient to use a mechanism built into Flavors called inheritance. A new flavor called *AgentRB* was defined and it was given the component flavor called *yaps-database* that describes the features of a YAPS rulebase. By including this component flavor in its definition, the new flavor inherited all instance variables and methods that make up a YAPS rulebase. These could be used under the new flavor just as though one were dealing with a generic YAPS database and rule set. The purpose of the new flavor was to allow additional structure to be placed on generic YAPS permitting rulebases to interface properly with *Control.*

When the system creates a new instance of the *AgentRB* flavor according to an agent definition provided in declaration and representation files, as part of its initialization the new instance sends *Control* a message indicating that it should be installed on the agent subsystem. In addition, a set of rules to be used by this particular agent is read from the representation file that was developed for the agent (see section above on agent definition). Finally, the database is cleared and initialized as required for the function of the particular agent being implemented.

The *yaps-database* flavor describing YAPS rulebases already accepts the 'run' message. Upon receiving this message, the system executes until no rules match, at which time control returns to the point of invocation. Since *AgentRB* inherits this method from the *yaps-database* flavor, it is not necessary to define a run method explicitly for the new flavor.

On the other hand, generic YAPS does not automatically monitor a rulebase and and inform *Control* when it is ready to be activated. Whenever a change is made to the working memory of a particular agent (e.g., whenever a fact, goal, or message is added or deleted) we would like the agent to check whether the change results in a new match on any of its rules.

The methods that change working memory by adding and removing elements are also a part of generic YAPS and were inherited from the *yaps-database* flavor. To monitor changes one could opt to modify the code for these built-in methods. After the main body of code for such a method, one could add the code needed to check whether there are any matches and to inform *Control* accordingly. But since we are not really changing the main task for these methods (which is still to install or remove working memory elements) a more convenient and more modular option would be to use a mechanism called a daemon.

In Flavors a daemon is a special method that is 'attached' to a regular (primary) method and that executes immediately before or immediately after the primary method. These methods are called 'before daemons' and 'after daemons,' respectively. Thus whenever an object receives a message, any before daemons that were defined for the message execute first, then the primary method executes, and finally any after daemons for the message execute.

In our case we simply attach after daemons to each method that might modify the set of rules that match. These daemons then automatically call a function that checks for newly created or lost matches immediately after any such change occurs and informs *Control* of the ready status of the agent. Not only is this a safer and more convenient alternative to modifying the code in the *yaps-database* flavor methods, but it also conforms to the doctrine of modularity and reuse of code.

The methods just discussed for the *AgentRB* flavor, together with a few auxiliary functions, provide the glue needed to incorporate the YAPS production system as a form of an agent representation to be used in the agent subsystem. Of course, the methods and data structures that encompass YAPS were already available and thus the new flavor simply had to interface YAPS rulebases with the agent subsystem control. Nonetheless, this example should serve to illustrate the flexibility afforded in the computer implementation of agent control as well as that provided by the Flavors environment itself.

THE *COMM* FLAVOR

As has already been pointed out, an important facet of distributed decision making is communication between agents at distinct decision-making nodes. Originally, our system permitted any agent to send a message directly to any other agent at any time. However, the need for experimental access to messages between agents at separate DTDM nodes led us to structure inter-agent communications. In particular the system must allow the experimenter to monitor messages being passed between agents. It must also enable him to control and study the impact of changes in various aspects of the agents' communication channels. Such aspects include: 1) which channels or networks are available to a particular agent (e.g., agent connectivity), 2) operational status and availability of a given channel, 3) channel fidelity (bandwidth, level of noise, etc.), and 4) communications delays (as a function of bandwidth, queuing, etc.). Additional aspects may be identified and included in this list as needs for them are determined.

In order to centralize and encapsulate the mechanisms used to structure inter-agent communication we defined a new flavor called *Comm.* Like *Control,* there is just one *Comm* object per UNIX process, and that object is also given the name *Comm.* The *Comm* flavor has been designed to provide the agent subsystem with two important services. First it simulates the set of networks or channels connecting the various agents known to the system. And second, it allows the experimenter access and control over the aspects of these channels as just listed. Currently, *Comm* contains the skeleton needed for simulating a multi-network multi-decision-node system of agent communication. Many of the detailed capabilities such as simulation of channel noise and communication delays and experimental control of these features are still under development. However, we expect that the modular nature of Flavors and flavor objects will permit such features to be added or modified with relative ease.

In addition to the experimental requirements for our simulation of agent communication, there are important benefits with respect to implementation that are derived from distilling, centralizing, and encapsulating communications work. For one, a significant amount of common work is done in preparing messages to be sent, regardless of source and destination. By extracting this overhead from the agents and incorporating it within *Comm,* coding of the agents may be simplified considerably. Thus much of the formatting, sequencing, etc., of messages may now be handled automatically by the new *Comm* flavor object rather than by an agent itself. Also, since the agents now contain less code that is specific to the communications process, one may modify features or implementation of the communication simulation with minimal or no changes to individual agents. Finally, we are now adding interprocess and intermachine communication to the system and we wish to keep physical location transparent to the agents themselves. As the interview system is extended to operate in a multi-process, multi-machine environment, it will be useful to retain a common mechanism for agent communication. Because control of agent communication for each Franz Lisp process is modular and centralized in a *Comm* object for that process, new methods for interprocess communication may be added to the system's capabilities without significantly affecting the message-passing methods used by agents. In this way we are permitting agents to communicate with one another but at the same time are keeping the details of their physical location (e.g., on which process or machine they are running) as transparent as possible.

Together, the methods and instance variables of *Comm* provide convenient and automatic mechanisms for experimental monitoring, control, and coordination of agent communication. By centralizing this control in a flavor object, significant overhead is saved in generating and sending messages. And because of the modularity inherent in flavor objects this management can remain for the most part independent of and transparent to the agents that actually send the messages.

THE *CALENDAR* FLAVOR

In some instances it is necessary for the agent subsystem to perform time-deferred and time-dependent actions. In other words, certain actions in the agent subsystem must execute not immediately, but either at a specific time or after some specified time delay. One example requiring this capability was mentioned in the previous section, namely, the simulation of communications delays in passing messages. Another example is a situation in which an agent requests a response from another agent before taking some action. If no such response is elicited within some specified time frame, then the first agent would proceed either by taking the action as originally planned, by taking some other action, or by taking no action -- in accordance with his authority, judgement, and the situation. Still another example arises in synchronization of actions between decision nodes.

The last primary flavor covered in this paper, *Calendar,* provides the agent subsystem with a means of controlling and implementing such deferred actions and time-dependent events. Each Lisp process has one *Calendar* flavor object (also called *Calendar*) which keeps track of all actions that are to be performed at future times. Essentially, the *Calendar* encapsulates an event queue and the methods needed to operate that queue. Any agent, in fact any entity residing in the Lisp process, may then utilize the *Calendar* to save events or actions that are to be executed at a later time.

The event queue itself is simply a list of times and associated events sorted by increasing time. Each time *Calendar* is polled in the top-level control cycle, the event queue is checked against the current time of day to determine whether any events are ready to execute. All such actions are performed until *Calendar* determines that there are no more events scheduled up to the current time. Times are measured in seconds from midnight on the current day so that a real time of day can be reflected in event execution. The actions permitted include any evaluable Lisp expression, including messages to flavor objects such as adding and removing facts or goals from YAPS databases. Thus *Calendar* provides a flexible mechanism for executing time-dependent events and deferred actions based on a real clock time of day.

In addition to the top level method for running the *Calendar,* the flavor contains the obviously necessary methods for initializing the event queue and for scheduling new events on the *Calendar.* Utility methods are also associated with the flavor to examine the status of the *Calendar* from outside. For example, one method determines whether the *Calendar* currently has any events ready to fire. Another returns the time at which the next event is scheduled. Still another returns the next event itself.

One final consideration must be expressed with regard to the *Calendar* and its use by agents in the agent subsystem. Since our Lisp environment does not provide for concurrent programming within one process, the *Calendar* must be polled regularly in the top-level loop of *Control.* Thus if the main body of the loop takes an appreciable amount of time to complete, then scheduled events will not execute precisely on time. However, as long as the time delays associated with deferred events are long (say, on the order of minutes) in comparison to the execution time of the loop body, we anticipate this to be no major problem.

CURRENT AND FUTURE WORK

Having described the theoretical background for the agents/protocol concept and the definition and implementation of agents within the agent subsystem, we conclude this report by listing some of the directions being taken as we continue to develop the subsystem in conjunction with other elements of the DTDM interview system. As suggested earlier, in order to properly analyze distributed decision-making processes it is essential that the computer-based metaphor representing them reflect the concurrency such processes involve. Only in this way, we feel, will the experimental environment truly allow us to capture and focus upon those features of distributed problem solving behavior that arise from the concurrent operation of multiple agents in distinct decision-making nodes. For this reason a primary goal for the agent subsystem is the extension of the architecture and control first to multiple Franz Lisp processes and ultimately to multiple hosts.

A first step in reaching this goal is the development of an interprocess communications (IPC) capability for separate Lisp processes. Several approaches were considered. However, we chose to pursue this objective using UNIX sockets, since they are the fundamental mechanisms for IPC under Berkley's 4.2 BSD and also since they are convenient for interhost communications.

The design outlined in the previous section has now been expanded to implement a distributed agents environment across a computer network. The present environment consist of several computers (Microvax II's, Sun workstations, and a Pyramid 98X) connected by an ethernet local area network. The distributed environment, developed in the C and LISP languages, allows several YAPS processes, with one or more agents running on each process, to send application-level messages to any other local or remote agent. All the hosts computers in the network run UNIX 4.2 BSD and the TCP/IP communication protocol. The communication channels are established over a reliable, bidirectional, sequenced, and unduplicated flow of data without record boundaries.

To allow communication among agents throughout the network, a special process is activated at the time the first YAPS process is activated. This process, the CLERK, keeps an active blackboard of the addresses for all the YAPS processes and agents present in the network that have a communication channel opened to receive messages from other processes. There is only one CLERK process active in the network, and all other processes have methods defined to communicate with CLERK when they need to send messages to agents with whom they have not previously communicated. When an agent is instantiated on any host computer it sends a message to the CLERK process defining its address.

Local and remote agents can communicate in a way transparent to the user of the agent system. Messages between two agents are handled by an intermediate object (an instance of the flavor *Comm*) as in the original design, but this object will also determine if the destination is a local or a remote agent. If the destination is a local agent then the messages are handled by *Comm* as in the original design. If the destination is a remote agent then the message is passed to another object (the *Remote* flavor-instance) that will take further steps to assure delivery. First, *Remote* determines if it has a communication channel to the YAPS process where the destination agent resides, and, if it does, transmits the message. Otherwise, the CLERK process is queried for the address of the destination agent. If the CLERK process does not has the address then the atom *nil* is returned to YAPS. If the address is known by the CLERK process, then *Remote* uses that address to establish a communication channel between the local YAPS process and the process where the remote agent resides. Once the channel has been established, the message is transmitted.

In order to read messages from external agents, the top level control loop handled by *Control* has been modified to send a message to *Remote* to read all the messages logged on the communication channel and to decode them.

There is a second version of this distributed environment currently being developed in which a front-end object is created locally on behalf of the contacted remote agent. A *RemoteAgent* flavor has been defined to create this *phantom* agent on demand in the local environment. Messages will be sent to this front-end object, which in turn will take care of delivering the message to the real agent that resides in some other YAPS process or host. This *RemoteAgent* flavor handles the same messages as the *AgentRB* flavor, and is instantiated whenever *Comm* tries to transmit a message to an agent unknown in the local environment. As part of its instantiation process, the *RemoteAgent* flavor will contact the CLERK process for the address of the *assumed* remote agent. If the CLERK does not know about the remote agent then an error message is sent to the user and the initialization process is undone. Otherwise, a communication channel will be established to the remote agent and the front-end object instantiation will be completed.

Also under consideration for future work are possible extensions of YAPS itself. One computational difficulty involves information that must be shared among multiple agent rule sets, as found in the *AgentRB* flavor. As it is, YAPS provides no direct means of sharing such global data among multiple databases. Indeed, the only way for each rule-based agent to have access to the data is for it to have its own copy of that data or for it to send a message to some common clearing-house requesting the data whenever it is needed. In either case considerable overhead can be involved depending upon how much data must be shared. If multiple copies are kept, the overhead is in terms of the extra space required and the cost of maintaining each of those copies. If the data are kept in a common area, then the overhead arises in the time required to access the data whenever needed. By modifying YAPS we may be able to create an architecture that will allow us to control the scoping and sharing of data, possibly reducing both spatial and temporal overhead for such data. As goals, facts and rules are all acceptable database entities, the resulting structure will support a blackboard control architecture (actually several variants of this architecture) as well. Work has started in this area focusing on modifications, via flavors, of the discrimination net used in YAPS, and a functional design is being tested.

Finally, we intend to continue development of the agent subsystem under our theoretical framework and as a component of the entire interview system. This will entail the evolution of possibly new and different agent flavors. It will include the addition of new features to existing flavors, such as *Comm,* that currently make up the subsystem architecture. And it will involve the improvement of the interface between humans (both experimenter and subject) and the computer-based portion of the agent subsystem.

CONCLUSION

In conclusion, the interview system and the agent subsystem continue to develop as useful and flexible tools with which to record interactions of experimental subjects and to classify and study the DTDM processes involved in those interactions. In addition, it is based upon a theoretical framework and composite model that have been developed to represent such processes. Thus it provides an experimental framework within which theoretical issues may be addressed.

As a part of the theoretical framework the concept of DTDM protocols and agents has been introduced. This concept, we feel, will be useful in addressing the issues underlying distributed problem-solving behavior. By representing interactions at a variety of abstraction levels and along different processing dimensions, it will permit us to begin to decompose such behavior both qualitatively and quantitatively for examination and analysis. In addition, because it meshes closely with the notion of objects and object-oriented programming, computer-based implementation of the agent metaphor is possible using this kind of programming language. Hence the concept not only provides a theoretical abstraction but permits experimental representation and testing as well. By assisting the experimenter in isolating the variables and processes involved and by helping to classify those variables and processes with respect to the processing dimensions that have been discussed, this approach will, we expect, provide additional insight into distributed tactical decision-making processes.

ACKNOWLEDGEMENTS

We are especially grateful to Bruce W. Hamill for the comments, suggestions, and invaluable aid he provided in our preparation of this paper. In addition, we deeply appreciate the work done by Wilfredo V. Perez and Stephen G. Pimentel to establish an interprocess communication capability allowing the agent subsystem to operate in a multi-process, multi-host environment, as well as their assistance in documenting these recent extensions.

This research is sponsored by the Office of Naval Research. The principal investigators are Bruce W. Hamill and Robert L. Stewart.

REFERENCES

Allen, Elizabeth M., Randall H. Trigg, and Richard J. Wood. 'The Maryland Artificial Intelligence Group Franz Lisp Environment' College Park, MD: Dept. of Computer Science, Univ. of Maryland. TR-1226, 1983.

Allen, Elizabeth M. 'YAPS: Yet Another Production System.' College Park, MD: Dept. of Computer Science, Univ. of Maryland. TR-1146, 1983.

Halushynsky, G. D., and J. K. Beam. 'A Concept for Navy Command and Control in the Year 2000.' *Johns Hopkins APL Technical Digest.* Vol. 5, No. 1, pp. 9-18. Laurel, MD: Johns Hopkins University / Applied Physics Laboratory, 1984.

Hamill, Bruce W., and Robert L. Stewart. 'Modeling the Acquisition and Representation of Knowledge for Distributed Tactical Decision Making.' *Johns Hopkins APL Technical Digest.* Vol. 7, No. 1, pp. 31-38. Laurel, MD: Johns Hopkins University / Applied Physics Laboratory, 1986.

Lawson, J. S. *Engineering Design Guidance for the Navy Command Control System.* Report prepared for the C^{3}I Systems and Technology Directorate, Naval Electronic Systems Command, 29 Aug 1984.

Newell, A. and H. Simon. *Human Problem Solving.* Englewood Cliffs, NJ: Prentice-Hall, 1972.

Rasmussen, J. 'Skills, Rules, and Knowledge: Signals, Signs, and Symbols, and Other Distinctions in Human Performance Models.' *IEEE Trans. Systems, Man, and Cybernetics* 13. IEEE, 1983.

Rasmussen, J. and M. Lind. *A Model of Human Decision Making in Complex Systems and Its Use for Design of System Control Strategies.* Roskilde, Denmark: Riso National Laboratory, 1982. Report No. Riso-M-2349.

Tanenbaum, A.S. *Computer Networks.* Englewood Cliffs, NJ: Prentice-Hall, 1981.

A DISTRIBUTED HYPOTHESIS-TESTING TEAM DECISION PROBLEM WITH COMMUNICATIONS COST*

Jason D. Papastavrou
Michael Athans
Massachusetts Institute of Technology
Cambridge, Ma. 02139, U.S.A.

ABSTRACT

In this paper we formulate and solve a distributed binary hypothesis-testing problem. We consider a cooperative team that consists of two decision makers (DM's); one is refered to as the primary DM and the other as the consulting DM. The team objective is to carry out binary hypothesis testing based upon uncertain measurements. The primary DM can declare his decision based only on its own measurements; however, in ambiguous situations the primary DM can ask the consulting DM for an opinion and it incurs a communications cost. Then the consulting DM transmits either a definite recommendation or pleads ignorance. The primary DM has the responsibility of making a final definitive decision. The team objective is the minimization of the probability of error, taking into account different costs for hypothesis misclassification and communication costs. Numerical results are included to demonstrate the dependence of the different decision thresholds on the problem parameters, including different perceptions of the prior information.

1. Introduction and Motivation.

In this paper we formulate, solve, and analyze a distributed hypothesis- testing problem which is an abstraction of a wide class of team decision problems. It represents a normative version of the "second-opinion" problem in which a primary decision maker (DM) has the option of soliciting, at a cost, the opinion of a consulting DM when faced with an ambiguous interpretation of uncertain evidence.

1.1 Motivating Examples.

Our major motivation for this research is provided by generic hypothesis- testing problems in the field of Command and Control. To be specific, consider the problem of target detection formalized as a binary hypothesis testing problem (H_0 means no target, while H_1 denotes the presense of a target). Suppose that independent noisy measurements are obtained by two geographically distributed sensors (Figure 1). One sensor, the primary DM, has final responsibility for declaring the presense or absence of a target, with different costs associated with the probability of false alarm versus the probability of missed detection. If the primary DM relied only on the measurements of his own sensor, then we have a classical centralized detection problem that has been extensively analyzed; see, for example, Van Trees [1]. If the actual measurements of the second sensor were communicated to the primary DM, we have once more a classical centralized detection problem in which we have two independent measurements on the same hypothesis; in this case, we require communication of raw data and this is expensive both from a channel bandwidth point of view and, perhaps more importantly, because radio or acoustic communication can be intercepted by the enemy.

Continuing with the target detection problem, we can arrive at the model that we shall use in the sequel by making the following assumptions which model the desire to communicate as little as possible. The primary DM can look at the data from his own sensor and attempt to arrive at a decision using a likelihood-ratio test (lrt), which yields a threshold test in the linear-Gaussian case. Quite often the primary DM can be confident about the quality of his decision. However, we can imagine that there will be instances that the data will be close to the decision threshold, corresponding to an ambiguous situation for the primary DM. In such cases it may pay off to incur a communications cost and seek some information from the other available sensor. It remains to establish what is the nature of the information to be transmitted back to the primary DM.

In our model, we assume the existence of a consulting DM having access to the data from the other sensor. We assume that the consulting DM has the ability to map the raw data from his sensor into decisions. The consulting DM is "activated" only at the request of the primary DM. It is natural to speculate that its advise will be ternary in nature: YES, I think there is a target; NO, I do not think there is a target; and, SORRY, NOT SURE MYSELF. Note that these transmitted decisions in general require less bits than the raw sensor data, hence the communication is cheap and more likely to escape enemy interception. Then, the primary DM based upon the message received from the consulting DM has the responsibility of making the final binary team decision on whether the target is present or absent.

The need for communicating with small-bit messages can be appreciated if we think of detecting an enemy submarine using passive sonar (Figure 2). We associate the primary DM with an attack submarine, and the consulting DM with a surface destroyer. Both have towed-array sonar capable of long-range enemy submarine detection. Request for information from the submarine to the destroyer can be initiated by having the sub eject a slot- buoy with a prerecorded low-power radio message. A short active sonar pulse can be used to transmit the recommendation from the destroyer to the submarine. Thus, the submarine has the choice of obtaining a "second opinion" with minimal compromise of its covert mission.

Of course, target detection is only an example of more general binary hypothesis-testing problems. Hence, one can readily extend the basic distributed team decision problem setup to other situations. For example, in the area of medical diagnosis we imagine a primary physician interpreting the outcomes of several tests. In case of doubt, he sends the patient to another consulting physician for other tests (at a dollar cost), and seeks his recommendation. However, the primary physician has the final diagnostic responsibility. Similar scenarios occur in the intelligence field where the "compartmentalization" of sensitive data, or the protection of a spy, dictate infrequent and low-bit communications. In more general military Command and Control problems, we seek insight on formalizing the need to break EMCON, and at what cost, to resolve tactical situation assessment ambiguities.

1.2 Literature Review.

The solution of distributed decision problems is quite a bit different, and much more difficult, as compared to their centralized counterparts. Indeed there is only a handful of papers that deal with solutions to distributed hypothesis-testing problems. The first attempt to illustrate the difficulties of dealing with distributed hypothesis-testing problems was published by Tenney and Sandell [2]; they point out that the decision thresholds are in general coupled. Ekchian [3] and Ekchian and Tenney [4] deal with detection networks in which downstream DM's make decisions based upon their local measurements and upstream DM decisions. Kushner and Pacut [5] introduced a delay cost (somewhat similar to the communications cost in our model) in the case that the observations have exponential distributions, and performed a simulation study. Recently, Chair and Varshney [6] have pointed out how the results in [2] can be extended in more general settings. Boettcher [7] and Boettcher and Tenney [8], [9], have shown how to modify the normative solutions in [4] to reflect human limitation constraints, and arrive in at normative/descriptive model that captures the constraints of human implementation in the presense of decision deadlines and increasing human workload; experiments using human subjects showed close agreement with the predictions of their normative/descriptive model. Finally, Tsitsiklis [10] and Tsitsiklis and Athans [11] demonstrate that such distributed hypothesis-testing problems are NP-complete; their research provides theoretical evidence regarding the inherent complexity of solving optimal distributed decision problems as compared to their centralized counterparts (which are trivially solvable).

1.3 Contributions of this Research.

The main contribution of this paper relates to the formulation and optimal solution of the team decision problem described above. Under the assumption that the measurements are conditionally independent, we show that the optimal decision rules for both the primary and the consulting DM are deterministic and are expressed as likelihood-ratio tests with constant thresholds which are tightly coupled (see Section 3).

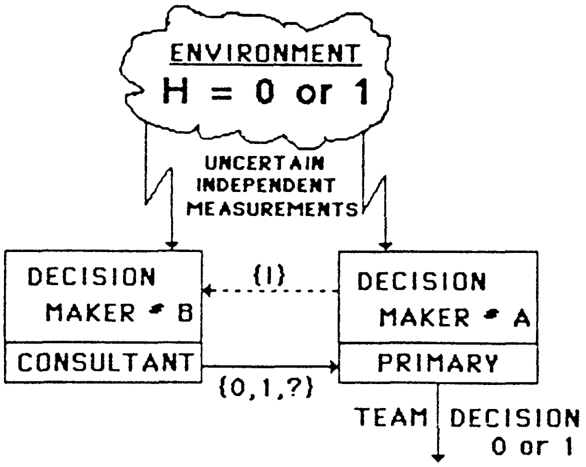


Figure 1: Problem Formulation.

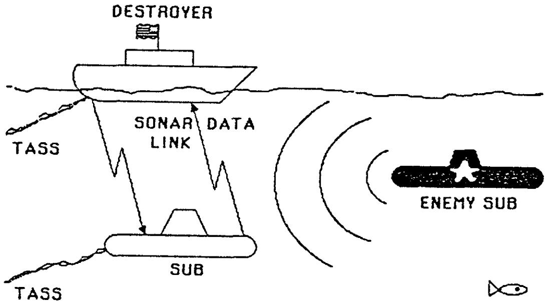


Figure 2: Anti-Submarine Warfare(ASW) Example.

When we specialize the general results to the case that the observations are linear and the statistics are Gaussian, then we are able to derive explicit expressions for the decision thresholds for both the primary and consulting DM's (see Section 4). These threshold equations are tightly coupled, thereby necessitating an iterative solution. They provide clear-cut evidence that the DM's indeed operate as team members; their optimal thresholds are very different from those that they would use in isolation, i.e. in a non-team setting. This, of course, was the case in other versions of the distributed hypothesis-testing problem, e.g. [2].

The numerical sensitivity results (summarized in Section 5) for the linear-Gaussian case provide much needed intuitive understanding of the problem and concrete evidence that the team members operate in a more-or-less intuitive manner, especially after the fact. We study the impact of changing the communications cost and the measurement accuracy of each DM upon the decision thresholds and the overall team performance. In this manner we can obtain valuable insight on the optimal communication frequency between the DM's. As to be expected, as the communication cost increases, the frequency of communication (and asking for a second opinion) decreases, and the team performance approaches that of the primary DM operating in isolation. In addition, we compare the overall distributed team performance to the centralized version of the problem in which the primary DM had access, at no cost, to both sets of observations. In this manner, we can study the degree of inherent performance degradation to be expected as a consequence of enforcing the distributed decision architecture in the overall decision making process.

Finally, we study the team performance degradation when one of the team members, either the primary or the consulting DM, has an erroneous estimate of the hypotheses prior probabilities. This corresponds to mildly different mental models of the prior situation assesment; see Athans [12]. As expected the team performance is much more sensitive to misperceptions by the primary DM as compared to similar misperceptions by the consulting DM. This implies that, if team training reduces misperceptions on the part of the DM's, the greatest payoff is obtained in training the primary DM.

2. Problem Definition

The problem is one of hypothesis testing. The team has to choose among two alternative hypotheses H_0 and H_1, with a priori probabilities

$$P(H_0)=p_0 \qquad P(H_1)=p_1 \tag{1}$$

Each of two DM's, one called primary (DM A) and one consulting (DM B), receives an uncertain measurement y_α and y_β respectively (Figure 1), distributed with known joint probability density functions

$$P(y_\alpha,y_\beta \mid H_i) \quad ; \quad i=0,1 \tag{2}$$

The final decision of the team u_f (0 or 1, indicating H_0 or H_1 to be true) is the responsibility of the primary DM. DM A initially makes a preliminary decision u_α where it can either decide (0 or 1) on the basis of its own data (ie y_α), or at a cost ($C \geq 0$) can solicit DM B's opinion (u_α=I), prior to making the commital decision.

The consulting DM's decision u_β consists of three distinct messages (call them : x,v and z) and is activated only when asked. We decided to assign three messages to DM B, because we wanted to have one message indicating each of the two hypotheses and one message indicating that the consulting DM is 'not sure.' In fact, we proved that the optimal content for the messages of DM B is the one mentioned above.

When the message from DM B is received,the burden shifts back to the primary DM, which is called to make the commital decision of the team based on his own data and the information from the consulting DM.

We now define the following cost function :

$$J \ : \{0,1\} x \{H_0,H_1\} \rightarrow \mathbf{R} \tag{3}$$

with $J(u_f,H_i)$ being the cost incurred by the team choosing u_f, when H_i is true.

Then, the optimality criterion for the team is a function

$$J^*: \{0,1,I\} x \{0,1\} x \{H_0,H_1\} \rightarrow \mathbf{R} \tag{4}$$

with :

$$J^*(u_\alpha,u_f,H_i)= \begin{matrix} J(u_f,H_i)+C \ ; \ u_\alpha=I \text{ (information requested)} \\ J(u_f,H_i) \quad ; \ \text{otherwise} \end{matrix} \tag{5}$$

The cost structure of the problem is the usual cost structure used in Hypothesis Testing problems, but also includes the non-negative communication cost, which the team incurs when the DM A decides to obtain the consulting DM's information.

Remark : According to the rules of the problem, when the preliminary decision u_α of the primary DM is 0 or 1, then the final team decision is 0 or 1 respectively (ie $P(u_f=i \mid u_\alpha=i)=1$ for i=0,1).

The objective of the decision strategies will be to minimize the expected cost incurred

$$\min \ E[J^*(u_\alpha,u_f,H)] \tag{6}$$

where the minimization is over the decision rules of the two DMs. Note that the decision rule of the consulting DM is implicitly included in the cost function, through the final team decision u_f (which is a function of the decision of the consulting DM).

All the prior information is known to both DMs. The only information they do not share is their observations. Each DM knows only its own observation and, because of the conditional independence assumption, nothing about the other DM's observation.

The problem can now be stated as follows :

Problem : Given p_0, p_1, the distributions $P(y_\alpha,y_\beta \mid H_i)$ for i=0,1 with $y_\alpha \in Y_\alpha$, $y_\beta \in Y_\beta$, and the cost function J^*, find the decision rules u_α,u_β and u_f as functions

$$\gamma_\alpha : Y_\alpha \rightarrow \{0,1,I\} \tag{7}$$

$$\gamma_\beta : Y_\beta \rightarrow \{x,v,z\} \tag{8}$$

and

$$\gamma_f \ : \ Y_\alpha \, x \, \{x,v,z\} \rightarrow \{0,1\} \tag{9}$$

(subject to : $P(u_f=i \mid u_\alpha=i)=1$ for i=0,1), which minimize the expected cost.

NOTE : The centralized counterpart of the problem, where a single DM receives both observations is a well known problem. The solution is deterministic and given by a likelihood ratio test (lrt). That is :

$$\gamma_c : Y_\alpha \times Y_\beta \rightarrow \{0,1\} \qquad (10)$$

with

$$\gamma_c(y_\alpha,y_\beta)= \begin{cases} 0 \; ; & \Lambda(y_\alpha,y_\beta) \geq t \\ 1 \; ; & \text{otherwise} \end{cases} \qquad (11)$$

where

$$\begin{aligned} \Lambda(y_\alpha,y_\beta) &= [P(y_\alpha,y_\beta \mid H_0)p_0]/ [P(y_\alpha,y_\beta \mid H_1)p_1] \\ &= P(H_0 \mid y_\alpha,y_\beta)/ P(H_1 \mid y_\alpha,y_\beta) \qquad (12) \end{aligned}$$

and t is a precomputed threshold

$$t = [J(0,H_1)- J(1,H_1)]/ [J(1,H_0)- J(0,H_0)] \qquad (13)$$

provided $J(1,H_0) > J(0,H_0)$. Thus, the difficulty of our problem arises because of its decentralized nature.

We will show that, under certain assumptions, the most restrictive of which is conditional independence of the observations, the optimal decision rules for the *Problem* are deterministic and given by lrt's with constant thresholds. The thresholds of the two DMs are coupled, indicating that the DMs work as a team rather than individuals.

3. About the Solution to the General Problem

In order to be able to solve the *Problem* , we make the following assumptions.

ASSUMPTION 1: $J(1,H_0) > J(0,H_0) \;\; ; \;\; J(0,H_1) > J(1,H_1)$ (14)

or it is more costly for the team to err than to be correct.

This logical assumption is made in order to motivate the team members to avoid erring and in order to enable us to algebraically put the optimal decisions in lrt form.

ASSUMPTION 2 : $P(y_\alpha \mid y_\beta,H_i) = P(y_\alpha \mid H_i)$ and

$$P(y_\beta \mid y_\alpha,H_i) = P(y_\beta \mid H_i) \;\; ; \;\; i=0,1 \qquad (15)$$

or the observations y_α and y_β are conditionally independent.

This assumption removes the dependence of the one observation on the other and thus allows us to write the optimal decision rules as lrt's with constant thresholds.

ASSUMPTION 3 : Without loss of generality assume that :

$$\frac{P(u_\beta=x \mid u_\alpha=I,H_0)}{P(u_\beta=x \mid u_\alpha=I,H_1)} \geq \frac{P(u_\beta=v \mid u_\alpha=I,H_0)}{P(u_\beta=v \mid u_\alpha=I,H_1)} \geq \frac{P(u_\beta=z \mid u_\alpha=I,H_0)}{P(u_\beta=z \mid u_\alpha=I,H_1)} \qquad (16)$$

This assumption is made in order to distinguish between the messages of DM B.

The optimal decision rules for all three decisions of our problem (u_α, u_β, u_f) are given by *deterministic* functions which are expressed as *likelihood ratio tests*, with *constant* thresholds. The three thresholds of the primary DM (two for u_α and one for u_f) and the two thresholds of the consulting DM (for u_β) can not be obtained in closed form. They are coupled, that is the thresholds of one DM are given as functions of the thresholds of the other DM.

Another important result is that, when the optimal decision rules are employed and the consulting DM's decision is x (or z), then the optimal final decision rule of the primary DM is always 0 (or 1) :

$$P(u_f=0 \mid u_\alpha=I,u_\beta=x,y_\alpha)=1 \qquad (17)$$

$$\text{for all } y_\alpha \in \{y_\alpha \mid P(u_\alpha=I \mid y_\alpha)=1, y_\alpha \in Y_\alpha\}$$

and

$$P(u_f=1 \mid u_\alpha=I,u_\beta=z,y_\alpha)=1 \qquad (18)$$

$$\text{for all } y_\alpha \in \{y_\alpha \mid P(u_\alpha=I \mid y_\alpha)=1, y_\alpha \in Y_\alpha\}$$

Thus, we can simplify our notation by changing the DM B decisions from x to 0, from z to 1 and from v to ? (which is interpreted as :"I am not sure"). The team's decision process can be now described as follows : Each of the two DMs receives an observation. Then, the primary DM can either make the final decision (0 or 1) or can decide to incur the communication cost (u_α=I) and pass the responsibility of the final decision to the consulting DM. When called upon, the consulting DM can either make the final decision or shift the burden back to DM A (u_β=?), in which case the primary DM is forced to make the final decision, based on its own observation (y_α) and the fact that DM B decided u_β=? .

4. A Gaussian Example

We now present detailed threshold equations for the case where the probability distributions of the two observations are Gaussian. We selected the Gaussian distribution, despite its cumbersome algebraic formulae, because of its generality. Our objective is to perform numerical sensitivity analysis to the solution of this example, in order to gain information on the team 'activities.'

We assume that the observations are distributed with the following Gaussian distributions :

$$y_\alpha \sim N(\mu,\sigma_\alpha^2) \;\; ; \;\; y_\beta \sim N(\mu,\sigma_\beta^2) \qquad (19)$$

The two alternative hypotheses are :

$$H_0 : \mu=\mu_0 \quad \text{or} \quad H_1 : \mu=\mu_1 \qquad (20)$$

Without loss of generality, assume that :

$$\mu_0 < \mu_1 \qquad (21)$$

The rest of the notation is the same as in the general problem described above.

We can show that the optimum decision rules for this example are given by thresholds on the *observation* axes, as shown in Figure 3.

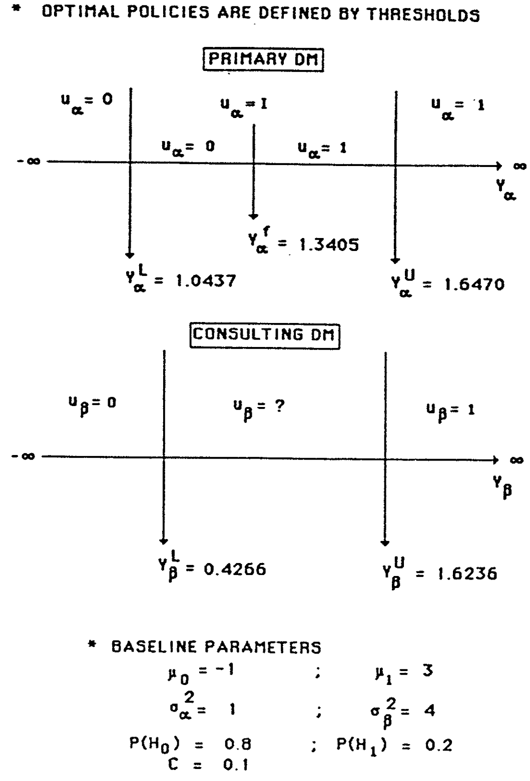


Figure 3: The Gaussian Case.

Before presenting the equations of the thresholds, we define some variables.

$Y_\alpha{}^l$: lower threshold of DM A

$Y_\alpha{}^u$: upper threshold of DM A

$Y_\alpha{}^f$: threshold for the final decision of DM A

$Y_\beta{}^l$: lower threshold of DM B

$Y_\beta{}^u$: upper threshold of DM B

$$\Phi_i^j(k) = \int_{-\infty}^{\frac{Y_i^j - \mu_k}{\sigma_i}} (2\pi)^{-0.5} \exp\left(-0.5\, x^2\right) dx$$

$$\text{for } i=\alpha,\beta \;\; ; \;\; j=l,f,u \;\; ; \;\; k=0,1$$

Note that the above function is the well-known error function, presented with notational modifications to fit the purposes of the problem.

$$W^1 = 0.5\, [\Phi_\beta{}^u(0)-\Phi_\beta{}^u(1)] \qquad (22)$$

$$W^2 = \frac{\Phi_\beta{}^u(0)-\Phi_\beta{}^l(0)-\Phi_\beta{}^u(1)+\Phi_\beta{}^l(1)+\Phi_\beta{}^l(0)\Phi_\beta{}^u(1)-\Phi_\beta{}^u(0)\Phi_\beta{}^l(1)}{\Phi_\beta{}^u(0)-\Phi_\beta{}^l(0)+\Phi_\beta{}^u(1)-\Phi_\beta{}^l(1)} \qquad (23)$$

$$W^3 = \frac{\Phi_\beta{}^l(0)\Phi_\beta{}^u(1)-\Phi_\beta{}^l(1)\Phi_\beta{}^u(0)}{\Phi_\beta{}^u(0)-\Phi_\beta{}^l(0)+\Phi_\beta{}^u(1)-\Phi_\beta{}^l(1)} \qquad (24)$$

$$W^4 = 0.5\,[\Phi_\beta{}^u(0)-\Phi_\beta{}^u(1)] \qquad (25)$$

$$Y_\alpha{}^* = (\mu_0+\mu_1)/2 \;+\; [\sigma_\alpha{}^2/(\mu_1-\mu_0)]\ln[p_0/(1-p_0)] \qquad (26)$$

$$Y_\beta{}^* = (\mu_0+\mu_1)/2 \;+\; [\sigma_\beta{}^2/(\mu_1-\mu_0)]\ln[p_0/(1-p_0)] \qquad (27)$$

In (26) and (27), the (centralized) maximum likelihood estimators for each DM are defined.

COROLLARY 1 : If $P(u_\alpha=I)>0$ (i.e. information is requested for some y_α) and if $P(u_\beta=?\mid u_\alpha=I)>0$ (i.e. "I am not sure" is returned for some y_β, when information is requested), then the optimal final decision rule of the primary DM is a deterministic function defined by :

$$\gamma_f(y_\alpha) = \begin{cases} 0 & \text{if} \quad y_\alpha \le Y_\alpha{}^f \\ 1 & \text{if} \quad y_\alpha > Y_\alpha{}^f \end{cases} \qquad (28)$$

where :

$$Y_\alpha{}^f = Y_\alpha{}^* + \frac{\sigma_\alpha{}^2}{\mu_1-\mu_0}\ln\left(\frac{\Phi_\beta{}^u(0)-\Phi_\beta{}^l(0)}{\Phi_\beta{}^u(1)-\Phi_\beta{}^l(1)}\right) \qquad (29)$$

COROLLARY 2 : If $P(u_\alpha=I)>0$ (i.e. information is requested for some y_α) and the primary DM's final decision rule is the one given by Corollary 1, then the optimal decision rule of the consulting DM is a deterministic function defined by :

$$\gamma_\beta(y_\beta) = \begin{cases} 0 & \text{if} \quad y_\beta \le Y_\beta{}^l \\ ? & \text{if} \quad Y_\beta{}^l < y_\beta \le Y_\beta{}^u \\ 1 & \text{if} \quad Y_\beta{}^u < y_\beta \end{cases} \qquad (30)$$

where :

$$Y_\beta{}^l = Y_\beta{}^* + \frac{\sigma_\beta{}^2}{\mu_1-\mu_0}\ln\left(\min\left\{\frac{\Phi_\alpha{}^u(0)-\Phi_\alpha{}^f(0)}{\Phi_\alpha{}^u(1)-\Phi_\alpha{}^f(1)}, \frac{\Phi_\alpha{}^u(0)-\Phi_\alpha{}^l(0)}{\Phi_\alpha{}^u(1)-\Phi_\alpha{}^l(1)}\right\}\right) \qquad (31)$$

and :

$$Y_\beta{}^u = Y_\beta{}^* + \frac{\sigma_\beta{}^2}{\mu_1-\mu_0}\ln\left(\max\left\{\frac{\Phi_\alpha{}^u(0)-\Phi_\alpha{}^l(0)}{\Phi_\alpha{}^u(1)-\Phi_\alpha{}^l(1)}, \frac{\Phi_\alpha{}^f(0)-\Phi_\alpha{}^l(0)}{\Phi_\alpha{}^f(1)-\Phi_\alpha{}^l(1)}\right\}\right) \qquad (32)$$

COROLLARY 3 : Given that the final decision rule employed by the primary DM is the one of Corollary 1 and that the decision rule employed by the consulting DM is the one of Corollary 2, then the optimal decision rule for the preliminary decision u_α of the primary DM is a deterministic function defined by :

$$\gamma_\alpha(y_\alpha) = \begin{cases} 0 & \text{if} \quad y_\alpha \le Y_\alpha{}^l \\ I & \text{if} \quad Y_\alpha{}^l < y_\alpha \le Y_\alpha{}^u \\ 1 & \text{if} \quad Y_\alpha{}^u < y_\alpha \end{cases} \qquad (33)$$

where :

$$Y_\alpha{}^l = \begin{cases} Y_\alpha{}^* + \frac{\sigma_\alpha{}^2}{\mu_1-\mu_0}\ln\left(\frac{1-\Phi_\beta{}^l(0)+C}{1-\Phi_\beta{}^l(1)-C}\right) & ; \quad 0 \le C < \min\{W^1, W^2\} \\ Y_\alpha{}^* + \frac{\sigma_\alpha{}^2}{\mu_1-\mu_0}\ln\left(\frac{1-\Phi_\beta{}^u(0)+C}{1-\Phi_\beta{}^u(1)-C}\right) & ; \quad W^2 < C \le W^4 \\ Y_\alpha{}^* & ; \quad \text{otherwise} \end{cases} \qquad (34)$$

and :

$$Y_\alpha{}^u = \begin{cases} Y_\alpha{}^* + \frac{\sigma_\alpha{}^2}{\mu_1-\mu_0}\ln\left(\frac{\Phi_\beta{}^l(0)-C}{\Phi_\beta{}^l(1)+C}\right) & ; \quad 0 \le C < \min\{W^3, W^4\} \\ Y_\alpha{}^* + \frac{\sigma_\alpha{}^2}{\mu_1-\mu_0}\ln\left(\frac{\Phi_\beta{}^u(0)-C}{\Phi_\beta{}^u(1)+C}\right) & ; \quad W^3 < C \le W \\ Y_\alpha{}^* & ; \quad \text{otherwise} \end{cases} \qquad (35)$$

REMARK : Observe that the equations of all the thresholds include (and possibly reduce to) a "centralized" part ($Y_i{}^*$) indicating the relation of our problem to its centralized counterpart.

5. NUMERICAL SENSITIVITY ANALYSIS

We now perform sensitivity analysis to the solution of the Gaussian example. Our objective is to analyze the effects on the team performance from varying the parameters of our problem, in order to obtain better understanding of the decentralized team decision mechanism. We vary the quality of the observations of each DM (the variance of each DM), the a priori likelihood of the hypotheses and the communication cost.

Finally, we study the effects of different a priori knowledge for each DM.

We use the following 'minimum error' cost function :

$$J(u_f,H_i) = \begin{cases} 0 & ; \quad u_f = i \\ 1 & ; \quad u_f \ne i \end{cases} \qquad (36)$$

We do not need to vary the cost function, because this would be mathematically equivalent to varying the a priori probabilities of the two hypotheses.

5.1 Effects of varying the quality of the observations of the Primary DM

Denote :

$C1^*$ = cost incured if the consulting DM makes the decision alone .

We distinguish two cases depending on the cost associated with the information (ie the of quality of information)

CASE 1 : $\min(P_0, 1-P_0) \le C1^* + C$

As the variance of the primary DM increases, it becomes less costly for the team to have the primary DM always decide the more likely hypothesis, than request for information. This occurs becayse the observation of DM A becomes increasingly worthless. Thus, the primary DM progressively ignores its observation and in order to minimize cost has to choose between "de facto" deciding the more likely hypothesis (and incuring cost equal to the probability of the least likely hypothesis) or "de facto" requesting for information (and thus incuring the communication cost plus the cost of the consulting DM). In this case, the prior is less than the latter and so the optimum decision of the primary DM, as its variance tends to infinity is to always decide the more likely hypothesis (Figure 4, P_0= .8). Thus :

$$\lim_{\sigma_0{}^2 \to \infty} P(u_\alpha=I) \to 0$$

Moreover, the percentage gain in cost achieved by the team of DMs, relative to the cost incured by a single DM obtaining a single observation, assymptotically goes to 0, as the variance of the primary DM goes to infinity (Figure 5, P_0= .8).

An interesting insight can be obtained from Figure 5 (P_0= .8). As the variance of the primary DM increases the percentage improvement in cost (defined above) is initially increasing and then decreasing assymptotically to zero. The reason for this is that for very small variances, the observations of the primary DM are so good that it does not need the information of the consulting DM. As the variance increases, the primary DM makes better use of the information and so the percentage improvement increases. But, at a certain point as the quality of the observations worsens, the primary DM finds less costly to start declaring more often the more likely hypothesis (ie to bias its decision towards the more likely hypothesis) than requesting for information, for reasons mentioned above, and so the percentage improvement from then on decreases.

CASE 2 : $\min(P_0, 1-P_0) > C1^* + C$

With reasoning similar to the above, we obtain that (Figure 4, P_0= .5) :

$$\lim_{\sigma_0{}^2 \to \infty} P(u_\alpha=I) \to 1$$

Moreover, the percentage improvement is strictly increasing (and keeps incresing to a precomptutable limit ; Figure 5, P_0= .5). This reinforces the last point we made in (Case 1) above. Since in the present case it is always less costly for the primary DM to request and use the

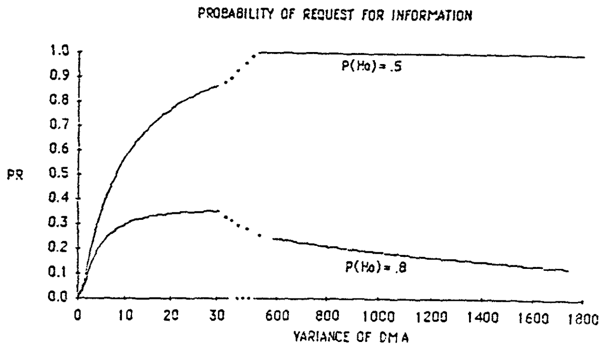


FIGURE 4:

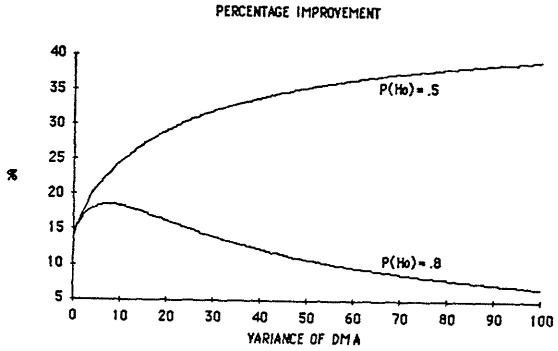


FIGURE 5

information than to bias its decision towards the more likely hypothesis, the percentage improvement curve does not exhibit the non-monotonic behavior observed in (Case 1) above (where P_0= .8).

5.2 Effects of varying the quality of the observations of the Consulting DM

As the variance of the consulting DM's observations increases, less information is requested by the primary DM, that is the primary DM's upper and lower thresholds move closer to each other (Figure 6). This is something we expected, since information of lesser quality is less profitable (more costly) to the team of DMs.

We should note here that the thresholds of a DM is an alternative way of representing the probabilities of the DM's decisions, since the decision regions are characterized by the thresholds. For example :

$$P(u_\alpha=I) = \sum_{H} \int_{y_\alpha:\, Y_\alpha^{l} < y_\alpha < Y_\alpha^{u}} P(y_\alpha \mid H)\, P(H)$$

•

The thresholds of the consulting DM demonstrate some interesting points of the team behavior (Figure 7). For small values of the variance they are very close together, as the quality of the observations is very good and so the consulting DM is willing to make the final team decision. As the variance increases, DM B becomes more willing to return u_β=? (i.e. "I am not sure") and let DM A make the final team decision. As the variance continues to increase, the thresholds of the consulting DM converge again. It might seem counter-intuitive, but there is a simple explanation. The consulting DM recognizes that the primary DM, despite knowing that the quality of the consulting DM's information is bad, is willing to incur the communication cost to obtain the information. This indicates that the primary DM is 'confused', that is, the a posteriori probabilities of the two hypotheses (given its observation) are very close together. Hense, the consulting DM becomes more willing to make the final decision. After a certain point ($\sigma_\beta^2 \approx 62.4$) the primary DM does find it worthwhile to request for information at all.

REMARK : Note in Figure 7 that the thresholds of the consulting DM converge to 1 which would have been the maximum likelihood threshold had the a priori probabilities of the two hypotheses been equal. But, the a priori probabilities *which the consulting DM uses in its calculations* are functions of the given a priori probabilities (ie p_i) and the fact that the primary DM requested for information (ie $P(u_\alpha=I \mid H_i)$). In fact, the consulting DM uses as its a priori probabilities its own estimates of the primary DM's a posteriori probabilities. That is :

$$P(H_0 \mid u_\alpha=I) = \frac{P(H_0)\, P(u_\alpha=I \mid H_0)}{\sum_{H} P(H)\, P(u_\alpha=I \mid H)} \tag{37}$$

From the above, we deduce that for large variances ($\sigma_\beta^2 \approx 62$) the estimates, of the consulting DM, for the a posteriori probabilities of the primary DM are very close to .5, reenforcing our point about the primary DM "being confused."

Finally, it is clear, that as the variance of the consulting DM increases, the perçentage gain in cost, achieved by the team of DMs, decreases to 0, since the primary DM eventually makes all the decisions alone (centralized).

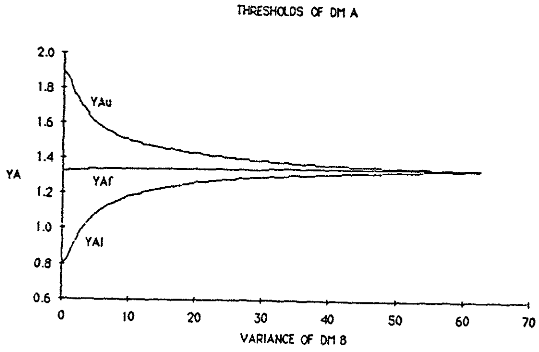


FIGURE 6

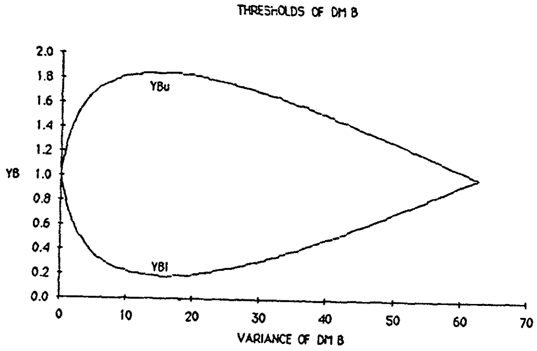


FIGURE 7

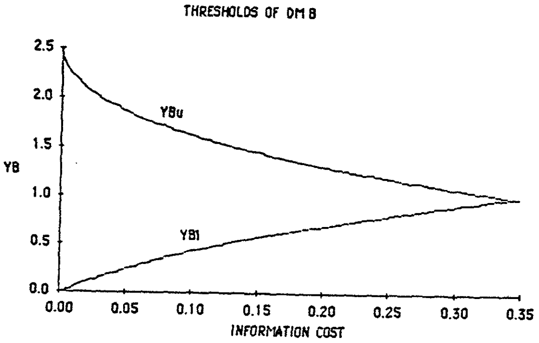


FIGURE 8

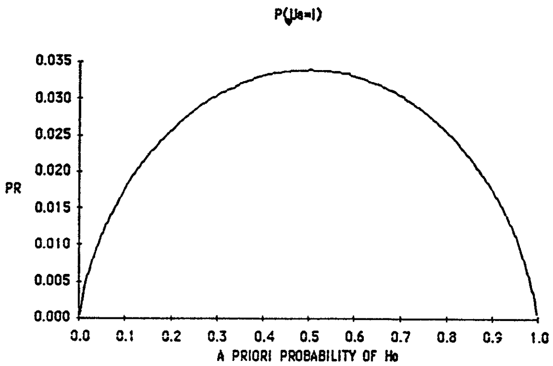


FIGURE 9

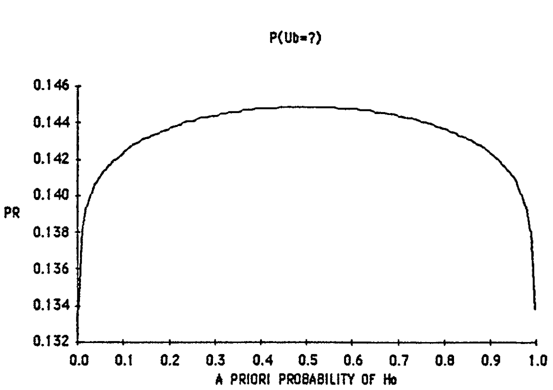


FIGURE 10

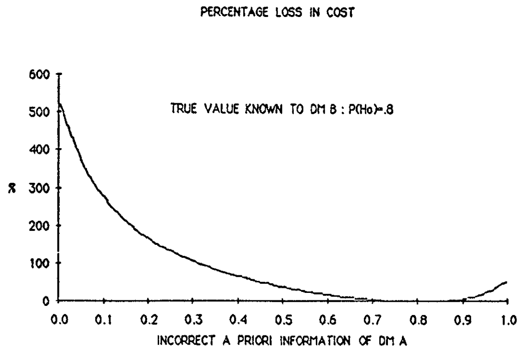


FIGURE 11

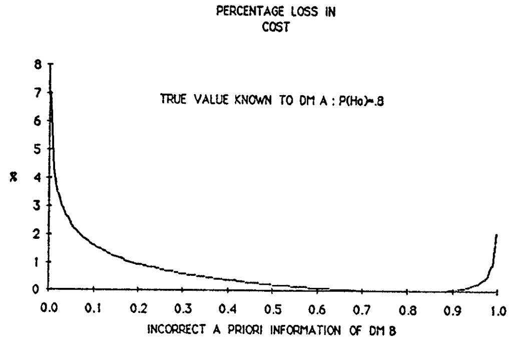


FIGURE 12

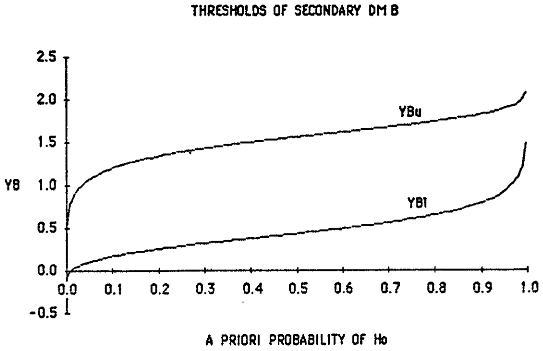


FIGURE 13

5.3 Effects of varying the Communication Cost

Increasing the communication cost is very similar to increasing the variance of the consulting DM, since in both cases the team "gets less for its money" (because the team has to incur an increased cost, either in the form of an increased communication cost, or in the form of the final cost, because of the worse performance of the consulting DM).

The thresholds of the primary DM, exhibit the same behavior as in 5.2 above (converging together at C≈.35). The thresholds of the consulting DM (Figure 8) converge together for the same reasons as in 5.2 above. Of course, the thresholds do not start together for small values of the communication cost (as in 5.2), because low communication cost does not imply ability for the consulting DM to make accurate decisions. In fact, for small values of the communication cost, DM A is compelled to request for information more often than what is really needed and so the consulting DM returns more often u_β=? (ie "I am not sure") and lets DM A make the team final decision.

Again it is clear that, as the communication cost increases, the percentage gain achieved by the team of the DMs decreases to zero (as the communication becomes more costly and less frequent, until we reach the centralized case).

5.4 Effects of varying the a priori probabilities of the hypotheses

This case does not present many interesting points. As expected, there is symmetry in the performance of the team around the line p_0= 0.5 . The closer p_0 is to 0.5 the more often information is requested by DM A (Figure 9) and the more often "I am not sure" is returned by DM B (Figure 10). This is understandable, because the closer p_0 is to 0.5, the bigger the a priori uncertainty. Consequently, the percentage improvement achieved by the team of the DMs is monotonically increasing with p_0 from 0 to 0.5 and monotonically decreasing from 0.5 to 1.

5.5 Effects of imperfect a priori information

CASE 1 : Only the consulting DM knows the true p_0

From Figure 11, where the true p_0 is 0.8, we deduce that our model is relatively robust. If the primary DM's erroneous p_0 is anywhere between 0.7 and 0.9, performance of the team will be not more than 10% away from the optimum.

CASE 2 : Only the primary DM knows the true p_0

As we see in Figure 12, where the true p_0 is 0.8, our model exhibits remarkable robustness qualities. If the consulting DM's erroneous p_0 is as far out as 0.01, the performance of the team will not be further than 7% away from the optimal. This can be explained by looking at the consulting DM's thresholds as functions of p_0 (Figure 13). We observe that for values of p_0 between 0.01 and 0.99, the thresholds do not change by much. This occurs because, as explained in detail in 5.2 above, the consulting DM knows that the primary DM requests for information when its a posteriori probabilities of the two hypotheses are roughly equal, which is the case indeed. As already stated, the consulting DM uses as its a priori probabilities its estimates of the a posteriori probabilities of the primary DM. Therefore, the consulting DM's estimates of the primary DM's a posteriori probabilities are good, besides the discrepancy in p_0, and the team's performance is not tampered by much.

Acknowledgment

The authors would like to thank Professor John N. Tsitsiklis for his valuable suggestions.

6. REFERENCES

[1] Van Trees, H.L. , "Detection, Estimation and Modulation Theory, vol. I", New York : Wiley, 1969 [2] Tenney, R.R., and Sandell, N.R., "Detection with Distributed Sensors", IEEE Trans. of Aerospace and Electronic Systems, AES-17, July 1981, pp. 501-509

[3] Ekchian, L.K., "Optimal Design of Distributed Detection Networks", Ph.D. Dissertation, Dept. of Elect. Eng. and Computer Science Mass. Inst. of Tech., Cambridge, Mass, 1982

[4] Ekchian, L.K., and Tenney, R.R., "Detection Networks", Proceedings of the 21st IEEE Conference on Decision and Control, 1982, pp. 686-691

[5] Kushner, H.J., and Pacut, A., "A Simulation Study of a Decentralized Detection Problem", IEEE Trans. on Automatic Control, 27, Oct. 1982, pp. 1116-1119

[6] Chair, Z., and Varshney, P.K., "Optimal Data Fusion in Multiple Sensor Detection Systems", IEEE Trans. on Aerospace and Electronic Systems, 21, January 1986, pp. 98-101

[7] Boettcher, K.L., "A Methodology for the Analysis and Design of Human Information Processing Organizations",Ph.D. Dissertation, Dept. of Elect. Eng. and Computer Science, Mass. Inst. of Tech., Cambridge, Mass., 1985

[8] Boettcher, K.L., and Tenney, R.R., "On the Analysis and Design of Human Information Processing Organizations", Proceedings of the 8th MIT/ONR Workshop on C^3 Systems, 1985, pp. 69-74

[9] Boettcher, K.L., and Tenney, R.R., "Distributed Decisionmaking with Constrained Decision Makers: A Case Study", Proceedings of the 8th MIT/ONR Workshop on C^3 Systems, pp. 75-79

[10] Tsitsiklis, J.N., "Problems in Decentralized Decision Making and Computation", Ph.D. Dissertation, Dept. of Elec. Eng. and Computer Science, Mass. Inst. of Tech., Cambridge, Mass., 1984

[11] Tsitsiklis, J.N., and Athans, M., "On the Complexity of Decentralized Decision Making and Detection Problems", IEEE Trans. on Automatic Control, 30, May 1985, pp. 440-446

* Research conducted at the MIT Laboratory for Information and Decision Systems with support provided by the Office of Naval Research under contract ONR/N00014-84-K-0519 (NR 649-003)

PARALLELISM IN MULTITARGET TRACKING AND ADAPTATION TO MULTIPROCESSOR ARCHITECTURES*

Thomas Kurien Thomas G. Allen Robert B. Washburn, Jr.

ALPHATECH, Inc.
2 Burlington Executive Center
111 Middlesex Turnpike
Burlington, Massachusetts 01803

ABSTRACT

In practical surveillance problems, the computational requirements of a model-based multitarget tracking algorithm are large due to the combinatorial problem of associating returns with targets. Since dynamic models for target motion are the same for all targets, the processing requirements are identical and, in general, can be done in parallel. This paper provides an analysis of the parallelism of a particular model-based multitarget tracking algorithm. Achievable speed-up and efficiency of processor utilization, when the algorithm is implemented on suitable multiprocessor architectures, are also examined.

1. INTRODUCTION

Ballistic Missile Defense and Airborne Surveillance require identification and tracking of several hundred targets in real time. Multitarget tracking algorithms designed for these problems demand large computational resources which generally cannot be fulfilled with conventional von Neumann types of processors. Fortunately, since multitarget tracking involves the simultaneous tracking of several targets, the algorithm will contain several computational tasks** which may be run in parallel. Furthermore, since each of the targets tracked is assumed to have the same model, these parallel tasks will require similar sequences of operations on a computer.

With the advent of multiprocessor architectures, adapting multitarget tracking algorithms to these new computer architectures poses a challenging problem.

Past studies in this area [1]-[3] have two features in common:

1. They examine "operational" multitarget tracking algorithms which use simple rules for associating target tracks with returns, and $\alpha - \beta$ tracking filters to combine information in the data for track-report pairs.

2. They examine the adaptation of the tracking algorithm onto array processors which have several processing elements executing the same instruction broadcast by a Control Unit.

During the course of the last decade, several model-based multitarget tracking algorithms have been developed [4-6]. These algorithms permit the evaluation of a mathematically optimal set of target tracks using the measurements available through the sensors and any other information available for the model describing the targets and the environment. Evaluation of the optimal solution will require substantially larger computer resources than the operational tracking algorithms; however, by specifying a set of heuristics, the computational requirements of the model-based tracking algorithms can be brought down to levels close to that of operational algorithms. The advantage of designing an algorithm using the model-based approach is that it starts by postulating all possible target tracks and then systematically eliminates unlikely ones. Implementation of such an algorithm requires the use of Kalman filters for updating target tracks and maintaining hypotheses corresponding to combinations of likely target tracks. This approach is markedly different from operational tracking algorithms and, to our knowledge, its implementations on parallel processor architectures has not been studied as yet.

This paper provides preliminary estimates of:

1. Computational requirements of a model-based multitarget-tracking algorithm, and

2. Achievable speed up and efficiency of processor utilization using suitable parallel processor architectures.

Among the various model-based multitarget tracking algorithms proposed in the literature, we will confine attention to the one proposed in [6]. An overview of this algorithm, which we refer to as Multitracker, is provided in Section 2. Computational requirements of a multitarget tracking algorithm is a complex function of algorithm structure and the scenario in which it is used. Section 3 describes how we evaluate simple bounds for the computational requirements of Multitracker. Finally, in Section 4, we quantify the inherent parallelism in Multitracker. Typical values of the speed-up achieved and the efficiency of processor utilization when the algorithm is adapted to a multiprocessor architecture are also provided in Section 4.

2. OVERVIEW OF MULTITRACKER ALGORITHM [6]

Multitracker uses the mathematical framework of Hybrid State Estimation to formulate the solution methodology for the multitarget tracking problem. The general hybrid state model consists of continuous-valued states and discrete-valued states. Using measurements related to the hybrid state, it is possible to compute an optimal (minimum-mean-squared or maximum-a-posteriori) estimate of the hybrid state. Variables in multiobject tracking algorithm can be identified with the generic hybrid model as follows: The state (generally position and velocity) of all existing targets constitutes the continuous-valued state; the noisy range, angle, and range-rate measurements from targets and clutter at every scan constitute the measurements;

*Research supported by the Naval Air Systems Command under Contract N00014-84-C-0378 with the Office of Naval Research.

**We define a task as the smallest unit of computation that may be assigned to a processor.

indicators for target status (straight line trajectory model, maneuver model) and measurement status (associated with target, false alarm) constitute the discrete-valued state.

The Hybrid State approach indicates the form of the optimal solution for the multiobject tracking problem; however, the postulation of all the possible values of the discrete state (referred to as a global hypotheses) and computing their likelihoods poses a difficult problem.

The track-oriented approach provides a systematic approach for generating the global hypotheses and computing their likelihoods. This approach maintains a set of target-trees and a list of global hypotheses. The root of each target tree represents the birth of the target and the branches represent the different dynamics that the target can assume and the various measurements it can be associated with in subsequent scans. A trace of successive branches from a leaf to the root of the tree corresponds to a potential track of the target. The leaf of each such trace is unique and is referred to as a track node of the target tree.

Each element of the global hypotheses list contains a set of pointers which point to track nodes. They represent the combination of track nodes postulated by the global hypothesis which that element represents. By assumption, the collection of pointers in any one such global hypothesis cannot point to two track nodes within the same target tree. This implies that there is at most one return per target per scan.

The track-oriented approach is a systematic methodology for constructing the optimal solution for multiobject tracking. However, for all practical scenarios which consist of several measurements in each scan, the computational requirements (both processing time and memory) of the algorithm will deplete the resources of any currently available computer. The reason for this problem is that the optimal algorithm postulates and retains all possible global hypotheses including the ones that are only remotely probable.

In order to construct a practical algorithm, all such unlikely global hypotheses have to eliminated. The key techniques incorporated in Multitracker are discussed below.

N-Scan Approximation: The optimal multiobject tracking algorithm requires that each branch of each target tree should be asssociated with each of the measurements in the scan, since all such associations may be possible. In reality, we know that each target should have only one branch corresponding to an association with the measurement it generates or to no association in the case it is not detected. An algorithm that waits N scans to resolve measurement associations in the current scan is referred to as an N-scan algorithm.

Gating: Gating is a screening technique that eliminates unlikely associations of measurements with targets. It proves to be very effective in cutting down the number of unlikely tracks and has been used in most tracking algorithms in the past. The gating process consists of constructing a region (gate) around a predicted target position, and choosing only those measurements which lie within this region to be associated with the target track.

Classification of Targets: A powerful screening technique that can be used at the global hypotheses generation stage involves the selection of only a group of targets while forming global hypotheses. The selection of targets is based on the criterion that the age of the target should be greater than α. Targets that fulfill this criterion will be referred to as Confirmed targets.

Since the set of Confirmed targets, included in any global hypothesis at any scan, should be a consistent set in that there exists no ambiguities in the assignment of measurements to targets, suitable rules should be formulated for promoting targets to the level of Confirmed targets. For this reason, and also to allow a variable value of α, we have found it convenient to define three other groups of targets. The first of these, having an age exactly equal to α, is referred to as Intermediate targets. It is from this group that Confirmed targets are selected. An Intermediate target is promoted to the status of a Confirmed target only if its presence does not cause any measurement assignment ambiguities with existing Confirmed targets. Targets with age 1 are referred to as Born targets. Each of the measurements received at a particular scan could potentially represent the birth of a new target.

The remaining targets, with ages between 2 and α-1, are referred to as Tentative targets. They represent a buffer group through which Born targets have to go through, before they get promoted to Intermediate targets. This form of grouping of targets by age is termed Classification.

Clustering: The computational complexity in a multiobject tracking algorithm arises mainly during the formation of global hypotheses. The larger the number of tracks that need to be considered, the larger the combinatorial problem. Since the purpose for forming global hypotheses is to resolve ambiguities in the assignment of measurements to targets, another form of grouping is possible. If targets lie in different regions of the surveillance area such that no common measurements are assigned to them, then obviously there is no need to look for measurement assignment ambiguities for those targets. This motivates the need for grouping targets, based on geographical locations. We will refer to such a grouping as Clustering.

3. COMPUTATIONAL REQUIREMENTS OF MULTITRACKER

Computational requirements of an algorithm may be specified in terms of operation count and memory requirements. For a recursive algorithm, such requirements computed for one iteration will be representative provided that the algorithm reaches some form of steady-state operation. Such a condition is certainly not achieved for the optimal multitarget tracking algorithm since the computational requirements grow exponentially with every iteration. However, for Multitracker, which incorporates all the screening and pruning features discussed in Section 2, close to steady-state operation is established.

Factors Which Influence Computational Requirements: Parameters that control screening and pruning in Multitracker will be referred to as Algorithm parameters. These are chosen essentially on the basis of the anticipated scenario. For example, the number of Confirmed targets accommodated by the algorithm should correspond to the maximum number of targets anticipated within the surveillance region; the number of tracks permitted for each target should take into account the clutter density, the proximity of other targets, and the probability of detection for targets. Similarly, the number targets and tracks per target permitted for Intermediate, Tentative, and Born targets should be based on target birth and death distributions and clutter distribution.

Algorithm parameters provide a limiting influence on the computational requirements. However, the actual requirements in a particular scenario become a complex function of these algorithm parameters in addition to the scenario parameters (which is defined by parameters such as the statistical distribution of the targets and clutter). Rather than relating these requirements to these scenario parameters, and also accounting for the influence of the algorithm parameters, we have chosen to determine bounds on the requirements that are automatically imposed by the algorithm parameters. Specifically, data structures defined for storing target tracks and global hypotheses screen and limit the computational requirements during an iteration of Multitracker. The resulting bounding analysis allows us to ignore the effect of the scenario parameters in determining computational requirements. However, it should be noted that the choice of algorithm parameters is based on anticipated scenario parameters, and this choice forms a crucial step in the design of Multitracker.

Algorithm Parameters That Bound Computational Requirements: Algorithm parameters have a bounding influence on both the operation count and the memory requirements of Multitracker since they restrict the number of targets, the number of tracks per target, and the number of global hypotheses. From the discussion provided in Section 2, it can be seen that there are several such algorithm parameters. We will discuss the pertinent ones below.

Classification allows different data structures to be defined for targets belonging to different age groups. Confirmed targets, having the largest age, are assigned a data structure which reflects the number of anticipated targets within the surveillance volume. The number of tracks permitted for each Confirmed target accounts for the associations with returns (correct or incorrect) received in each scan.

At the other extreme, Born targets, having the smallest age, are assigned a data structure which can accommodate all returns obtained in any scan as potential targets. Tentative and Intermediate targets are assigned data structures which allow the true targets to rise from Born to Confirmed category. The total number of tracks permitted may be computed from the number of targets (N_c, N_i, N_t, N_b) and the number of branches (B_c, B_i, B_t, 1) in each group.

In addition to maintaining tracks, Multitracker also maintains global hypotheses. The number of global hypotheses that can be formed in any scan is an exponential function of the number and length of Confirmed target tracks; this number is limited by the data structures defined for storing global hypotheses and past history of the tracks. The maximum number of global hypotheses is limited to NGH and the stored history of each target track is limited to NSCANs in the past.

Clustering enables groups of Confirmed targets to be processed independently. The number of subclusters, NS, and the number of targets in each subcluster, NTS, limit the processing requirements for clustering. The number of Connected clusters (NC) and the number of targets per Connected cluster (NTC) limit the processing requirements for global hypotheses formation.

Determination of Computational Requirements: The major computational effort in Multitracker is confined to three functional steps:

1. Track Predictions;
2. Track Updates; and
3. Track Prunning.

We will estimate the computational requirements for each of these steps. For this analysis we will confine attention to floating point multiplications, floating point additions (divisions are treated as multiplications and substractions are treated as additions), and integer number comparisons to determine the operation count. Further, we confine attention to target track storage needs since it forms the major portion of the memory requirements.

The prediction step involves predicting tracks of all targets. The set of operations is identical for all target tracks and involves a Kalman prediction* (time update) which requires

$1.5\ (n^3 + n^2)$ multiplications, and

$1.5\ (n^3 - n)$ additions

where n is the number of states modeled in the Kalman filter. Storage space required for each track is

$0.5\ n(n+1) + n + 1$	32 bit words
(NSCAN + 1)	16 bit words
1	4 bit word.

Since the prediction step involves a 1:1 transformation for each track, target tracks computed during the prediction step may be stored in the old track location. Additional storage is thus not required during this step. An obvious but important point about the prediction step is that each of the tracks can be predicted in parallel since each track depends on variables associated with that track alone.

Update of each track can be perceived to have two stages. First, all returns are screened (gated) against the track and a fixed number of them are selected for association with the track. Next, the track is updated using the selected returns. Setting up the gate for each track requires

$m(n^2 + 2n)$ multiplications

$m(n^2 + n - 1)$ additions

where m is the number of measurements in each return. If N_r denotes the number of returns per scan, gating of returns for each track requires

$N_r * m$ multiplications, and

$N_r * 2m$ additions.

The second stage of actually updating each track with each selected return represents a Kalman measurement update** requiring

$m(1.5n^2 + 4.5n + 1)$ multiplications, and

$0.5m(3n^2 + 5n)$ additions.

*The Kalman filter can be implemented either in the normal form or the factorized form. The computational requirements of both are about the same [7] and the requirements specified here correspond to the normal Kalman filter.

**We have assumed that the measurements in each return are updated sequentially, i.e., as m scalar measurement updates as opposed to a vector update.

Computation of the track likelihood involves

3m multiplications, and

m additions.

Assuming each existing track gets associated with an average of R returns in each scan, the number of operations per track gets multiplied by this factor. Updating each existing track with R returns represents a 1:R transformation for storage requirements. Generally, R is greater than unity and so additional storage has to be provided during this step.

There are several ways in which the update step can be executed when implemented on a multiprocessor architecture. For the gating stage, it is conceivable that all returns can be screened concurrently for all tracks*. However, from practical considerations (to avoid requiring too many processors for just this step), we will rule out this option and assume that each return is processed sequentially for each track. Even for the next stage of updating each track with selected returns, it is conceivable that all such updates can be done concurrently. Again, the alternative procedure of updating each track sequentially with each selected return is chosen from practical considerations.

After updating of all tracks, the next step is to prune the unlikely ones. There is no pruning for Born and Tentative target tracks, i.e., all updated tracks in these groups get promoted to the next higher group. On the other hand, Intermediate targets are promoted only if it does not create a conflict in the most likely global hypothesis of retained Confirmed targets. Hence, this step has to await the completion of Confirmed target pruning. Further, if promotion of targets from one group to the next moves tracks into locations used by the higher group (to minimize storage requirements), then this has to be done sequentially.

Pruning Confirmed targets involves the formation of global hypotheses and elimination of tracks not included in the most likely one. If target clustering is used, the formation of Connected clusters from Current clusters requires, at the most,

$$\frac{\text{NTS (NS (NS-1))}}{2}$$

16 bit word comparisons. After the Connected clusters are formed, the formation of global hypotheses requires

$$\left[\text{Min}\{(B_c+1)^{\text{NTC}},\ \text{NGH}\}\right]\left[\frac{\text{NTC (NTC+1) (2NTC+1) NM}}{6}\right]$$

16 bit comparisons and

$$\left[\text{Min}\{(B_c+1)^{\text{NTC}},\ \text{NGH}\}\right]\left[\text{NTC-1}\right] \text{ multiplications}$$

for each of the NC Connected clusters. Each of the Connected clusters can be processed in parallel.

Target tracks not included in the most likely global hypothesis are pruned away. The operation count for this stage of the pruning process requires only a few comparisons per track and we have chosen to neglect it. Based on the steady-state assumption, the pruning step represents an R:1 transformation in terms of storage requirements. Hence, we can expect the number of tracks per Confirmed target to reduce back to B_c.

Promotion of Intermediate targets is conditioned upon the available room in the Confirmed target data structure and can take place only after the pruning of Confirmed targets. Promotion of Tentative and Born targets to the next higher group is independent of the processing of the higher group. As mentioned earlier, storage requirements provided for each group dictates the sequence of operations. As in the case of Confirmed targets, the operation count for pruning and promotion of targets, and for the creation of Born targets is negligible and, for this preliminary analysis, we have chosen to ignore it.

4. ADAPTATION OF MULTITRACKER TO MULTIPROCESSOR ARCHITECTURES

Parallelism in a computational algorithm can be analyzed and exploited only if the flow of data, data dependencies among various tasks, and timing requirements during the execution of the algorithm are clearly understood. Data flow and data dependencies during the execution of an algorithm may be represented in the form of a directed graph. The nodes represent some task in the algorithm, the input arcs to a node represent the data required by the task, and the output arcs from a node represent the data produced by the task.

An algorithm generally exhibits parallelism at various levels of granularity at which operations can be defined. The instruction level can be thought of as the finest level of granularity. By aggregating operations at each level, coarser levels may be constructed. If the algorithm can be mapped on to a multiprocessor architecture that exploits parallelism down to the finest level of granularity, then it would appear that the maximum possible speed-up can be achieved. Unfortunately, the finer the granularity of operations, the larger will be the overhead requirements. For example, if scalar multiplications and additions associated with a function evaluation are distributed over several nodes, then scheduling these operations at the different nodes and synchronization of the data on each arc could require substantial time [8]. Clearly, there is an optimum choice of the granularity of operations at which parallelism in an algorithm should be identified.

In this section, we define the granularity of operations at a task level and evaluate the parallelism of Multitracker at that level. The achievable speed-up and the efficiency of processor utilization on a general multiprocessor architecture are then evaluated.

Parallelism in Multitracker: A directed flow graph summarizing the steps executed in one cycle of Multitracker is shown in Figure 1. In order to define a unit of operation (which we will call a task), we assume the following typical requirements for arithmetic operations.

Time for 32 bit multiplication	4 micro seconds
Time for 32 bit addition	2.6 micro seconds
Time for 16 bit comparison	1.3 micro seconds

*The question as to whether several processors can access the same data (memory contention) is not addressed here since it is conceivable that different processors could have their own private memories and commonly required data could be made available by broadcasting it.

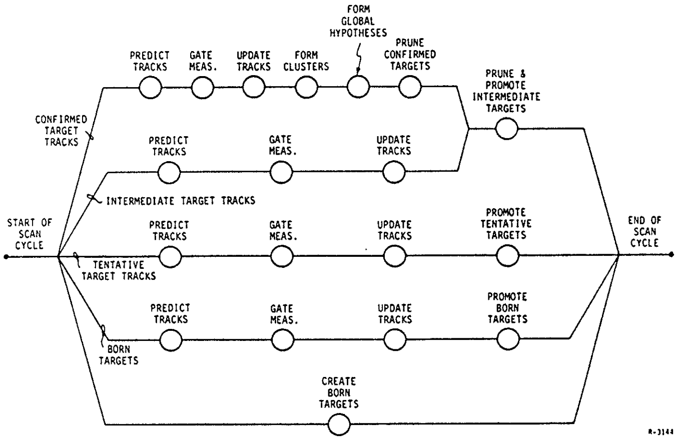


Figure 1. Directed Flow Graph of Operations in One Scan of Tracking Algorithm

Further, we assume the following algorithm parameters for Multitracker:

n	=	4
m	=	3
N_c	=	100
B_c	=	6
R	=	1.5
N_i	=	20
B_i	=	3
N_t	=	30
B_t	=	3
N_b	=	N_r = 100
NTS	=	2
NS	=	50
NTC	=	4
NC	=	25
NGH	=	100

The computational requirements for the various steps identified in Figure 1 may then be evaluated as:

Predict Confirmed Tracks:	600	*	714	micro sec.
Predict Intermediate Tracks:	60	*	714	micro sec.
Predict Tentative Tracks:	90	*	714	micro sec.
Predict Born Tracks:	100	*	714	micro sec.
Gate Confirmed Tracks:	600	*	3,196.2	micro sec.
Gate Intermediate Tracks:	60	*	3,196.2	micro sec.
Gate Tentative Tracks:	90	*	3,196.2	micro sec.
Gate Born Tracks:	100	*	3,196.2	micro sec.
Update Confirmed Tracks:	900	*	825	micro sec.
Update Intermediate Tracks:	60	*	825	micro sec.
Update Tentative Tracks:	90	*	825	micro sec.
Update Born Tracks:	90	*	825	micro sec.
Clustering:			3,185	micro sec.
Global Hypotheses Generation:	25	*	16,380	micro sec.

Total Time to Process a Scan of Reports = 4.68 seconds

We define a task as the operations associated with the Kalman prediction of one track. With this definition, the computational requirements of Multitracker are shown in Figure 2. Processing each group of targets is shown in separate paths since they can be processed concurrently.

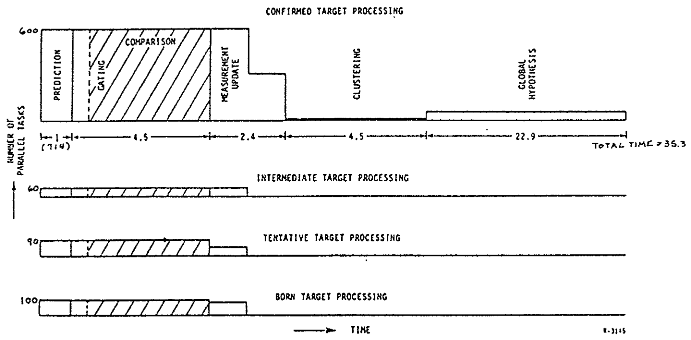


Figure 2. Representation of Parallelism in Multitracker

The maximum number of tasks is contained in the path processing Confirmed target tracks. Consistent with the assumptions made in Section 3, notice that gating of returns and update of each track with selected returns is carried out sequentially. In the terminology used for PERT analyses of networks, the critical path lies in the Confirmed target track processing path. Hence, the maximum possible speed-up of the algorithm implemented on a multiprocessor architecture will be controlled by the processing requirements of this path. To achieve this speed-up, tasks in the remaining paths should be processed concurrently with this path.

There are several ways of combining tasks from the various paths with the tasks in the critical path (without altering the sequence in any path). The easiest way would be to add tasks at each point in time for the various paths. However, this may not result in a combination which exhibits the maximum parallelism ratio (see below). In order to achieve the maximum parallelism ratio of an algorithm, the paths should be combined in such a manner so that we minimize the maximum number of tasks at any point in time. For a general problem, this combination represents a complex scheduling problem [9]. For the Multitracker problem depicted in Figure 2, it is easy to see that initiating the processing of Intermediate, Tentative, and Born targets with clustering of Confirmed targets will result in this optimum schedule. The set of superimposed tasks from all paths in either case is depicted in Figure 3.

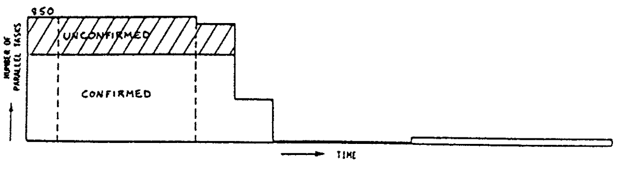


Figure 3a. Straightforward Combination of Tasks from All Paths

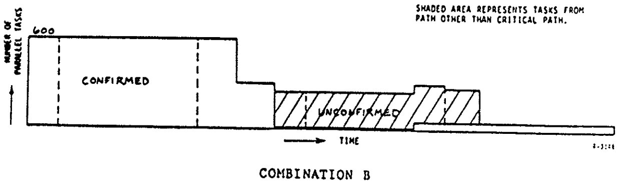


Figure 3b. Optimum Combination of Tasks from All Paths

For a computational algorithm consisting of two steps, parallelism has been expressed quantitatively as [10]

$$\rho = \frac{t_2 w_2}{t_1 + t_2 w_2} \qquad (4\text{-}1)$$

where

ρ is defined as the parallelism ratio,

t_2 is time interval containing parallel tasks,

w_2 is the number of parallel tasks,

t_1 is the time interval over which the algorithm is sequential.

In order to evaluate the parallelism for an algorithm which has a more general structure (such as the one for Multitracker), we define the parallelism ratio in a slightly modified form viz.,

$$\rho = \frac{\sum_{i=1}^{T} t_i w_i}{\underset{i}{\text{Max}}\,(w_i) \sum_{i=1}^{T} t_i} \qquad (4\text{-}2)$$

where T represents tht total number of steps in the critical path. The parallelism ratio for both combinations of paths in Figure 3 can be evaluated as

$\rho a = 0.22$, and

$\rho b = 0.31$.

A few comments can be made at this point. Inspite of the lower parallelism ratio of the first combination, the fact that all operations are the same for the parallel tasks in each time interval makes it attractive for implementation. For example, it is straightforward to implement this algorithm on an array processor where all processing elements execute the same set of instructions synchronously. The second combination requires a multiprocessor architecture wherein different processors can perform different functions at the same time. To achieve a better parallelism ratio in either case, the key lies in developing algorithms that can execute the clustering and the formation of global hypotheses in a concurrent fashion. The final comment relates to the step that imposes the major computational effort. In Figure 2 we have shaded the area which represents the gating operation and this obviously requires the major computational effort (58 percent). Developing efficient ways for gating (such as the use of associative memory processing) will help speed up the algorithm by

1. Exploiting the parallelism available in this step, and
2. Streamlining the processing requirements of comparing the returns with the allowed gates.

Performance of Multitracker on Multiprocessor Architectures: The key parameters that summarize the effectiveness of implementing a parallel algorithm on a multiprocessor architecture are the speed-up that is achievable and the efficiency of processor utilization. Both these parameters will obviously depend on how the algorithm is mapped onto a multiprocessor architecture.

Mapping of an algorithm onto a multiprocessor architecture is equivalent to graph isomorphism provided that both the algorithm and the architecture are represented as graphs [11]. A graph for Multitracker has been derived in the preceding subsection. Rather than deriving the graphs for specific computer architectures, we assume that a flexible architecture that can be tailored to the requirements of Multitracker is available; the only restriction we impose is that only a fixed number of processors is available. (We reiterate that the granularity of operations considered is at the task level).

Since we have assumed a flexible architecture, the mapping of Multitracker onto a multiprocessor will be controlled by the graph of the former. Assuming that a maximum of N processors are available in the architecture, let

T^1 be the number of steps require to execute Multitracker on the architecture.

p_i be the number of processors utilized during step i, and

t_i be the time required for step i. Then the achievable speed-up (η) of the algorithm is defined as

$$\eta = \frac{\sum_{i=1}^{T^1} p_i t_i}{\sum_{i=1}^{T^1} t_i} \qquad (4\text{-}3)$$

It is a measure of how fast the algorithm will run on a multiprocessor architecture compared to that on a uniprocessor architecture. The efficiency of processor utilization (E) is defined as the ratio of the number of tasks actually performed by the architecture and the number of tasks that the architecture is capable of performing, during the execution of the algorithm, i.e.,

$$E = \frac{\sum_{i=1}^{T^1} p_i t_i}{\underset{i}{\max}\,(p_i) \sum_{i=1}^{T^1} t_i} \qquad (4\text{-}4)$$

A comparison of Equations 4-2 and 4-4 shows that when the number of processors is matched to the maximum number of parallel tasks in the algorithm, the efficiency of processor utilization is the same as the parallelism ratio of the algorithm, i.e.,

$$E = \rho \; \left(\text{when } N = \underset{i}{\text{Max}}\,(w_i)\right) \qquad (4\text{-}5)$$

Furthermore, the individual p_i's are the same as the w_i's and so

$$N = \underset{i}{\text{Max}}\,(w_i) = \underset{i}{\text{Max}}\,(p_i) \qquad (4\text{-}6)$$

Hence,

$$\eta = NE = N\rho \qquad (4\text{-}7)$$

which states that the achievable speed-up is the product of the number of processors and the parallelism ratio of the algorithm. For the case where there is complete parallelism in an algorithm (ρ=1), Eq. 4-7 reduces to the ideal speed-up of N that is expected using N processors.

For the two algorithms shown in Figure 3, these performance measures may be evaluated as

$N_a = 850$ $N_b = 600$

$E_a = 0.22$ $E_b = 0.31$

$\eta_a = 186$ $\eta_b = 186$

In case the number of processors is less than the maximum number of parallel tasks in the algorithm $(N < \max_i(w_i))$, speed-up of the algorithm will reduce, but efficiency will improve. (The limiting case corresponds to N=1 where η=1 and E=1). For the general case, the speed-up and efficiency can be evaluated as follows:

Each of the steps in the graphs of Figure 3 which have more than N parallel tasks are divided into new steps. If we denote the larger of the two integers closest to $(\frac{w_i}{N})$ by $[\frac{w_i}{N}]_{\text{large}}$, then

$$E = \frac{\sum_{i=1}^{T} w_i * t_i}{N \sum_{i=1}^{T} [\frac{w_i}{N}]_{\text{large}} \quad t_i} \qquad (4\text{-}8)$$

and

$$\eta = \frac{\sum_{i=1}^{T} w_i \, t_i}{\sum_{i=1}^{T} [\frac{w_i}{N}]_{\text{large}} \cdot t_i} \qquad (4\text{-}9)$$

For example, if N = 400, then performance measures for the two algorithms in Figure 3 may be evaluated as

$E_a = 0.35$ $E_b = 0.39$

$\eta_a = 139$ $\eta_b = 157$

As anticipated, in both cases the efficiency of processor utilization improves but the speed-up is reduced. The decrease in speed-up for case a is worse because the first three tasks require a three-fold increase in processing time whereas in case b it requires only a two-fold increase.

5. SUMMARY AND CONCLUSIONS

We have computed the parallelism ratio of a model-based multiobject tracking algorithm (Multitracker) and evaluated the achievable speed-up and efficiency of processor utilitation when it is implemented on typical multiprocessor architectures. Prediction and update of target tracks may all be done in parallel and can be implemented on an array processor. However, the elimination of unlikely target tracks is not easily parallelizable and designing suitable architectures for implementing this step is the subject of current investigation. Our analysis shows that adapting Multitracker onto an array processor will provide the capability of tracking as many as 10,000 targets with an update rate (scan period) of 10 seconds.

REFERENCES

1. Bergland, G.D., and C.F. Hunnicutt, "Application of a Highly Parallel Processor to Radar Data Processing," IEEE Transactions on Aerospace and Electronic Systems, Vol. AES-8 No. 2, March 1972, pp. 161-162.

2. Batcher, K.E., "Bit-Serial Processing Systems," IEEE Transactions on Computers, Vol. C-31 No. 5, May 1982, pp. 377-384.

3. Summers, M.W., and D.F. Trad, "The Evolution of a Parallel Active Tracking Program," Proceedings of the 1974 Sagamore Computer Conference, Aug. 20-23, 1974, pp. 238-249.

4. Bar-Shalom, Y., "Tracking Methods in a Multiobject Environment," IEEE Transactions on Automatic Control, Vol. AC-23, August 1978, pp. 618-626.

5. Reid, D.B., "An Algorithm for Tracking Multiple Targets," IEEE Transactions on Automatic Control, Vol. AC-24, No. 6, December 1979, pp. 843-854.

6. Kurien, T., and R.B. Washburn, "Multiobject Tracking Using Passive Sensors," Proceedings of the 1985 American Control Conference, Boston, MA, June 19-21, 1985, pp. 1032-1038.

7. Bierman, G.J., Factorization Methods for Discrete Sequential Estimation, Academic Press, New York, 197 .

8. Gajski, D.D., and J.K. Peir, "Essential Issues in Multiprocessor Systems," IEEE Computer, June 1985, pp. 9-26.

9. Coffman, E.G., and P.J. Denning, Operating Systems Theory, Prentice Hall, N.J., 1973 (Chapter 2).

10. Baer, Jean-Loup, Computer Systems Architecture, Computer Science Press, Maryland, 1980.

11. Bokhari, S.H.," On the Mapping Problem," IEEE Transactions on Computers, Vol. C-30, No. 3, March 1981, pp. 207-214.

A PROBABILISTIC / POSSIBILISTIC APPROACH TO MODELING C^3 SYSTEMS

I.R. Goodman

Command & Control Department
Code 421
NAVAL OCEAN SYSTEMS CENTER
San Diego, California 92152

ABSTRACT

This paper continues the development of a general model for C^3 systems based upon possibilistic as well as probabilistic considerations, as presented earlier[1] The main approach is a microscopic one, building the model up from basic variables, such as node states, detection states, hypotheses, algorithms available, responses, and received inputs. These variables are connected quantitatively by use of general conditioning techniques with minimal assumptions. Explicit relations are derived connecting ten primitive relations between variables and node state outputs. Implementation issues are also discussed.

1. INTRODUCTION

The history of C^3 analysis as an organized approach to defining the general military problem - or at least the command aspects - goes back several years.(For a brief history of approaches throughout the MIT/ONR Workshop on C^3 Systems, see Goodman [1].) Despite the large amount of literature produced involving C^3 issues- whether it be from the C^3 Workshop or the myriad Government and private industry publications - a basic pattern emerges: little attention has been paid to establishing an overall C^3 model from a quantitative "bottoms-up"(or microscopic) viewpoint. Instead, much of the work has been devoted to either qualitative analysis or quantitative investigations of small portions of the general C^3 problem, or to macroscopic "top-down" approaches. Examples of the first two types of analysis are numerous. Perusing through the last several issues [2-5] of the annual MIT/ONR Workshop Proceedings, one finds articles on command planning, fire control, tracking, correlation, surveillance, limited interactive multi-person decision games, time studies, stochastic control problems, etc. Examples of the last-type are not as plentiful, but include papers on markovian models of C^3 systems, variations of Lawson's macro-thermodynamic analogue approach, Lanchester's attrition equations and generalizations, use of general resource allocation principles, and even use of analogues with laws governing large scale systems in other fields of interest such as economics and natural language. Of course, this is not to say that these efforts do not directly contribute greatly to our knowledge of C^3 system behavior; it is still necessary to carry out this work in order to produce an eventual cohesive general theory of C^3 systems. Nevertheless, it is the thesis of this paper that it is not too early in the development of C^3 theory to attempt to produce a microscopic approach.

The ongoing work of Levis et al. [6-8], for at least the intranodal or localized (but interacting) decision maker aspect is a microscopic approach, building upon pieces of information transmittal and processing into an overall organization model. In some sense, this paper has been influenced by this philosophy. Another contributing source for this work has been the establishment of models which integrate both stochastic and linguistic-based information into posterior descriptions of parameters of interest- in particular, correlation or data association levels between track histories. (See Goodman [9-11] and the more general work of Goodman and Nguyen [12].) Finally, this overall C^3 model may be perceived as the more general C^3 analogue of the author's previous microscopic approach to the multi-target correlation problem [1],[13],[14].

The objectives of this work consist of the following:

(1) Derive from first principles a microscopic model

(2) Model is to be quantitative in nature resulting in an implementable algorithm whose inputs are measures of initial C^3 node states and all relevant time evolving primitive relations between variables and whose outputs are updated node states which can be used in in turn to compute measures of performance or measures of effectiveness.

(3) Model is to incorporate both linguistic and stochastic type information, analogous to the PACT approach in [9-11]. (PACT = Possibilistic Approach to Correlation and Tracking)

(4) Resulting algorithm is to be testable for feasibility of implementation and as a decision aid for real world situations.

(5) Algorithm should be useful as a yardstick in some sense comparing, contrasting, and analyzing other C^3 models.

2. BASIC APPROACH

The basic approach taken here is to view the general warfare problem as a collection of possibly overlapping (or nested or related in some hierarchical way) C^3 systems divided among adversaries and allies. For purpose of simplicity, it is assumed throughout that only two C^3 systems are present: one friendly, one adversary Each C^3 system is identified as a network of nodes (N variable with appropriate subscripts) which transmits responses (R variable with appropriate subscripts) including the possibility of ordinary signals containing information or actual weapons fired, and which receives "signals" corresponding to the responses (S variable, similarly), possibly distorted by the medium and interfered with by additive noise.

Each node is roughly one of two types: a decision maker (DM) complex or a follower (FL) complex. Examples of the former include various echelon commanders as well as non-human information coordination or tranfer points such as automatic sensing and automatic decision making groups. Examples of the latter include battalions of land troops directed in part by other sources, formations of airplanes, and various weapon systems, though based on the approach taken, often weapon systems can be modeled as part of the response/"signal" between nodes. Both DM and FL type nodes may be modeled internally by a basically similar structure which differs in time delays between component processors and the specific forms of the processors, but which retains the same fundamental design : the SHOR (Sense,Hypothesize, Options,Response) paradigm or related paradigms. (See e.g. Wohl et al. [15] or Levis [6].)

Each node is assumed to follow the basic input-output mode utilizing both linguistic and probabilistic information, where necessary, in its internal processing of information.following the reception of a "signal". The response by the node is assumed in general to cause a change in the state of the original node, such as attrition occurring when men are sent out or weapons fired or state of knowledge about the remaining nodes is changed. Of course, the node could also be changed almost immediately by the incoming "signal" before any processing if it represents a destructive weapon arriving. Responses can be vacuous - as could incoming signals stopped by the medium or just going undetected- reflecting stalling tactics, protocol, or other factors;or responses can be multi-directional, going out to several nodes, friendly or adversary. Included in the assumptions are situations where leaks occur in transmission, spying occurs, replenishment of forces takes place, and more generally, any linking of a node with others through input-output relations, where for convenience "signals" are processed sequentially.

Each node is identified with a state vector. This includes the node's own equations of motion, threat level reflecting physical plant, force size,and any other relevant measures of physical or geographical nature. In addition, the node state vector includes a knowledge component where the mental state and knowledge of the node about the remaining nodes are listed. This includes the algorithm set available to the node for response. The overall behavior of a C^3 system can be defined in terms of the join of its individual node state vectors. All of these descriptions are dynamic and can change considerably in design as time progresses.

The analysis procedes according to two basic divisions: intranodal, reflecting the SHOR-like paradigm that a node goes through in processing a "signal", involving the initial node state, incoming "signal", and detection, hypotheses, algorithm choice, and output/ response variables; internodal, reflecting the change occurring for a response sent by one node to another, received as a "signal", involving variables representing additive error and medium distortion. Quantitatively, the posterior or averaged node state vector possibility or probability function is computed separately for each node. Under mild sufficiency conditions, it is shown that these functions are (at least theoretically) computable functions of ten primitive relations between variables ;each such relation corresponds to a conditional (or in some cases a non-conditional) possibility or probability function among certain of the variables. The entire derivation uses the simple technique of representing the possibility(or probability) function of a variable of interest - in this case any node state vector - in terms of an iterated disjunction (or integral) of conjunctions (or products) of conditional possibility (or probability) functions of auxiliary variables - in this case the variables discussed above, including "signal", response, detection, hypotheses,etc. (Again, see [12] for background and general discussion of possibility functions and representation of natural language information.)

3. NOTATION

Each C^3 system is denoted as C^3_a , where a=1 corresponds to the friendly one and a=2 corresponds to the adversary. Further, each C^3 system's time evolution is represented by the notation

$$C^3_a = (C_{a,t})_{t\ge0} \quad , \qquad (1)$$

where C^3 system a at t is denoted by $C^3_{a,t}$.

From now on, assume time index t occurs discretely, corresponding to initial reception time of a "signal" by a node or to the time of node response following "signal" processing only. These times are put in the natural order

$$0=t_0 < t_1 < t_2 <\cdots \qquad (2)$$

Thus t_5 could correspond to the time when say node 7 in C^3_1 has just finished processing its received "signal" and is ready to respond to some other nodes, while t_6 could apply to that response being received as a new "signal" by node 15 in C^3_1. It is assumed that successive times correspond to a node's input-output cycle, if the initial time corresponds to the input time.

For simplicity, all possibility and probability functions will be denoted by the common symbols p() and p(|) (the latter to denote conditioning) and all operations relative to possibility theory will be denoted by simple probabilistic counterparts. Thus all iterated disjunctions will be denoted by a summation (Σ) or integral (∫), depending on the context.

At any given time a C^3 system is identified with the triple

$$C^3_{a,t} = (N_{a,t} \, , R_{a,t} \, , S_{a,t}) \, , \qquad (3)$$

where $N_{a,t}$ is the set of all nodes $N_{a,i,t}$ of $C^3_{a,t}$, $R_{a,t}$ is the set of all responses of $C^3_{a,t}$, and $S_{a,t}$ is the set of all "signals" of $C^3_{a,t}$. Similarly, the set of all nodes throughout time of $C^3_{a,t}$ is denoted as N_a , with similar designations for R_a and S_a. At any time the set of all nodes is N_t , that of all "signals" is S_t , that of all responses is R_t. Thus

$$N = (N_1,N_2),\ N_a = (N_{a,t})_{t\ge0},\ N_{a,t}=(N_{a,i,t})_{i=1,2,..} \qquad (4)$$

$$R = (R_1,R_2),\ R_a = (R_{a,t})_{t\ge0},\ R_{a,t}=(R_{a,i,t})_{i=1,2,..} \qquad (5)$$

$$S = (S_1,S_2),\ S_a = (S_{a,t})_{t\ge0},\ S_{a,t}=(S_{a,i,t})_{i=1,2,..} \qquad (6)$$

$$N_t = (N_{1,t},N_{2,t}) \qquad (7)$$

$$R_t = (R_{1,t},R_{2,t}) \qquad (8)$$

$$S_t = (S_{1,t},S_{2,t}) \qquad (9)$$

$$C = (C_1,C_2) \ . \qquad (10)$$

Also when there is no ambiguity in meaning, the notation $N_{g,k}$ will replace N_{a,i,t_k} , where g is identified with (a,i). Similar remarks hold for $R_{g,k}$ and $S_{g,k}$. Sometimes the index variable h will be used for g. Also, denote the initial time(t_0) values for nodes, responses, and "signals" as N_0,R_0,S_0, respectively.

Additional variables include:

$W_{g,k}$, possible true node source which yielded "signal" $S_{g,k}$(to $N_{g,k}$ at t_k) at t_{k-1}. The range values of $W_{g,k}$ are in the index set of all nodes at t_{k-1}

$$I_{k-1}=\{(1,1,k-1),(1,2,k-1),\ldots,(2,17,k-1),\ldots\} \ ; \qquad (11)$$

$D_{g,k}$, possible detection state for $N_{g,k}$ relative to $S_{g,k}$, where as usual

$$D_{g,k} = \begin{cases} 1 \leftrightarrow \text{detection of "signal" } S_{g,k} \\ 0 \leftrightarrow \text{no detection of "signal } S_{g,k} \ ; \end{cases} \qquad (12)$$

$H_{g,k}$, hypotheses chosen by $N_{g,k}$ when processing $S_{g,k}$. Typical range values for $H_{g,k}$ are in $H_{(g,k)}$, where

$$H_{(g,k)} = \mathcal{P}(H'_{(g,k)}) \ , \qquad (13)$$

where $\mathcal{P}()$ is the power class - or class of all subsets - operator and the basic hypotheses set for $N_{g,k}$ is $H'_{(g,k)}$ where typically,

$$H'_{(g,k)}=\{..,\text{"enemy is now attacking","am being given retreat command"},..,\text{"just hit by bomb"},..\}; \qquad (14)$$

$F_{g,k}$,set of algorithms chosen following decision about which hypotheses (or hypothesis) holds. Typical values of $F_{g,k}$ are in $F_{(g,k)}$, where

$$F_{(g,k)} = \mathcal{P}(F'_{(g,k)}) \ , \qquad (15)$$

where

$$F'_{(g,k)}=\{..,\text{weapon firing alg. }17,..,\text{control eq.for own maneuvering }28,..,\text{flank ing technique }6,...,\text{correlation alg. }3,...\}; \qquad (16)$$

$G_{h,g,k}$, internodal transmission distortion function of $R_{h,k-1}$ from $N_{h,k-1}$ to $N_{g,k}$ contributing to the "signal" $S_{g,k}$;

$Q_{h,g,k}$, internodal transmission additive noise/error for $R_{h,k-1}$ going from source $N_{h,k-1}$ and contributing to $S_{g,k}$ being received by $N_{g,k}$ at t_k.

The range values of $R_{h,k}$ and $S_{g,k}$ are determined through a regression relation given in (17).

Further notation will be introduced as needed.

4. ANALYSIS

The basic internodal analysis is developed via the additive nonlinear regression relation

$$(S_{g,k+1}|W_{g,k+1}=(h,k))=G_{h,g,k+1}(R_{h,k})+Q_{h,g,k+1}, \qquad (17)$$

noting the dimension of $S_{g,k+1}$ need not match that of $R_{h,k}$, i.e., some of the response can get lost. Typical range values of $S_{g,k+1}$ are in

$$S_{(g,k+1)}=\{...,\text{ordinary signal type 46- message from commander},..,\text{ordinary signal type 135- message from enemy node intercepted},..,\text{incoming potentially destructive weapon 7 incoming weapon (missile)19},..\} . \qquad (18)$$

Similar remarks hold for $R_{h,k}$, for all g,h,k.

For the intranodal analysis, the SHOR paradigm is followed, where $S_{g,k}$ is detected or not ($D_{g,k}$=1,0), followed by choice of $H_{g,k}$, in turn - possibly using consultations with other nodes - by choice of $F_{g,k}$ and finally by $R_{g,k+1}$, resulting in the change of original $N_{g,k}$ to $N_{g,k+1}$. At this point it is appropriate to introduce some additional subvariables. The node state is partitioned into

$$N_{g,k} = \left(\begin{array}{c} M_{g,k} \\ \hline K_{g,k} \end{array}\right), \qquad (19)$$

where $M_{g,k}$ represents the state vector proper part of the node, while $K_{g,k}$ represents the knowledge portion, where, in turn,

$$M_{g,k} = (ID_{g,k},CL_{g,k},IM_{g,k},EQ_{g,k},TH_{g,k},DA_{g,k}), \qquad (20)$$

where $ID_{d,k}$ is identity;$CL_{g,k}$ is class; $IM_{g,k}$ is military importance of $N_{g,k}$, as a function of number of troops left, physical condition, location,..; $EQ_{g,k}$ represents the node's own equations of motion, including any maneuvering present, lack of motion,etc., all suitably updated; $TH_{g,k}$ and $DA_{g,k}$, related to $IM_{g,k}$ represent threat and damage levels of $N_{g,k}$, again as possible vector or scalar-valued functions of many contributing subcomponents. In addition,

$$K_{g,k} = (\hat{N}_{(g,k)},F_{(g,k)}) \ , \qquad (21)$$

where $F_{(g,k)}$, the algorithm set,is discussed in (15) and (16) and $\hat{N}_{(g,k)}$ is the estimate of the remaining nodes of both C^3 systems by $N_{g,k}$ at t_k. Using obvious notation

$$\hat{N}_{(g,k)}= (\hat{N}_{h,k\ (g,k)})_{\text{all } h} \ , \qquad (22)$$

where some or all of the entries $\hat{N}_{h,k\ (g,k)}$ may be vacuous (Ø).

In a related vein, one could extend this simplified view of knowledge concerning one node with respect to another to include the estimate by $N_{h,k}$ of $\hat{N}_{h,k(g,k)}$

$$\hat{\hat{N}}_{h,k\ (g,k)(h,k)} =\left(\widehat{\hat{N}_{h,k\ (g,k)}}\right)_{(h,k)}, \qquad (23)$$

and further the estimate of the above by(g,k), etc. One way of treating this apparently hopeless infinite nesting of mental states is through fixed-point theory. Assuming the operation in change of knowledge from (h,k) to (g,k) is stable and similarly for (h,k) estimating (g,k)-behavior, writing g for (g,k) estimating (h,k) and conversely h simply for (h,k) estimating (g,k), one quickly sees that the infinite iterated estimation beginning with (22) is no more than

$$q = \cdots\circ g\circ h\circ g\circ h \cdot\cdot h\circ g \quad , \qquad (24)$$

where ∘ denotes functional composition. But (24) can be regrouped into the infinite iterate

$$q = \cdots\psi\circ\psi\circ\psi\circ\cdots\circ\psi\circ\psi \ , \qquad (25)$$

where

$$\psi = h\circ g \ . \qquad (26)$$

Then by applying both sides of (25) to ψ, one obtains the basic fixed-point relation

$$q\circ\psi = q \ , \qquad (27)$$

which can be further analyzed through standard techniques, which will not be persued further here. (See for example [16].)

Next, define the following ten relations among variables introduced, assumed obtainable for all h,g, k :

INTRANODAL

$$(1)_{g,k} = p(H_{g,k}|D_{g,k},S_{g,k}) \ , \qquad (28)$$

$$(2)_{g,k} = p(F_{g,k}|H_{g,k}) \ , \qquad (29)$$

$$(3)_{g,k+1}= p(R_{g,k+1}|F_{g,k},S_{g,k},N_{g,k}), \qquad (30)$$

$$(4)_{g,k+1}= p(N_{g,k+1}|R_{g,k+1},N_{g,k}) \ , \qquad (31)$$

$$(5)_{g,k} = p(D_{g,k}|S_{g,k}, N_{g,k}) \ ; \qquad (32)$$

INTERNODAL

$$(6)_{h,g,k+1} = p(Q_{h,g,k+1}) \text{ with } G_{h,g,k+1}, \qquad (33)$$

$$(7)_{h,g,k+1} = p(W_{g,k+1} = h | N_0); \qquad (34)$$

PRIOR/INITIAL TIME

$$(8)_0 = p(N_0) \quad , \qquad (35)$$

$$(15)_{h,g,0} = p(R_{h,0} | W_{g,1} = h \; , \; N_0), \qquad (36)$$

$$(16)_{g,0} = p(S_{g,0} | N_0) \; . \qquad (37)$$

The above functions are denoted as primitives.

Assume the following sufficiency and independence relations among the variables:

$$(1)_{g,k} = p(H_{g,k} | D_{g,k}, S_{g,k}; N_{g,k}, N_{h,j}, N_0, W_{h,j} = g), \qquad (38)$$

$$(2)_{g,k} = p(F_{g,k} | H_{g,k}; D_{g,k}, S_{g,k}, N_{g,k}, N_{h,j}, N_0, W_{h,j} = g), \qquad (39)$$

$$(3)_{g,k+1} = p(R_{g,k+1} | F_{g,k}, S_{g,k}, N_{g,k}; D_{g,k}, H_{g,k}, N_{h,j}, N_0, W_{h,j} = g), \qquad (40)$$

$$(4)_{g,k+1} = p(N_{g,k+1} | R_{g,k+1}, N_{g,k}; D_{g,k}, S_{g,k}, N_{h,j}, N_0, W_{h,j} = g), \qquad (41)$$

$$(5)_{g,k} = p(D_{g,k} | S_{g,k}, N_{g,k}; N_{h,j}, N_0, W_{h,j} = g), \qquad (42)$$

$$(6)_{h,g,k+1} = P(Q_{h,g,k+1} | N_{g,k}, N_0, W_{g,k+1} = h), \qquad (43)$$

$$(7)_{h,g,k+1} = p(W_{g,k+1} = h | N_0; N_{g,k}) \; . \qquad (44)$$

Thus, for example, $(3)_{g,k+1}$ can be interpreted as:

> p($N_{g,k}$ decides to respond with initial output response $R_{g,k+1}$ following intranodal processing of $S_{g,k}$, where $F_{g,k}$ was chosen) ,

while $(4)_{g,k+1}$ is naturally interpreted as:

> p(new node state is $N_{g,k+1}$ just following the transmission of the response $R_{g,k+1}$, given the old node state was $N_{g,k}$) .

Most of the remaining relations have also obvious interpretations in terms of the intranodal or internodaspect of the C^3 systems.

The above ten primitive functions, together with their simplifying conditions must be known in order to compute the basic output for all g,k

$$(19)_{g,k} = p(N_{g,k}) \; , \qquad (45)$$

the marginal possibility or probability function of node state $N_{g,k}$ averaged over all relevant variables.

This output is determined through a series of operations upon the ten primitive functions- essentially products and integrals or sums for the probability case and conjunction and iterated disjunction operations for the possibility case. (See the comments prior to (3).) These operations are spelled out in the following 11 computations leading to the evaluation of $(19)_{g,k}$:

$$(9)_{g,k+1} = p(R_{g,k+1} | D_{g,k}, S_{g,k}, N_{g,k})$$

$$= \iint\limits_{(\text{over all } F_{g,k}, H_{g,k})} (1)_{g,k} \cdot (2)_{g,k} \cdot (3)_{g,k+1} \, dF_{g,k} \, dH_{g,k} \; , \qquad (46)$$

$$(10)_{g,k+1} = p(N_{g,k+1} | D_{g,k}, S_{g,k}, N_{g,k})$$

$$= \int\limits_{\left(\substack{\text{over all} \\ R_{g,k+1}}\right)} (4)_{g,k+1} \cdot (9)_{g,k+1} \, dR_{g,k+1} \; , \qquad (47)$$

$$(11)_{g,k+1} = p(R_{g,k+1} | S_{g,k}, N_{g,k})$$

$$= \sum_{D_{g,k}=0}^{1} ((9)_{g,k+1} \cdot (5)_{g,k}) \; , \qquad (48)$$

$$(12)_{g,k+1} = p(N_{g,k+1}, D_{g,k} | S_{g,k}, N_{g,k})$$

$$= (10)_{g,k+1} \cdot (5)_{g,k} \; , \qquad (49)$$

$$(13)_{g,k+1} = p(N_{g,k+1} | S_{g,k}, N_{g,k})$$

$$= \sum_{D_{g,k}=0}^{1} ((12)_{g,k+1}) \; , \qquad (50)$$

$$(14)_{h,g,k+1} = p(S_{g,k+1} | R_{h,k})$$

$$= p(Q_{h,g,k+1} = S_{g,k+1} - G_{h,g,k+1}(R_{h,k})), \qquad (51)$$

$$(15)_{h,g,k} = p(R_{h,k} | N_0, W_{g,k+1} = h)$$

$$= \iint\limits_{\left(\substack{\text{over all} \\ S_{h,k-1}, N_{h,k-1}}\right)} (11)_{h,k} \cdot (16)_{h,k-1} \cdot (18)_{h,k-1} \, dS_{h,k-1} \, dN_{h,k-1} , \qquad (52)$$

$$(16)_{g,k} = p(S_{g,k} | N_0)$$

$$= \iint\limits_{\left(\substack{\text{over all} \\ h, R_{h,k-1}}\right)} (14)_{h,g,k} \cdot (15)_{h,g,k-1} \cdot (7)_{h,g,k-1} \, dh \, dR_{h,k-1} , \qquad (53)$$

$$(17)_{g,k+1} = p(N_{g,k+1} | N_{g,k}, N_0)$$

$$= \int\limits_{\left(\substack{\text{over all} \\ S_{g,k}}\right)} (13)_{g,k+1} \cdot (16)_{g,k} \, dS_{g,k} \; , \qquad (54)$$

$$(18)_{g,k} = p(N_{g,k} | N_0) = \int\limits_{\left(\substack{\text{over all} \\ N_{g,k-1}}\right)} (17)_{g,k} \cdot (18)_{g,k-1} \, dN_{g,k-1} , \qquad (55)$$

$$(19)_{g,k} = \int\limits_{\left(\substack{\text{over all} \\ N_0}\right)} (18)_{g,k} \cdot (8)_0 \, dN_0 \; . \qquad (56)$$

For later reference, denote the set of all primitive functions up to some time t_{k_0} as

$$\text{PRIM}_{k_0} = ((1)_{g,k}, (2)_{g,k}, \ldots, (7)_{h,g,k+1}, \ldots, (16)_{g,0})_{\substack{\text{all } g,h \\ k \le k_0}}$$

$$= (\text{PRIM}^{(1)}_{k_0}, \text{PRIM}^{(2)}_{k_0}) \; , \qquad (57)$$

where $\text{PRIM}^{(1)}_{k_0}$ is the vector of functions consisting of all intranodal functions given in (28)-(32) and the two priors given in (35),(37), for all g and all $k \le k_0$; $\text{PRIM}^{(2)}_{k_0}$ is the vector of functions consisting of the two internodal functions given in (33),(34) and the prior given in (36), for all g,h and all $k \le k_0$.

The above results may be summarized as follows:

Theorem 1.

Suppose that PRIM_{k_0} is known, i.e., has a known functional form for each of its components. Then for all g and $k \le k_0$ $(19)_{g,k}$ is a computable function of PRIM_k,

$$(19)_{g,k} = \mathcal{F}_{g,k}(\mathrm{PRIM}_k), \qquad (58)$$

where $\mathcal{F}_{g,k}$ is a functional consisting of formal products, sums, and integrals (with the usual interpretation for the possibilistic case) given in sequential form in (46)-(56).

Proof: Consider only the probability case. (Again, see [12] for the possibilistic analogue of the technique used here.) By representing a probability distribution function as the integral of products of suitably chosen conditional probability distribution functions, all results follow, beginning with (56), going back to (55), and then back to (55),(54),..., until (46)-(56) have all been used. ∎

A comprehensive flow chart connecting the primitive function inputs with the marginal node state distributions as outputs through, in effect, $\mathcal{F}_{g,k}$ is given in Figure 1.

5. DISCUSSION

Although it would be more desirable to obtain the entire joint distribution of node states at a given time (averaged over all relevant variables), rather than the separate marginel node states as given here, such joint forms appear to lead to intractable calculations. Indeed the calculations for the model proposed here grow exponentially and pruning techniques must be established for implementations - analogous to the difficulty in implementing overall tracking-correlation models [13].

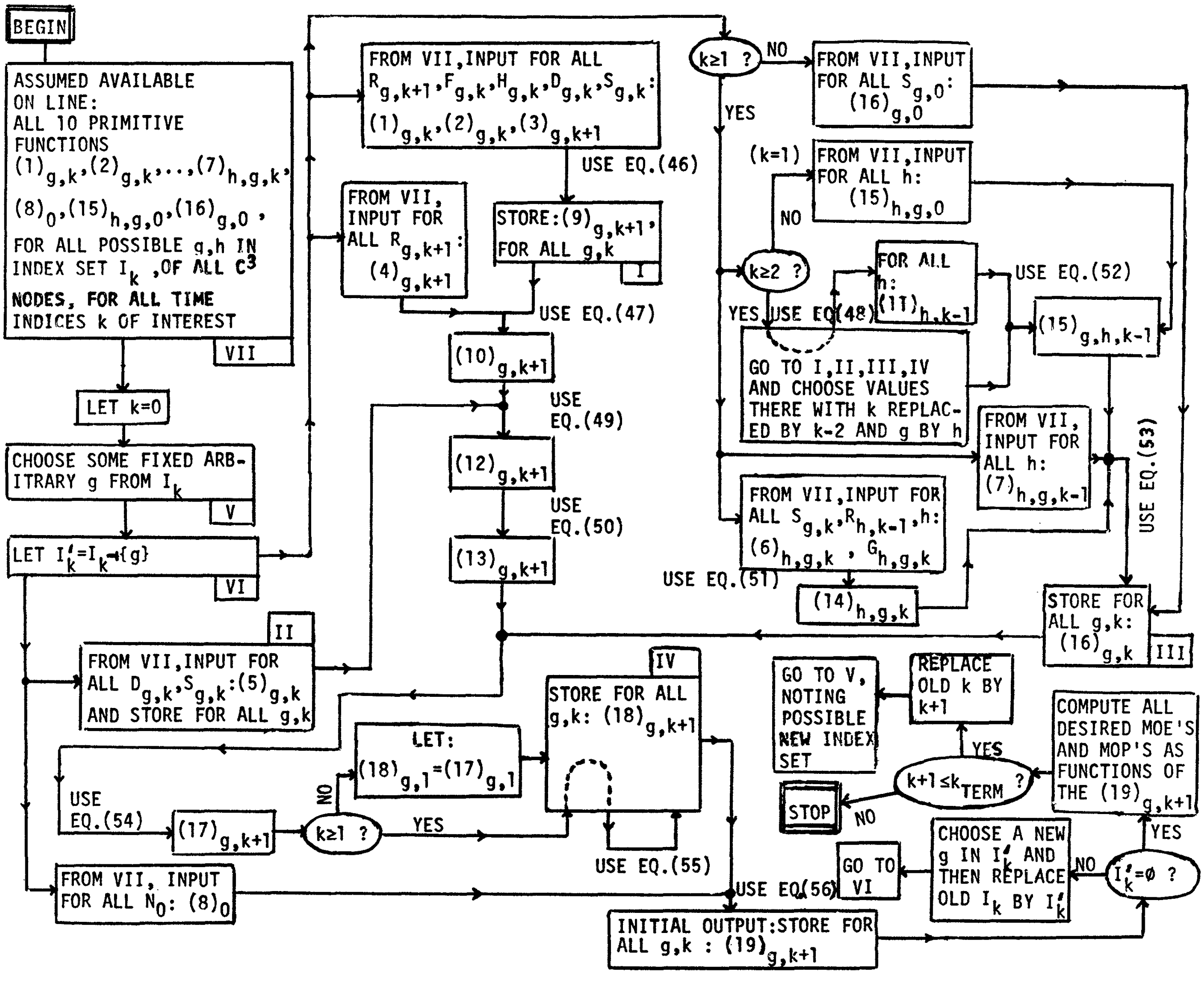
Nevertheless, a relatively large class of measures of performance (mop's) or of effectiveness (moe's) can be computed once the marginal node state distributions are known. For example, the following measures for each C^3 system $C^3_{a,k}$ are useful:

$$\overline{IM}_{a,k} = (1/J_{a,k}) \sum_{i=1}^{J_{a,k}} E(IM_{a,i,k})$$

= averaged measure of importance, (59)

$$\overline{TH}_{a,k} = (1/J_{a,k}) \sum_{i=1}^{J_{a,k}} E(TH_{a,i,k})$$

= averaged measure of threat , (60)



BEGIN
ASSUMED AVAILABLE ON LINE: ALL 10 PRIMITIVE FUNCTIONS $(1)_{g,k},(2)_{g,k},\ldots,(7)_{h,g,k},(8)_0,(15)_{h,g,0},(16)_{g,0}$, FOR ALL POSSIBLE g,h IN INDEX SET I_k, OF ALL C^3 NODES, FOR ALL TIME INDICES k OF INTEREST
VII
LET k=0
CHOOSE SOME FIXED ARBITRARY g FROM I_k
V
LET $I_k'=I_k-\{g\}$
VI
FROM VII,INPUT FOR ALL $R_{g,k+1},F_{g,k},H_{g,k},D_{g,k},S_{g,k}$: $(1)_{g,k},(2)_{g,k},(3)_{g,k+1}$
USE EQ.(46)
FROM VII, INPUT FOR ALL $R_{g,k+1}$: $(4)_{g,k+1}$
STORE: $(9)_{g,k+1}$, FOR ALL g,k
I
USE EQ.(47)
$(10)_{g,k+1}$
USE EQ.(49)
$(12)_{g,k+1}$
USE EQ.(50)
$(13)_{g,k+1}$
II
FROM VII,INPUT FOR ALL $D_{g,k},S_{g,k}$: $(5)_{g,k}$ AND STORE FOR ALL g,k
USE EQ.(54)
$(17)_{g,k+1}$
k≥1 ?
NO
YES
LET: $(18)_{g,1}=(17)_{g,1}$
IV
STORE FOR ALL g,k: $(18)_{g,k+1}$
USE EQ.(55)
FROM VII, INPUT FOR ALL N_0: $(8)_0$
USE EQ.(56)
INITIAL OUTPUT: STORE FOR ALL g,k : $(19)_{g,k+1}$
k≥1 ?
NO
FROM VII,INPUT FOR ALL $S_{g,0}$: $(16)_{g,0}$
YES
(k=1)
FROM VII,INPUT FOR ALL h: $(15)_{h,g,0}$
NO
k≥2 ?
YES
USE EQ.(48)
FOR ALL h: $(11)_{h,k-1}$
USE EQ.(52)
$(15)_{g,h,k-1}$
GO TO I,II,III,IV AND CHOOSE VALUES THERE WITH k REPLACED BY k-2 AND g BY h
FROM VII, INPUT FOR ALL h: $(7)_{h,g,k-1}$
USE EQ.(53)
FROM VII,INPUT FOR ALL $S_{g,k},R_{h,k-1},h$: $(6)_{h,g,k}$, $G_{h,g,k}$
USE EQ.(51)
$(14)_{h,g,k}$
STORE FOR ALL g,k: $(16)_{g,k}$
III
GO TO V, NOTING POSSIBLE NEW INDEX SET
REPLACE OLD k BY k+1
YES
$k+1 \le k_{TERM}$?
STOP
NO
COMPUTE ALL DESIRED MOE'S AND MOP'S AS FUNCTIONS OF THE $(19)_{g,k+1}$
YES
$I_k'=\emptyset$?
NO
CHOOSE A NEW g IN I_k' AND THEN REPLACE OLD I_k BY I_k'
GO TO VI

Figure 1. Flow Chart Depicting Input-Output Computations for C^3 Model

$$\overline{ENT}_{a,k} = \sum_{i=1}^{J_{a,k}} ENT(M_{a,i,k})$$

$$= \text{upper bound total entropy ,} \qquad (61)$$

$$\overline{ACC}_{a,k} = (1/J_{a,k}) \sum_{i=1}^{J_{a,k}} E(\| M_{a,i,k} - \mathring{M}_{a,i,k} \|^2)$$

$$= \text{averaged measure of performance accuracy ,} \qquad (62)$$

where E( ) indicates expectation, ENT( ) is entropy, $J_{a,k}$ is the number of nodes in $C^3_{a,k}$, and $\mathring{M}_{a,i,k}$ is some specified performance level for $M_{a,i,k}$.

Assuming the C^3 model presented here is a reasonable representation, a two person decision game can be played, where each player(a) corresponds to C^3_a, for a=1, 2. Modifying the notation in (57) in the obvious way, any strategy of player(a) corresponds uniquely to choice of $PRIM^{(1)}_{a,k}$ with the set of shared (internodal) functions between a=1 and a=2, $PRIM^{(2)}_k$, assumed fixed - or some variation of this theme, where for example only those internodal functions involving both of the C^3 systems are assumed given, with the remainder of the internodal functions adjoined to the appropriate $PRIM^{(1)}_{a,k}$ to form the strategy of player(a). Since Theorem 1 establishes a functional relation between the set of primitives and the outputs, in light of the above discussion, the game payoff or loss function can be considered to be any one, or an entire vector, of moe's or mop's, as given e.g., in (60)-(62). Then all usual properties of decision games may be obtained, such as game value, Bayesian, minimax, and least favorable strategies.

With suitable reduction of complexity, such a decision game can be of value in determining optimal design of C^3 systems and sensitivities of their performance as functions of the choices of set of primitive functions - subject to the usual constraints of limited supplies, geographical factors, weaponry available, sensor systems available to choose from, etc.

Although the basic C^3 model presented here has the formal structure of a probabilistic model, as mentioned before, by making appropriate substitutions (iterated disjuntion operations for integrals, conjunction operations for products, etc.), a possibilistic model can be established. In particular, this will occur when in the modeling of $(2)_{g,k}$, linguistic evidence is taken into account and combined with the usual probabilistic-based information in making decisions (such as in the tracking-correlation problem [9-11]) and in the modeling of $(6)_{h,g,k+1}$, "signals" received from responses sent are assumed to contain narrative information which can be treated through possibilistic descriptions. Since all of the remaining primitive functions involve variables connected with these functions and since all compound functions, including the output, are determined from the primitive set, it follows that the appropriate description must be in possibilistic form rather than strict probabilistic.

One area of potential interest in the further exploration of the model derived here is the determination of those conditions which will yield macroscopic or mesoscopic C^3 models as limiting cases. In particular the rather extensive statistical mechanics approach of Ingber [17] appears promising for such connections.

ACKNOWLEDGMENTS

This work was in part supported by the NOSC IR (Independent Research) Program Office, Code 013, and in part by the JDL (Joint Directors of Laboratories), C^3 Basic Research Group.

REFERENCES

1. Goodman,I.R.. "Combination of evidence in C^3 systems", *Proc. 8 MIT/ONR Wrkshp. C^3 Systems*, Dec. 1985, 161-166.

2. Athans,M., Ducot,E.R., Levis,A.H., Tenney,R.R.(eds.) *Proceedings of the 5th MIT/ONR Workshop on C^3 Systems*, Dec., 1982, LIDS, MIT, Cambridge,MASS.

3. Athans,M., Ducot,E.R., Levis,A.H., Tenney,R.R.(eds.) *Proceedings of the 6th MIT/ONR Workshop on C^3 Systems*, Dec., 1983, LIDS, MIT, Cambridge,MASS.

4. Athans,M., Levis,A.H. (eds.), *Proceedings of the 7th MIT/ONR Workshop on C^3 Systems*, Dec., 1984, LIDS, MIT, Cambridge, MASS.

5. Athans,M., Levis,A.H. (eds.), *Proceedings of the 8th MIT/ONR Workshop on C^3 Systems*, Dec., 1985, LIDS, MIT, Cambridge, MASS.

6. Levis,A.H., "Information processing and decision-making organizations: a mathematical description", *Proc. 6 MIT/ONR Wrkshp. C^3 Sys.*, Dec., 1983, 30-38.

7. Tomovic,M.M., Levis,A.H., "On the design of organization structures for command and control", *Proc. 7 MIT/ONR Wrkshp. C^3 Sys.*, Dec., 1984, 131-144.

8. Karam,J.G., Levis,A.H., "Effectiveness analysis of evolving systems", *Proc. 8 MIT/ONR Wrkshp. C^3 Sys.*, Dec., 1985, 53-64.

9. Goodman,I.R., "Use of literal information in multitarget data association", *Proc. 1985 Amer. Control Conf.*, 836-841.

10. Goodman,I.R., "Applications of a combined probabilistic and fuzzy set technique to the attribute problem in ocean surveillance", *Proc. 20 IEEE Conf. Decis. & Control*, Dec., 1981, 1409-1411.

11. Goodman,I.R., *PACT: An Approach to Combining Linguistic-Based and Probabilistic Information for Correlation and Tracking*, NOSC Tech. Doc. 878, March, 1986, Naval Ocean Systems Center, San Diego, CAL.

12. Goodman,I.R., Nguyen,H.T., *Uncertainty Models for Knowledge-Based Systems*, North-Holland Press, Amsterdam, 1985.

13. Goodman,I.R., "A general model for the multiple target correlation and tracking problem", *Proc. 18 IEEE Conf. Decis. & Control*, Dec., 1979, 383-388.

    A greatly expanded version can be found in: Goodman,I.R., *A General Model for the Contact Correlation Problem*, NRL Report 8417, July 27, 1983, Naval Research Lab., Wash., D.C.

14. Goodman, I.R., "A scoring procedure for the multiple target correlation and tracking problem", *Proc. 19 IEEE Conf. Decis. & Control*, Dec., 1980, 829-834.

15. Wohl,J.G., Entin,E.E., Kleinman,D.L., Pattipatti,K., "Human decision processes in military command and control", in *Advances in Man-Machine Systems Research, Vol. I*, JAI Press, Inc., 1984, 261-307.

16. Dieudonné, J., *Foundations of Modern Analysis*, Academic Press, New York, 1960, Chapt. X.

17. Ingber,L., "Nonlinear nonequilibrium statistical mechanics approach to C^3 systems", *Proc. 9 Wrkshp. C^3 Sys.* (to appear, 1986).

Stochastic Task Selection and Renewable Resource Allocation

Peter B. Luh, Xi-Yi Miao, Shi-Chung Chang
Dept. of Electrical and Systems Engineering
University of Connecticut
Storrs, CT 06268

David A. Castanon
Alphatech, Inc.
Burlinton,
MA 01808

Abstract

In this paper, we study an important class of resource allocation problems for the processing of dynamically arriving tasks with deterministic deadlines. This class of problems has numerous applications. For example, consider the operation of a Naval Battle Group with finite renewable resources (airctraft, warships, submarines, etc.). The battle group's mission is to process hostile threats, which can be of various types (air, surface, and underwater threats), have different strengths, and arrive stochastically. The battle group has only limited amount of time to process a threat before it causes substantial damages. Since available resources are finite, and a resource will be tied up for a certain amount of time once it is allocated, an effective task selection/resource allocation policy is highly desireable to maximize the survivability of the battle group. Another example involves the assignment of repairmen in a manufacturing plant. When a machine breaks down, repairmen have to be sent. With a limited number of repairmen, multiple machines, finite repairing times and random breakdowns, an dffective assignment policy is essential in maintaining the productivity of the plant under various contingencies. This class of problems conforms to neither the standard resource allocation model nor the standard optimal control model. A new problem formulation has to be developed and analyzed to provide a satisfactory answer to these problems.

In this paper, a new formulation for the task selection and renewable resource allocation problem is presented. As the tying up of resources in task processing implies time delay in resource flow, state augmentation is employed to convert the problem into a Markovian decision problem. The problem can then be treated, at least in principle, by using the stochastic dynamic programming (SDP) method. However, since the system dynamics involves the evolution of sets, the implementation of the dynamic programming equation is by no means straightforward. For a problem with infinite planning horizon, the optimal strategy is shown to be stationary under mild conditions. An SDP algorithm based on a successive approximation technique is developed to obtain the optimal stationary strategy. The implementation of the algorithm employs a special coding scheme to handle set variables, and utilizes a dominance property for computational efficiency. Effects of key system parameters on optimal decisions are investigated and analyzed through numerical examples. As the computational complexity of the algorithm is of exponential increase, practical applications of the algorithm is limited to problems of moderate size. Two heuristic rules are therefore investigated and compared to the optimal policy. The result of this study can serve as a starting point for further characterization of the optimal policy, for understanding and designing effective heuristic rules, and for developing (in conjunction with experimental studies) normative-descriptive models of human task selection and resource allocation.

1. Introduction

Motivation

In this paper, we study an important class of resource allocation problems for the processing of dynamically arriving tasks with deterministic deadlines. This class of problems has numerous applications. For example, consider the operation of a Naval Battle Group with finite renewable resources (airctraft, warships, submarines, etc.). The battle group's mission is to process hostile threats, which can be of various types (air, surface, and underwater threats), have different strengths, and arrive stochastically. The battle group has only limited amount of time to process a threat before it causes substantial damages. Since available resources are finite, and a resource will be tied up for a certain amount of time once it is allocated, an effective task selection/resource allocation policy is highly desireable to maximize the survivability of the battle group. Another example involves the assignment of repairmen in a manufacturing plant. When a machine breaks down, repairmen have to be sent. With a limited number of repairmen, multiple machines, finite repairing times and random breakdowns, an dffective assignment policy is essential in maintaining the productivity of the plant under various contingencies.

The class of resource allocation problems studied in this paper can be characterized as follows: Tasks arrive randomly in a scenario; associated with each task is a resource level required to process the task, a required time to process the task, a deadline by which the task should be processed, and a value function depeinding on the completion time and the number of resources allocated which specifies the reward of processing this task. In order to process these tasks, a decision maker has a finite number of renewable resources. The decision problem consists of dynamically allocating resources to tasks in order to maximize the total discounted reward.

The problems of task selection and resource allocation have long been studied from different perspectives and with different emphases. Queuing analysis and control have been used to study task priority assingment problems when values of servicing tasks do not depend on the service completion time, eg., [CAR66], [HAR75], [WHI77], [WAL78], [ROU80], [PAT81] and [WU85]. Although queuing formulations provide useful insights, they do not consider the effects of soft or hard deadlines and the selection of resource levels to process tasks. On the other hand, many static models and deterministic dynamic models have been suggested and solved for renewable resource allocation problems in operations research (e.g., [BAK74], [COF76], [SCH82], [DIA85]). However, since the stochastic aspect has not been included in these models, the resulting solution methodologies can not be easily extended to the problems which we are contemplating.

Motivated by decision making probems in scenarios such as Naval Battle Group/Battle force operations [KLE84], we develop in this paper mathematical formulations of the optimal deadline-driven task selection and resource allocation problem with multiple types of tasks and one class of renewable resource. There are two major purposes of our study. First, by setting up a simple but generic model that captures key ingredients of the problems, solution techniques and computational algorithms can be developed. Refinement of the model and solution methodology can be done later on. Second, the result will serve as a foundation in understanding the types of herurstic strategies which can be employed in human task selection and resource allocation. Though the present model is primitive, the results obtained do bring a fair amount of insights to key factors of the human decision making process.

The paper is organized as follows. In Section 2, a new formulation for the task selection and renewable resource allocation problem is presented. The model explicitly considers time available, time required, resource available, resource required, importance of a task, stochastic arrivals of multiple types of tasks, timeliness of processing and adequacy of resource allocation. As the tying up of resource in task processing implies delay in resource flow, state augmentation is employed to convert the problem into a Markovian decision problem. The optimal policy can thus be obtained, at least in principle, by applying the stochastic dynamic programming (SDP) method. However, since the system dynamics

* This work was supported in part by Alphatech Inc. under subcontract SC-000192-01 as part of the Office of Naval Research program on Distributed Tactical Decision Making, and by National Science Foundation under Grant ECS-8513163. The authors appreciate avluable inputs from Professor D. L. Kleinman and Mr. D. Serfaty.

involves the evolution of sets, the implementation of the dynamic programming equation is by no means straightforward. For a problem with infinite planning horizon, the optimal strategy is shown to be stationary under mild conditions. In section 3, the development of a SDP algorithm for the infinite horizon case is presented, and a successive approximation technique is used in obtaining numerical results. The implementation of the algorithm employs a special coding scheme to handle set variables, and utilizes a dominance property for computational efficiency. The computational complexity is also briefly analyzed. In section 4, effects of key system parameters on optimal decisions are investigated and analyzed through numerical examples. As the computational complexity of the SDP algorithm is of exponential increase, practical applications of the algorithm is limited to problems of moderate size. Two heuristic rules are therefore studied in Section 5. Their results are compared to optimal results of section 4.

2. Problem Formulation

Task States and Dynamics

Consider a discrete time task selection and resource allocation system with I types of tasks and one type of renewable resource. Tasks arrive stochastically and wait to be processed by a decision maker (DM). It is assumed that once a task appears the DM knows perfectly the type of the task, say type i ($1 \le i \le I$), together with the following attributes:

(1) the time period during which the task will stay in the system, i.e., initial time available, $T_{ao}(i)$;

(2) the amount of resource required to process the task, $\bar{r}_i$ (the strength of the task);

(3) the time required to process the task, $T_r(i)$;

(4) the reward for processing the task, g(i) (the importance of the task).

It is assumed that tasks of the same type are identical in attributes. At the appearance of a task, the time available is $T_{ao}(i)$, the initial time available. As time elapses, the time available decreases. A type i task with j units of time available is denoted as (i,j), where $1 \le i \le I$ and $1 \le j \le T_{ao}(i)$. New tasks appear stochastically. At any time, there may be more than one new arrivals. For simplicity we assume that all new tasks at a particular instant are of different types. The set of new tasks at time k is denoted by $n(k) = \{(i, T_{ao}(i))\}$, and the set of tasks selected to be processed is denoted by u(k). We define the "Active Task Set" S(k) as the set of all existing (with $1 \le j < T_{ao}(i)$) but yet unprocessed tasks at time k. The evolution of the active task set follows the "Task Flow Equation":

$$S(k+1) \equiv f(s(k), n(k), u(k)) \qquad (2.1)$$

$$= \{(i, j-1) \mid (i,j) \in S(k) \cup n(k),\ j > 1,\ (i,j) \notin u(k)\}.$$

At time k, the DM is supposed to select from S(k) a set of tasks u(k) to process. It is assumed that a task arrives at time k can not be processed immediately (i.e., at time k), as it takes time for the DM to make decisions. Note that the time available j of an unprocessed task decreases as time evolves until it reaches zero and leaves the system. Also, once a task is processed, it is removed from the active task set.

Resource States and Dynamics

The DM owns in total R units of renewable, discrete resources. When the DM allocates $r_{(i,j)}(k)$ units of resource to process task (i,j) at time k, this amount of resource will be tied up for $T_r(i)$ units of time before it can be utilized again. The flow of resource therefore involves dynamics with time delay. State augmentation is employed so that resource flow can be described by standard dynamic equations without delay. Let $x_m(k)$ denote the amount of tied up resource to be released at time k+m. For simplicity but without loss of generality, the types of tasks are arranged in the ascending order of $T_r(i)$, i.e.,

$$1 \le T_r(1) \le T_r(2) \le \cdots T_r(I) \equiv M, \qquad (2.2)$$

where M is largest time required in task processing. Then the "Resource Flow Equations" are:

$$x_{M-1}(k+1) = \sum_{(i,j) \in u(k),\ T_r(i) = M} r_{(i,j)}(k),$$

$$x_{M-2}(k+1) = x_{M-1}(k) + \sum_{(i,j) \in u(k),\ T_r(i) = M-1} r_{(i,j)}(k), \qquad (2.3)$$

$$\vdots$$

$$x_1(k+1) = x_2(k) + \sum_{(i,j) \in u(k),\ T_r(i) = 2} r_{(i,j)}(k).$$

The vector $X(k) \equiv (x_1(k), x_2(k), \ldots, x_{M-1}(k))^T$ is called the "Resource Usage Vector", where superscript T denotes transpose. Since the total amount of resource owned by the DM is R, the following constraint holds for every k:

$$0 \le \sum_{m=1}^{M-1} x_m(k) + \sum_{(i,j) \in u(k)} r_{(i,j)}(k) \le R, \qquad (2.4)$$

i.e., the total resource usage (tied up units plus newly allocated units) can not be greater than R. The resource usage vector X(k) and the active task set S(k) together then form the state of the system.

Task Arrival Statistics and Reward Structure

Task arrivals are likely to depend on system states. For example, new threats often appear in specific patterns based on current threats and enemy's estimation of our resource utilization situation. We therefore model the probability of task arrivals as $p(n(k) \mid S(k), X(k))$, i.e., a Markov process.

The DM maximizes his rewards in processing tasks. Two aspects are emphasized in the reward structure: timeliness of processing and adequacy of resource allocation. Timely processing means that when task (i,j) is selected, its time available is no less than its time required, i.e., $j \ge T_r(i)$. Adequate processing means that the amount of resource allocated to task (i,j) is no less than the amount of resource required $\bar{r}_i$ (these definitions can be modified without much difficulty). Untimely and/or inadequate processing result in diminishing rewards. Let g(i) be the reward for timely and adequate processing of a type I task. The reward for processing task (i,j) is then described by

$$g(i) g_1(j - T_r(i)) g_2\left[\frac{r_{(i,j)}}{\bar{r}_i}\right] \qquad (2.5)$$

where $0 \le g_1(\cdot) \le 1$ and $0 \le g_2(\cdot) \le 1$ are functions representing discounting factors for untimely and inadequate processing, respectively. They can be defined at users' disposal, and in general should satisfy the following conditions:

$$g_1(j - T_r(i)) = \begin{cases} 1 & \text{if } j - T_r(i) \ge 0, \\ \le 1 & \text{if } j - T_r(i) < 0, \end{cases} \qquad (2.6)$$

$$g_2\left[\frac{r_{(i,j)}}{\bar{r}_i}\right] = \begin{cases} 1 & \text{if } \dfrac{r_{(i,j)}}{\bar{r}_i} \ge 1, \\ <1 & \text{if } \dfrac{r_{(i,j)}}{\bar{r}_i} < 1. \end{cases} \qquad (2.7)$$

Since we assume that the DM has perfect information about existing tasks, and there is no uncertainty in task processing once resource is allocated, rewards can be counted at the time when decision is made. This thus justifies the deletion of u(k) from the active task set as described by (2.1).

The Markovian Decision Problem and Problems with Infinite Planning Horizon

The formulation of a K-stage problem can now be summarized as

$$\max_{U,r} E\left[\sum_{k=0}^{K-1} \sum_{(i,j)\in u(k)} g(i)g_1(j-T_r(i))g_2\left[\frac{r_{(i,j)}}{r(i)}\right]\right] \tag{2.8}$$

subject to (1) the task flow equation (2.1)

(2) the resource flow equations (2.3),

(3) the resource constraint (2.4),

where $U\equiv(u(0),u(1),...,u(K-1))$ and $r\equiv(r(0),r(1),...,r(K-1))$. It is not difficult to see that this is a Markovian decision problem with stagewise additive objective function. The problem can therefore be solved, at least in principle, by using the stochastic dynamic programming (SDP) method.

Now consider an infinite horizon problem with the following discounted objective function:

$$\lim_{K\to\infty} E\left[\sum_{k=0}^{K-1} \alpha^k \sum_{(i,j)\in u(k)} g(i)g_1(j-T_r(i))g_2\left[\frac{r_{(i,j)}}{\bar{r}_i}\right]\right] \tag{2.9}$$

where $0 < \alpha < 1$. If the task arrival statistics $P(\bullet|\ X(k), S(k))$ is a countable measure and g(i) is bounded above, then an optimal stationary policy exists following Chapter 6 of [BER77]. We shall next describe the development of an algorithm in obtaining the optimal stationary policy for such a problem.

3. Solution Methodology

Stationary Dynamic Programming Equation and the Handling of Set Variables

Consider an infinite horizon, discounted stochastic dynamic programming problem with the objection function described by (2.9). Let us define

$$J^*(S,X) = \max_{(u,r)} E\left[\sum_{k=0}^{\infty} \alpha^k \sum_{(i,j)\in u(k)} g(i)g_1(j-T_r(i))g_2\left[\frac{r_{(i,j)}}{\bar{r}_i}\right]\right] \tag{3.1}$$

as the optimal value function for the given initial state (S,X). By utilizing the stationarity of the the optimal policy, the dynamic programming equation can be written as

$$\begin{aligned} J^*(S,X) &= \max_{(u,r)} E[G_{(u,r)} + \alpha J^*(S^1,X^1)] \qquad (3.2)\\ &= \max_{(u,r)}\left[G_{(u,r)} + \alpha P(S^1,X^1|\ S,X,u,r)J^*(S^1,X^1)\right] \end{aligned}$$

subject to (2.1), (2,3) and (2.4),

where (u,r) is an admissible stationary policy, $G_{(u,r)} \equiv \sum_{(i,j)\in u(k)} g(i)g_1(j-T_r(i))g_2\left[\frac{r_{(i,j)}}{r(i)}\right]$ is the single stage reward, and $P(S^1,X^1|\ S,X,u,r)$ is the stationary state transition probability matrix derived from (u,r) and task arrival statistics $p(.|\ X,S)$.

It is apparent that the problem involves both vector and set variables, which include S(k) and u(k). The latter is unusual in SDP. A special coding scheme is therefore developed to handle set variables. The idea is explained by using a simple example. Suppose that there are only two types of tasks, i.e., I=2 with $T_{ao}(1)=4$, $T_{ao}(2)=3$. Clearly, the largest possible active task set is $\bar{S}\equiv$ {(1,1), (1,2), (1,3), (2,1), (2,2)}, and any active task set S(k) is a subset of $\bar{S}$. The total number of active task sets is 2^5, and any active task set can be represented by a binary number

$$B(k) = b_1b_2b_3b_4b_5 \tag{3.3}$$

where b_n is a 0-1 indication variable corresponding to the nth element of $\bar{S}$. In other words, $b_n=1$ if the nth task of $\bar{S}(k)$ is in S(k). We thus have a one to one mapping between all possible active task sets and binary numbers ranging from 0 to 11111. For instance, b=01101 corresponds to S(k)={(1,2), (1,3), (2,2)}, and 00000 corresponds to S(k)= ϕ, etc. These binary numbers can then be mapped to integer numbers. Both set variables S(k) and u(k) are handled by using this scheme.

For a finite dimensional problem, i.e., both X and S take finite values, (3.2) can be solved by using the successive approximation technique described in [BER77] (p.246). It begins with an initial guess, $J^0(S,X)$, of the optimal value function $J^*(S,X)$. A new approximation of $J^*(S,X)$ is obtained successively by

$$J^{t+1}(S,X) = \max_{(u,r)}\left[G_{(u,r)} + \alpha P(S^1,X^1|\ S,X,u,r)J^t(S^1,X^1)\right], \tag{3.4}$$

where t=0, 1, 2, ... The procedure continues until it converges.

The algorithm is implemented on a simplified version of the problem, in which the reward structure is of the following form:

$$g_1(j-T_r(i)) = \begin{cases}1 & \text{if } j-T_r(i)\geq 0,\\ 0 & \text{if } j-T_r(i)<0,\end{cases} \tag{3.5}$$

and

$$g_2\left[\frac{r_{(i,j)}}{\bar{r}_i}\right] = \begin{cases} 1 \text{ if } \dfrac{r_{(i,j)}}{\bar{r}_i}\geq 1,\\ 0 \text{ if } \dfrac{r_{(i,j)}}{\bar{r}_i}<1,\end{cases} \tag{3.6}$$

i.e., no partial reward for an untimely and/or inadequate processing. The statistics of new task arrivals is assumed to be independent, identically distributed for each type of tasks. These simplifying assumptions can be removed without much difficulty.

Computational Complexity and a Dominance Property

Since at most one new task per type can appear at a time, and a type i task stays in the system for $T_{ao}(i)$ units of time if left unprocessed, the largest number of existing tasks is $\sum_{i=1}^{I} T_{ao}(i)$. From the reward structure (3.5), a task with time available less than time required deserves no attention, and the "effective selection time" available for task (i,j) is $j-T_r(i)+1$. We therefore define the "Reduced Active Task Set" $S_r(k)$ to drop all those worthless tasks and to use the effective selection time to replace the time avilable.

$$S_r(k)\equiv\{(i,j-T_r(i)+1)|\ (i,j)\in S(k),\ T_r(i)\leq j<T_{ao}(i)\}. \tag{3.7}$$

In this notation, (i,j') represents a type i task with j' units of effective selection time. This representation will be used for the remaining part of the paper. Note that the set $S_r(k)$ has at most $L = \sum_{i=1}^{I} [T_{ao}(i)-T_r(i)]$ elements in it. The number of possible S_r is therefore 2^L. It is also clear that the resource states are of similar combinatorial nature. Without considering any other properties of the solution, the decision space will be roughly the same as the state space. Consequently, the computational complexity of the dynamic programming algorithm is of exponential increase.

To improve the computational efficiency in solving (3.4), a dominance property of the optimal strategy is used to reduce the admissible decision space: among the same type of tasks, the task with the shortest time available has the highest priority. The reason is that if we process a task of the same type but with longer time available, the reward and the resource tied up situation will be identical to the case if we process the one with the shortest time available. However, we are more likely to lose the task with the shortest time available. Starting with the most urgent task, we preserve the opportunity for processing other tasks in the future. Using this dominance property and also considering the resource constraint (eq.

(2.4)), the number of admissible decisions can be reduced, though it is still of exponential increase. Further refinement in modeling and solution methodology are needed to make the algorithm more efficient.

4. Numerical Results

Beyond the problem formulation and solution methodology, we are particularly interested in effects of following system parameters on optimal decisions:

(1) look-ahead factor α,

(2) total amount of resource R,

(3) reward g(i),

(4) task arrival probability p(i), and

(5) time required $T_r(i)$ and initial time available$T_{ao}(i)$.

The study is done through numerical examples under the simplified formulation of (3.5) and (3.6) with two types of tasks and specific sets of system parameters.

Example 1: Look-Ahead Factor α

The look-ahead factor α reflects how far the DM looks into the future. For a given α, the effective look ahead time is $1/(1-\alpha)$. For example, α = 0.9 implies the effective look ahead time of 10 stages. Here we examine the effects of look-ahead factor by comparing decisions with two different values of α . Part of parameters used are given in Table 4.1.

Table 4.1
Parameters for Example 1

	type 1	type 2
T_{ao}	7	5
T_r	4	2
$\bar{r}$	2	1
g	20	10

In addition, task arrival statistics are independent, identically distributed with p(1)=0.35, p(2)=0.35, and the total amount of available resource R is 3. Samples of optimal stationary decisions for α =0.9 and α =0.1 are given in Tables 4.2 and 4.3, respectively.

Table 4.2
Samples of Optimal Policy with $\alpha = 0.9$

	Reduced Active Task Set	Tied Up Resources			Decisions
	s_r	x_3	x_2	x_1	u
1	(1,1) (2,1)	0	0	0	(1,1) (2,1)
2	(1,3) (2,1)	0	0	0	(1,3) (2,1)
3	(1,3) (1,1) (2,1)	0	0	0	(1,1) (2,1)
4	(1,3) (2,2) (2,1)	0	0	0	(2,2) (2,1)
5	(1,3) (1,1) (2,3) (2,2) (2,1)	0	0	0	(2,3) (2,2) (2,1)
6	(1,1) (2,3) (2,2) (2,1)	0	0	0	(2,3) (2,2) (2,1)
7	(1,1) (2,3) (2,2)	0	0	1	(1,1)
8	(1,2) (2,3) (2,1)	0	0	1	(2,1)

Table 4.3
Samples of Optimal policy with $\alpha = 0.1$

	Reduced Active Task Set	Tied up Resources			Decisions
	S_r	x_3	x_2	x_1	u
1	(1,1) (2,1)	0	0	0	(1,1) (2,1)
2	(1,3) (2,1)	0	0	0	(1,3) (2,1)
3	(1,3) (1,1) (2,1)	0	0	0	(1,1) (2,1)
4	(1,3) (2,2) (2,1)	0	0	0	(1,3) (2,1)
5	(1,3) (1,1) (2,3) (2,2) (2,1)	0	0	0	(2,3) (2,2) (2,1)
6	(1,1) (2,3) (2,2) (2,1)	0	0	0	(2,3) (2,2) (2,1)
7	(1,1) (2,3) (2,2)	0	0	1	(1,1)
8	(1,2) (2,2) (2,1)	0	0	1	(1,2)

Decisions for cases 1,2,3, and 7 of these two tables are identical and intuitively clear. Urgent tasks are processed with all available resource. Decisions for cases 5 and 6 of these two tables, though identical, are not as easy to comprehend. Optimal policies select all type 2 tasks at the cost of losing the task (1,1). Such decisions are results of the look-ahead consideration. Suppose in case 5 tasks (1,1) and (2,1) are selected instead. Although task (1,1) will be processed in time, however, tasks (2,2) and (1,3) will surely be lost. According to the optimal decision, only task (1,1) is surely lost. In case 4, the resource constraint allows only the simultaneous processing of a type 1 and a type 2 tasks, or two type 2 tasks. With α = 0.1 as in Table 4.3, the DM puts more emphases on immediate rewards. He selects tasks (1,3) and (2,1) to get thirty units of immediate rewards, which is higher than the choice of (2,1) and (2,2) as in Table 4.2. However, when the DM puts more weight on future rewards than the case with α = 0.1, the selection of (2,2) and (2,1) is certainly superior. By doing so, the DM preserves the possibility of processing task (1,3) after finishing tasks (2,2) and (2,1). In case 8, one unit of resource is reserved in Table 4.2, as opposed to the full allocation of Table 4.3. With one unit of resource reserved, task (1,2) can be processed at the next stage as one unit of resource is going to be released at that time; and at the following stage, task (2,3) can be processed by using the resource currently allocated to task (2,2). Consequently, chances for processing (1,2) and (2,3) are preserved. On the other hand if (1,2) is processed as in Table 4.3, task (2,1) will surely be lost.

For all following examples, we keep α =0.9 and examine effects of other system parameters.

Example 2: Total Amount of Resource R

In this example, the total amount of resource is reduced from three to two, with all other parameters remained the same as in Example 1. Samples of optimal decisions are presented in Table 4.4.

Table 4.4
Samples of Optimal Policy with Scarce Resource

	Reduced Active Task Set	Tied up Resources			Decisions
	S_r	x_3	x_2	x_1	u
1	(1,1) (2,1)	0	0	0	(1,1)
2	(1,3) (2,1)	'0	0	0	(2,1)
3	(1,3) (1,1) (2,1)	0	0	0	(2,1)
4	(1,3) (2,2) (2,1)	0	0	0	(2,2) (2,1)
5	(1,3) (1,1) (2,3) (2,2) (2,1)	0	0	0	(2,2) (2,1)
6	(1,1) (2,3) (2,2) (2,1)	0	0	0	(2,2) (2,1)
7	(1,1) (2,3) (2,2)	0	0	1	(2,2)
8	(1,2) (2,2) (2,1)	0	0	1	(2,1)

We observe that the optimal policy tends to select type 2 tasks, which require less resource and less processing time. The reason is that to process a type 1 task, the DM will stay idle for the next three stages. When there are type 2 tasks in $S_r(k)$ and more are going to come (p(2)=0.35), it is worthwhile to process type 2 tasks.

Example 3: Reward g(i)

Consider the case where g(2) is changed from 10 to 40, with all other parameters remained as in Example 2. It is intuitively clear that if one type of tasks becomes more valuable, that type of tasks will be selected. This can be seen from Table 4.5:

Table 4.5
Samples of Optimal Policy with Valuable Tasks

	Reduced Active Task Set	Tied Up Resources			Decisions
	S_r	x_3	x_2	x_1	u
1	(1,1) (2,1)	0	0	0	(2,1)
2	(1,3) (2,1)	0	0	0	(2,1)
3	(1,3) (1,1) (2,1)	0	0	0	(2,1)
4	(1,3) (2,2) (2,1)	0	0	0	(2,2) (2,1)
5	(1,3) (1,1) (2,3) (2,2) (2,1)	0	0	0	(2,2) (2,1)
6	(1,1) (2,3) (2,2) (2,1)	0	0	0	(2,2) (2,1)
7	(1,1) (2,3) (2,2)	0	0	1	(2,2)
8	(1,2) (2,2) (2,1)	0	0	1	(2,1)

Note that in this table, no type 1 task is selected. The reason is that if a type 1 task is selected, the DM will lose the capability of processing the much valuable type 2 tasks for four stages. In fact, our results show that the optimal policy selects no type 1 task under any circumstance.

Example 4: Task Arrival Probability p(i)

Consider a problem the same as Example 1 except that the task arrival probabilities are changed from p(1)=p(2)=0.35 to p(1)=0.05, p(2)=0.35, with R=3. Samples of results are presented in Table 4.6.

Table 4.6
Samples of Optimal Policy with Different Arrival Rates

	Reduced Active Task Set	Tied Up Resources			Decisions
	s_r	x_3	x_2	x_1	u
1	(1,1) (2,1)	0	0	0	(1,1) (2,1)
2	(1,3) (2,1)	0	0	0	(1,3) (2,1)
3	(1,3) (1,1) (2,1)	0	0	0	(1,1) (2,1)
4	(1,3) (2,2) (2,1)	0	0	0	(2,2) (2,1)
5	(1,3) (1,1) (2,3) (2,2) (2,1)	0	0	0	(2,3) (2,2) (2,1)
6	(1,1) (2,3) (2,2) (2,1)	0	0	0	(1,1) (2,1)
7	(1,1) (2,3) (2,2)	0	0	1	(1,1)
8	(1,2) (2,3) (2,1)	0	0	1	(2,1)

Comparing Tables 4.6 and 4.2, we see that in Table 4.2 the optimal choices for case 6 are (2,1) (2,2) and (2,3), whereas in Table 4.6 the optimal choices are (1,1) and (2,1) instead. This indicates that with type 1 arrival probability reduced, the relative importance of type 1 tasks increases. It might become worthable to process the task (1,1) at the cost of lossing a more common type 2 task. In general, however, the relationship between importance and arrival probability is very complicated. It depends also on the total amount of resource R. For example, with the same arrival probabilities but having R=2, the optimal policy suggests that the relative importance of type 2 tasks increases.

Example 5: Time Required and Initial Time Available

The length of effective selection window for a type i task is $T_{ao}(i) - T_r(i)$. Consider a problem same as Example 1 except that $T_{ao}(1) = 5$ (the length of effective selection window is reduced from 3 to 1). Samples of results are presented in Table 4.7.

Table 4.7
Samples of Optimal Policy with Short Initial Time Available

	Reduced Active Task Set	Tied Up Resources			Decisions
	s_r	x_3	x_2	x_1	u
1	(1,1) (2,1)	0	0	0	(1,1) (2,1)
2	(1,1) (2,1)	0	0	0	(1,1) (2,1)
3	(1,1) (2,1) (2,1)	0	0	0	(1,1) (2,1)
4	(1,1) (2,2) (2,1)	0	0	0	(1,1) (2,1)
5	(1,1) (2,3) (2,2) (2,1)	0	0	0	(2,3) (2,2) (2,1)
6	(1,1) (2,3) (2,2) (2,1)	0	0	0	(1,1) (2,1)
7	(1,1) (2,3) (2,2)	0	0	1	(1,1)
8	(1,1) (2,3) (2,1)	0	0	1	(1,1)

We see that more type 1 tasks are selected as compared to Table 4.2, as the only chance to process a type 1 task in $S_r(k)$ is to process it now.

From above examples, we see that the optimal policy is quite complicated, and depends on system state and parameters in a very perplexing way.

5. Testing of Two Heuristic Rules

In previous sections, a stochastic dynamic programming algorithm is developed to obtain optimal decisions. Because of computational complexity, however, the application of this algorithm is limited to problems of moderate size. For large problems, the computational requirements in finding optimal solutions are formidable. The solutions, once obtained, might also be too complicated for efficient implementation. As human decision making is generally believed to be heuristic, and there are many heuristic rules in practical applications, we shall therefore study two heuristic rules through numerical examples.

The Myopic Policy

The first heuristic rule considered is the "myopic policy", in which decisions are made to achieve the maximum present rewards without considering future effects. Note that though the myopic policy might be adopted for the sake of simplicity, the DM himself could still be characterized by a nonzero look-ahead factor. The myopic policy can be regarded as a solution of the following problem:

$$\max_{(u,r)} \sum_{(i,j)\in u(k)} g(i) g_1(j - T_r(i)) g_2\left[\frac{r(i,j)}{r(i)}\right] \tag{5.1}$$

subject to (2,1), (2,3) and (2,4).

In solving (5.1), the dynamic and stochastic aspects of the problem are not involved. Therefore the decision at time k can be obtained by simply selecting tasks from $S_r(k)$ according to their reward values subject to the resource constraint. Consequently, there is no computational complexity issue, and implementation is also straightforward. For comparison purpose, samples of optimal decisions for Example 1 with corresponding expected rewards obtained by using the SDP algorithm for $\alpha = 0.9$ are presented in Table 5.1. Corresponding myopic decisions and expected rewards (assuming also $\alpha = 0.9$ in calculating expected rewards) are presented in Table 5.2.

Table 5.1
Samples of Optimal Look-Ahead Policy ($\alpha = 0.9$)

	Reduced Active Task Set	Tied Up Resources			Decisions	Rewards
	s_r	x_3	x_2	x_1	u	G
1	(1,1) (2,1)	0	0	0	(1,1) (2,1)	90.5775
2	(1,3) (2,1)	0	0	0	(1,3) (2,1)	90.5775
3	(1,3) (1,1) (2,1)	0	0	0	(1,1) (2,1)	90.5775
4	(1,3) (2,2) (2,1)	0	0	0	(2,2) (2,1)	91.4431
5	(1,3) (1,1) (2,3) (2,2) (2,1)	0	0	0	(2,3) (2,2) (2,1)	100.6092
6	(1,1) (2,3) (2,2) (2,1)	0	0	0	(2,3) (2,2) (2,1)	96.6855
7	(1,1) (2,3) (2,2)	0	0	1	(1,1)	88.5613
8	(1,2) (2,3) (2,1)	0	0	1	(2,1)	90.0229
9	(1,3) (2,1)	0	0	1	(2,1)	85.4686

Table 5.2
Samples of Myopic Policy

	Reduced Active Task Set	Tied up Resources			Decision	Reward
	S_r	x_3	x_2	x_1	u	G
1	(1,1) (2,1)	0	0	0	(1,1) (2,1)	90.5403
2	(1,3) (2,1)	0	0	0	(1,3) (2,1)	90.5403
3	(1,3) (1,1) (2,1)	0	0	0	(1,1) (2,1)	90.5403
4	(1,3) (2,2) (2,1)	0	0	0	(1,3) (2,1)	90.5403
5	(1,3) (1,1) (2,3) (2,2) (2,1)	0	0	0	(2,3) (2,2) (2,1)	100.5757
6	(1,1) (2,3) (2,2) (2,1)	0	0	0	(2,3) (2,2) (2,1)	96.6499
7	(1,1) (2,3) (2,2)	0	0	1	(2,2) (2,1)	87.3330
8	(1,2) (2,3) (2,1)	0	0	1	(2,3) (2,1)	87.3330
9	(1,3) (2,1)	0	0	1	(1,3)	81.2740

Many of the myopic decisions can be explained as in Example 1, and the myopic policy Note that for each case, the percentage difference in rewards for the two policies is not big. The myopic policy may lose more existing tasks in the future than the optimal one for being shortsighted. However, without processing those tasks, the resource will be left available. As probabilities of new task arrivals are quite high (p(1)=p(2)=0.35), the available resources can then be used to process the tasks in a myopic fahion. Discount factor tends to further reduces the difference. Consequently, the difference in reward between the two policies would not too large.

The μc Rule

The second rule is the so called μc rule in queueing theory [COX61]. The priority index of task {(i,j)} is computed as $\frac{g(i)}{r(i)} \frac{1}{T_r(i)}$, where $\frac{g(i)}{r(i)}$ is the reward per unit resource for type i task. For tasks of the same type, the one with the shortest time available has the highest priority. Resource is allocated in the descending order of priority index, until available resource is exhausted or the lowest priority task has been considered. The computational requirements are low. Sample results are presented in Table 5.3.

Table 5.3
Samples of μc Rule

	Reduced Active Task Set	Tied up Resources			Decision	Reward
	S_r	x_3	x_2	x_1	u	G
1	(1,1) (2,1)	0	0	0	(1,1) (2,1)	89.2510
2	(1,3) (2,1)	0	0	0	(1,3) (2,1)	89.2510
3	(1,3) (1,1) (2,1)	0	0	0	(1,1) (2,1)	89.2510
4	(1,3) (2,2) (2,1)	0	0	0	(2,2) (2,1)	89.6104
5	(1,3) (1,1) (2,3) (2,2) (2,1)	0	0	0	(2,3) (2,2) (2,1)	98.8982
6	(1,1) (2,3) (2,2) (2,1)	0	0	0	(2,3) (2,2) (2,1)	95.2213
7	(1,1) (2,3) (2,2)	0	0	1	(2,3) (2,2)	85.9389
8	(1,2) (2,3) (2,1)	0	0	1	(2,3) (2,1)	85.9389
9	(1,3) (2,1)	0	0	1	(2,1)	82.5660

This rule considers relative importance of tasks in a heuristic way. If tasks of different types have about the same reward but with quite different time required and/or resource required, this rule could yield better results than the myopic policy. However since the priority is determined in an ad hoc way, and the resource tied up situation and task arrival probabilities are not considered, the μc rule is not always better than the myopic scheme. In fact, for our example it is inferior to the myopic rule for most of the cases shown.

6. Summary

A major difficulty in studying stochastic task selection and renewable resource allocation is to come up with the right problem formulation which includes key ingredients and also lends itself to systematic analysis. The model presented here explicitly considers time available, time required, resource available, resource required, stochastic arrivals of multiple types of tasks, importance of a task, timeliness of processing and adequacy of resource allocation. After state augmentation, the problem becomes a Markovian decision problem, and can be solved by using the stochastic dynamic programming method. For a problem with infinite horizon, the optimal policy is stationary under mild conditions. A stochastic dynamic programming algorithm is developed to find the optimal stationary policy based on a successive approximation techniques. In implementating the algorithm, a specific coding scheme and a dominance property are used. Numerical results are obtained and effects of key system parameters are examined and analyzed. Two heuristic rules are investigated and compared with the optilmal policy. The results of this study will serve as a starting point in further characterization of the optimal policy, in understanding and designing effective heuristic rules, and in developing (in conjunction with experimental studies) normative-descriptive models of human task selection and resource allocation.

REFERENCES

[BAK74] K. R. Baker, *Introduction to Sequencing and Scheduling,* Wiley, New York, 1974.

[BER76] D.P. Bertsekas, *Dynamic Programming and Stochastic Control,* Academic Press, Inc., New York, 1976.

[CAR66] J. R. Carbonell, "A Queueing Model of Many Instrument Visual Sampling," *IEEE Trans. Hum. Factors Electron.,* Vol. HFE-7, pp. 154-164, Dec. 1966.

[COF76] E. G. Coffman, *Computer and Job Shop Scheduling,* Wiley, 1976.

[COX61] D. R. Cox and W. L. Smith, *Queues,* Methuen, London, 1961.

[DIA85] M.D. Diamond, and O.M. Carducci, "Decision Processes for Large Scale Resource Allocation Problems," *Proceedings of the 8th MIT/ONR Workshop on* c^3 *Systems,* June 1985, Cambridge, MA, pp. 153-160.

[HAR75] J. M. Harrison, "Dynamic Scheduling of a Multiclass Queue: Discount Optimality," *Operations Research,* Vol. 23, No.2, pp. 70-82, March/April, 1975.

[KLE84] D. K. Kleinman, D. Serfaty, P. B. Luh, "A Research Paradigm for Multi-Human Decision Making," *Proceedings of the 1984 American Control Conference,* San Diego, CA, June 1984, pp. 6-11.

[PAT81] K. R. Pattipati and D. L. Kleinman, "Priority Assignment Using Dynamic Programming for a Class of Queueing Systems, " *IEEE Transactions on Automatic Control,* Vol. AC-26, No.5, Oct., 1981, pp 1095-1106.

[ROU80] W. B. Rouse, *System Engineering Models of Human-Machine Interaction,* Series Vol. 6, North Holland, 1980.

[SCH82] M.K. Schaefer, "Optimal Allocation of Recoverable Items," *Decision Science,* pp. 147-155, 1982.

[WAL78] R. S. Walden, "A Queueing Model of Pilot Decision-making in a Multitask Flight Management Situation," *IEEE Trans. on System, Man and Cybernetics,* Vol. SMC-8, pp. 867-875, Dec. 1978.

[WHI77] J. A. White, J. W. Schmidt and G. K. Bennett, *Analysis of Queueing Systems,* New York, Academic Press, Inc. 1977

[WU85] Z. J. Wu, P. B. Luh, S. C. Chang, D. A. Castanon, "Optimal Task Allocation for Team of Two Decision Makers with Three Classes of Impatient Tasks," *Proceedings of the 8th MIT/ONR Workshop on C*^{3} *Systems,* Cambridge, MA, June 1985, pp. 95-100.

Multiple Target Estimation using Multiple Bearing-only Sensors

Dr. Patrick R. Williams

Hughes Aircraft Co.
Ground Systems Group
P.O. Box 3310
Fullerton, CA 92634

Summary

This paper describes a new method for determining the position of multiple targets in a region using three bearing-only sensors (sensors which measure only the bearing to the target). The method is a global maximum likelihood estimation procedure; the likelihood function is maximized through the use of discrete mathematical optimization (integer programming).

The method employed for solving the integer programming problem is an implicit enumeration algorithm which is easily reformulated for implementation in a parallel computer architecture. An initial investigation has shown that the proposed algorithm exhibits a high degree of parallelism.

The proposed method of solution is applicable to most netted passive surveillance systems such as jammed radar, sonar, ESM, and IR. No assumption of sensor type is made; the only assumption is that bearings are the sole data collected.

Background

This problem is related to the multitarget tracking problem which has been of interest in recent years. The maximum likelihood procedure presented herein is similar to a multitarget tracking procedure presented by Morefield [4]. However in this paper we will be concerned with bearing-only sensors and are only estimating positions, not tracking.

Multiple sensor bearing-only tracking is not new. Algorithms which locate and track targets using networks of sensors such as sonar and ESM have already been developed (see Morefield [3] , [5] and Lafrance and Rivers [2]). However, those techniques assume that additional information is available (e.g. frequency of the emissions) to the algorithm. This additional information is used for the correlation of data received by the different sensors as a discriminant of distinct targets. The disadvantages of these techniques are higher loads on the communications links (the additional information must be passed over the link) and more complicated signal processing (the sensor must accurately measure the additional information). The method presented herein differs from other methods in that it relies only on bearing information for target location. The algorithm is described below.

Algorithm Description

Consider a planar region where all targets can be described by their cartesian (x,y) position. Assume that all targets of interest can be detected by three non-collocated sensors (with known sensor positions). Let us assume that these sensors measure only an azimuth angle, θ, to the targets. We assume that n targets lie in the surveillance region (n unknown). The statistical errors associated with each data point, θ, are measurement errors and are assumed to be Gaussian with known variance. Further, data may be missing due to an imperfect detection process. We shall call a target estimable if all three sensors detect that target (no missing data for that target). We shall attempt to estimate positions only for estimable targets. Thus any data generated by inestimable targets can be considered extra or noise. By considering only estimable targets we have "eliminated" missing data but have added spurious measurements.

The problem is to decide how the measurement data is to be clustered or partitioned into sets corresponding to estimable targets and sets corresponding to noise.

The notation used for this discussion is as follows:

σ_i^2 = variance of measurements associated with sensor i

Φ_i = field of view of sensor i, $0 < \Phi_i \leq 2\pi$

m_i = number of targets (data points) detected by sensor i

$\{\theta_{ij}\}$ = the data collected by sensor i (j = 1,2, · · ·, m_i)

(u_i, v_i) = the x-y coordinates of sensor i.

Define $I = \{1, 2, \cdots, m_1\}$, $J = \{1, 2, \cdots, m_2\}$, and $K = \{1, 2, \cdots, m_3\}$. Since the arrangement of the data is a major obstacle for this estimation problem, we now provide a mathematical description of that arrangement. A partition of the data set is defined as the pair (Γ, Δ), where $I \times J \times K \supset \Gamma$ such that $\Gamma = (\Gamma_1, \Gamma_2, \Gamma_3)$ and $I \supset \Gamma_1$, $J \supset \Gamma_2$, and $K \supset \Gamma_3$. Further, $\Delta = (\Delta_1, \Delta_2, \Delta_3)$, $I \times J \times K \supset \Delta$, where $\Delta_1 = I \setminus \Gamma_1$, $\Delta_2 = J \setminus \Gamma_2$, $\Delta_3 = K \setminus \Gamma_3$.

The set, Γ, consists of ordered 3-tuples, $\gamma = (i, j, k)$, such that the measurement data triple, $(\theta_{i\cdot}, \theta_{j\cdot}, \theta_{k\cdot})$, corresponds to a single object. As each datum, $\theta(\cdot, \cdot)$ can be a measurement from exactly one object the set Γ is constructed so that if $\gamma, \gamma^* \varepsilon \Gamma$ then each entry of the 3-tuples γ, γ^* have no common elements. Thus the set Γ corresponds to the estimable targets and the set Δ corresponds to the spurious measurements. In general this partition is unknown and must be estimated along with the estimable target positions.

Since the maximum likelihood approach will be used for estimation, the density functions for the data are given below. Define θ_γ as $\theta_\gamma = (\theta_{i\cdot}, \theta_{j\cdot}, \theta_{k\cdot})^t$, $\gamma = (i, j, k) \varepsilon \Gamma$. We assume that θ_γ has the Gaussian distribution or for $\gamma \varepsilon \Gamma$,

$$f_\gamma = (1/2\pi)^{3/2} \, |\Sigma|^{-1/2} \exp\{ -\tfrac{1}{2}\theta_\gamma^t(x,y) \, \Sigma^{-1} \, \theta_\gamma(x,y)\}$$

where $\theta_\gamma = (\theta_i, \theta_j, \theta_k)^t$, $\theta_i(x, y) = \arctan (x - u_i)/(y - v_i)$, and $\Sigma = \mathrm{diag}(\sigma_1^2, \sigma_2^2, \sigma_3^2)$.

For the spurious measurements we shall assume the uniform distribution or for a measurement $\theta_{i,j}$ corresponding to a spurious measurement the densiity is given by:

$$f_{\theta_{i,j}} = 1/\Phi_i \qquad \text{whenever } j \,\varepsilon\, \Delta_i$$

Then the likelihood function for the surveillance region can be written as:

$$L = \prod_{\gamma \varepsilon \Gamma} (1/2\pi)^{3/2} \, |\Sigma|^{-1/2} \exp\{ -\tfrac{1}{2}\theta_\gamma^t(x,y) \, \Sigma^{-1} \, \theta_\gamma(x,y) \} \cdot \prod_{i \varepsilon \Delta_1} 1/\Phi_1 \cdot \prod_{j \varepsilon \Delta_2} 1/\Phi_2 \cdot \prod_{k \varepsilon \Delta_3} 1/\Phi_3$$

Rewriting the likelihood function, L, we have

$$L = \prod_{\gamma \varepsilon \Gamma} (1/2\pi)^{3/2} \, |\Sigma|^{-1/2} \exp\{ -\tfrac{1}{2}\theta_{\gamma}^{t}(x,y)\, \Sigma^{-1}\, \theta_{\gamma}(x,y) \}$$
$$\bullet \quad (1/\Phi_1)^{m_1-n} \quad \bullet \quad (1/\Phi_2)^{m_2-n} \quad \bullet \quad (1/\Phi_3)^{m_3-n}$$

In order to maximize L over the parameter space it is sufficient to compute:

$$\sup_{\substack{\gamma \varepsilon \Gamma \varepsilon \mathbf{A} \\ (x,y)_{\gamma}}} \operatorname{Ln} L$$

where $\mathbf{A}$ is the collection of all possible partitions Γ.

It is then sufficient to find the parameters Γ and $\{(x,y)_{\gamma}\}$ which satisfies:

$$\sup_{\Gamma \varepsilon A} \sum_{\gamma \varepsilon \Gamma} \{ 6 \ln(\Phi_1 \Phi_2 \Phi_3) - 3 \ln(2\pi) - \ln|\Sigma| - \theta_{\gamma}^{t}(x,y)\, \Sigma^{-1}\, \theta_{\gamma}(x,y) \}$$

For any fixed partition, Γ, the function L is maximized when the (non-linear) least-squares solution , (x_e, y_e), for (x,y) is computed. It is then sufficient to find Γ which maximizes the following function.

$$\sup_{\Gamma \varepsilon A} \sum_{\gamma \varepsilon \Gamma} \{ 6 \ln(\Phi_1 \Phi_2 \Phi_3) - 3 \ln(2\pi) - \ln|\Sigma| - \boldsymbol{\theta}_{\gamma}^{t}(x_e, y_e)\, \Sigma^{-1} \boldsymbol{\theta}_{\gamma}(x_e, y_e) \} \qquad (1)$$

This maximization problem can be placed in the context of a zero-one integer programming problem as follows.

Define

$$c_{ijk} = 6 \ln(\Phi_1 \Phi_2 \Phi_3) - 3 \ln(2\pi) - \ln|\Sigma| - \boldsymbol{\theta}_{\gamma}^{t}(x_e, y_e)\, \Sigma^{-1} \boldsymbol{\theta}_{\gamma}(x_e, y_e) \qquad (2)$$

Then equation (1) can be rewritten as

$$\left.\begin{array}{ll}
\text{maximize} & \displaystyle\sum_{i=1}^{m_1} \sum_{j=1}^{m_2} \sum_{k=1}^{m_3} c_{ijk}\, \delta_{ijk} \\
\text{subject to:} & \displaystyle\sum_{i} \sum_{j} \delta_{ijk} \le 1 \\
 & \quad \text{for all k} \\
 & \displaystyle\sum_{i} \sum_{k} \delta_{ijk} \le 1 \\
 & \quad \text{for all j} \\
 & \displaystyle\sum_{j} \sum_{k} \delta_{ijk} \le 1 \\
 & \quad \text{for all i} \\
\text{where:} & \delta_{ijk} = 0 \text{ or } 1.
\end{array}\right\} \quad (3)$$

Preassignments can be placed on (3) since if $c_{ijk} \le 0$ we preassign $\delta_{ijk} = 0$ as this always gives a larger value to the objective function.

Taking this preassignment into account, we then place the problem , (3), in the standard form of a particular zero-one integer programming problem known as set packing (see Balas and Padberg [1]) as follows:

Let $\{\alpha\}$ be any enumeration of the set $\{ c_{ijk} | \, c_{ijk} > 0\}$. Then (3) is equivalent to:

$$\begin{array}{ll}
\text{maximize} & \displaystyle\sum_{\alpha=1}^{N} c_{\alpha}\, Z_{\alpha} \\
\text{subject to} & \displaystyle\sum_{\alpha=1}^{N} \mathbf{A}_{\alpha}\, Z_{\alpha} \le \mathbf{1} \\
\text{where} & Z_{\alpha} = 0 \text{ or } 1,\ c_{\alpha} > 0, \mathbf{1} \text{ is a vector of all ones} \\
\text{and} & \mathbf{A}_{\alpha} \text{ is a vector of all zeroes and ones.}
\end{array}$$

for suitable choices of $\mathbf{A}_{\alpha}$. In vector notation this formulation may be written as:

$$\left.\begin{array}{ll}
\text{maximize} & \mathbf{c}^{t}\mathbf{Z} \\
\text{subject to} & A\,\mathbf{Z} \le \mathbf{1} \\
\text{where} & z_i = 0 \text{ or } 1 \\
\text{and} & \text{A is an mxn matrix} \\
 & \text{of all zeroes and ones.}
\end{array}\right\} \quad (4)$$

The formulation (4) can be solved using any set partitioning algorithm (Pierce [6] and Pierce and Lasky [7] give algorithms for solution). However, the above problem must first be reformulated into a set partitioning problem by changing the less-than constraints to equality constraints via slack variables and then changing the maximization problem to a minimization problem via the transformation : $\mathbf{c}^* = k A^t \mathbf{1} - \mathbf{c}$, where k is suitably chosen so that $\mathbf{c}^* > 0$.

Computational considerations can be made by eliminating more variables in the set packing problem (4). Variables were previously eliminated when the cost associated with a variable was negative. Further elimination can be made if the cost is too small though still positive. This may result in a suboptimal data arrangement but the computational benefits of a smaller programming problem may outweigh the disadvantages of suboptimality.

Simulation and Results

A simulation program has been written to test the algorithm using data from a network of three jammed radars. A radar is jammed when an aircraft transmits noise toward the radar at the same frequency as the radar's transmitter, thus denying the radar any range information. Targets were randomly placed in a rectangular region as shown in figure 1. The azimuth angle from each target to each radar and Gaussian random errors added to the measurements (σ_m = .25°). To further model an actual jammed radar a resolution model was employed. When two targets detected by a radar had an angular difference of less than one degree the data were averaged and only one line-of-bearing as reported. The data were then entered in the algorithm with results shown below. Figure 2 shows the likely target positions before the integer programming has been run (target estimates such that $c_{ijk} > 0$). True target positions are also shown. Figure 3 shows the final estimated target positions as the solution to the integer programming problem. Figure 4 shows the average performance of the algorithm reflecting over 100 Monte Carlo runs. Notice that overall performance does degrade as the number of targets increases but the percentage of ghost targets does not increase. In fact, much of the degraded performance can be attributed to the resolution model which denies a significant amount of data to the algorithm.

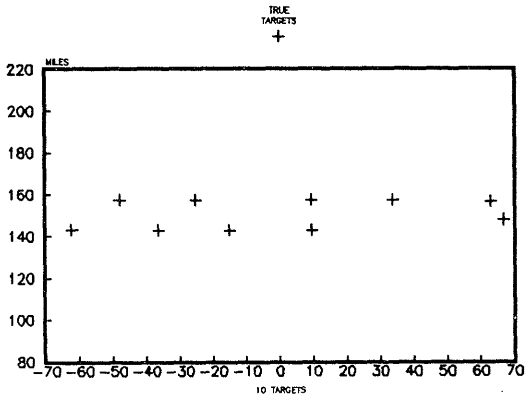


Figure 1. The True Targets

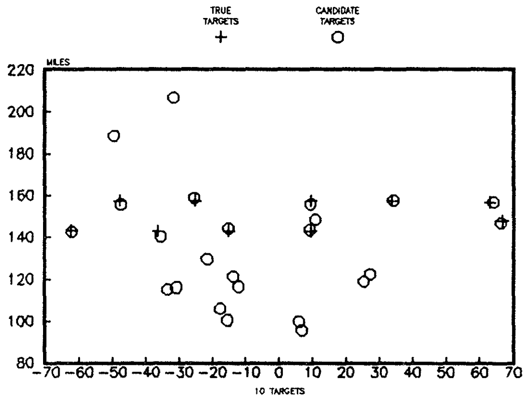


Figure 2. All Feasible Target Estimates ($C_{ijk} > 0$)

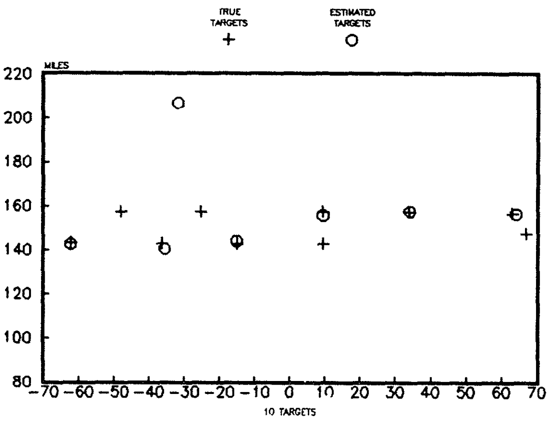


Figure 3. Integer Programming Solution

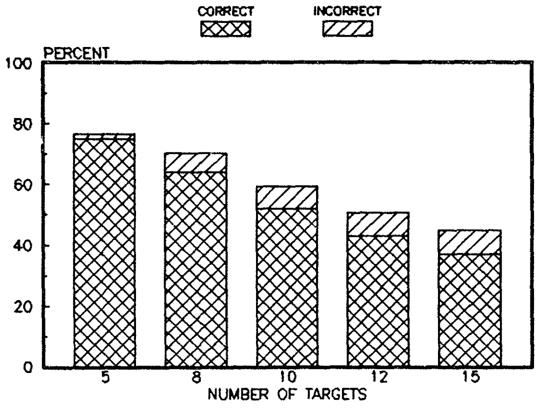


Figure 4. Average Algorithm Performance

Implementation in a Parallel Computer Architecture

The algorithm described above can be considered in two modules. The first module is the setup of the integer programming problem or the search for likely target postions. The second module solves the integer programming problem. The first module must be completed before the second starts so no parallelism is possible between the modules. However, the processes within each module can be executed, to a large degree, in parallel.

The first module searches for likely target positions by considering all possible combinations of the data from the three sensors (or triple intersections). This implies a search of n^3 possible combinations of data for n targets. Since all combinations must be considered (at least implicitly) the combining process may be performed concurrently. The parallel architecture considered used n processors, each processor operating on a single line-of-bearing from a single sensor and searching over the n^2 possible combinations of data from the other two sensors.

The second module solves the integer programming problem using a tree-search-type algorithm. The tree structure is shown below in figure 5.

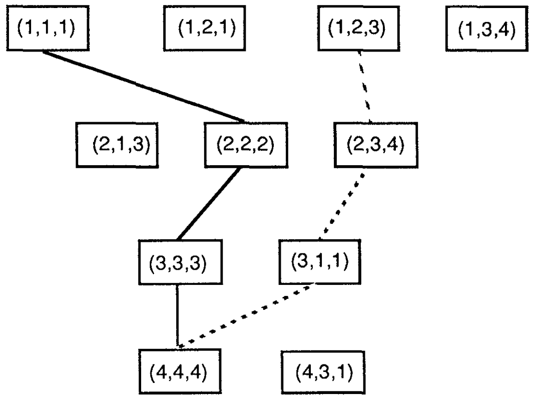


Figure 5. The Tree Structure of the Set Partitioning Algorithm

The triple at each node represents the feasible target position (triple intersection, γ). The integer programming algorithm must search down this tree for feasible solutions. A solution is feasible only when the triples at each node in the assignment do not share a value in the same place. The lines drawn represent two possible assignments or feasible solutions to the programming problem. By assigning a single processor to each node at the top of the tree, multiple paths can be checked simultaneously.

Conclusion

A method for locating multiple targets with multiple bearing-only sensors has been presented and is shown to provide accurate target position estimates. The major limitation of the method is that it utilizes significant computer resources. However, the algorithm is well suited to implementation in a parallel computer architecture and should provide sufficient computer resources for real-time tracking applications.

References

[1] E. Balas, and M. Padberg, "Set Partitioning, a Survey," *SIAM Review*, Vol. 18, pp. 710-760, Oct. 1976.

[2] R. Lafrance and W. Rivers, "Jammer Location Technique using Radar Passive Angle Tracks(U), " Technology Service Corp., Silver Spring MD, AD-C020688, Aug. 1979 (CONFIDENTIAL).

[3] C. L. Morefield, "Interim report on Multitarget Tracking using Multiple Acoustic Sensors (U)," Orincon Corp. La Jolla CA, AD-C028772, June 1976 (SECRET).

[4] _______, "Application of 0-1 Integer Programming to Multitarget Tracking Problems," *IEEE Trans. Automat. Contr.*, Vol. AC-22, pp. 302-312, June 1977.

[5] _______, "Multitarget Tracking for SASE (U)," vol I and II, Orincon Corp., AD-C028769 and C028770, Dec. 1978 (SECRET).

[6] J. F. Pierce, "Application of Combinatorial Programming to a Class of All-Zero-One Integer Programming Problems," *Management Science*, Vol. 15, pp. 191-209, Nov. 1968.

[7] J. F. Pierce and J. S. Lasky, "Improved Combinatorial Programming Algorithms for a Class of All-Zero-One Integer Programming Problems," *Management Science*, Vol. 19, pp. 528-543, Jan. 1973.

REGISTRATION TECHNIQUES FOR MULTIPLE SENSOR SURVEILLANCE

Martin P. Dana

Hughes Aircraft Company
Command and Control Systems Division
P.O. Box 3310
Fullerton, CA 91634

In order to integrate multiple sensor data into a single air picture, the individual sensor data must be expressed in a common coordinate system, free from errors due to site uncertainties, antenna orientation, and improper calibration of range and time. The process of ensuring the requisite "error free" coordinate conversion of sensor data is called registration. This paper develops a Generalized Least-Squares Estimation technique for sensor registration and compares quantitatively this technique with of some the standard methods in use today.

1. Background

Modern Command and Control systems depend on a surveillance subsystem to provide an air situation picture on which decisions must be based. In order to maintain an accurate, complete and current air picture, the surveillance subsystem will, in turn, depend on combinations of netted sensors to provide the raw data from which the air situation picture is developed. To date, unfortunately, attempts to net multiple sensors into a single surveillance system have met with limited success, due in large part to the failure to register adequately the individual sensor (see Ref. [1]).

Why the registration of multiple sensor systems has been, in general, inadequate is not easily explained. The problem does not seem to be understood or even recognized beyond a small circle of systems engineers at a few Government laboratories and aerospace companies. Certainly it has not received the attention which, for example, the problem of tracking or state estimation has received. Literally thousands of papers have been published on Kalman filtering; many excellent (as well as mediocre) texts have been written on the subject of optimal estimation. Publications on the registration problem are limited to a few technical reports funded by various Department of Defense agencies. Registration, it seems, has been an afterthought in most system efforts.

The purpose of this paper is two-fold: first, to define the registration problem in terms of the sources of registration error and their implication on multi-sensor target tracking; and, second, to provide a solution of the registration problem. The solution of the problem discussed below is based on the techniques of multivariate statistical analysis. Thus, there is an obvious parallel between this solution to the registration problem and the Kalman filter to the extent that both can be derived from the theory of Generalized Least-Squares Estimation. More importantly, however, the solution discussed below treats the problem with a similar level of detail and sophistication as has been applied to the tracking function.

Since radars are still the primary sensors in use today, and since the problem of radar registration has not yet been resolved adequately, this paper will address the problem of radar registration only. The same principles can be applied to sensor networks which include other kinds of sensors.

2. The Registration Problem

The fundamental problem in sensor netting to determine whether data reports from two or more remotely located sensors represent a common aircraft or a distinct aircraft. Before this can be accomplished successfully, however, the individual sensor data must be expressed in a common coordinate system, free from errors due to site uncertainties, antenna orientation, and improper calibration of range and time. The process of ensuring the requisite "error free" coordinate conversion of sensor data is called registration. Thus, registration is an ABSOLUTE prerequisite for sensor netting.

The major sources of registration error for radars are listed below in the left-hand column of Table 1, together with some possible corrective actions in the right-hand column.

TABLE 1. Registration Error Sources

Error Source	Corrective Measure
Range:	
Offset	Test Target
Scale	
Atmospheric Refraction	Tabular Corrections
Azimuth	
Offset	Solar Alignment; North Finders
Antenna Tilt	
Elevation (3-D Radars):	
Offset	
Antenna Tilt	
Time:	
Offset	
Scale	
Radar Location:	JITDS, PLRS, GPS, Satellite Survey
Coordinate Conversations:	
Radar Plane	
System Plane	

Source: Fischer, Muehe, Cameron: Registration Errors in a Netted Air Surveillance System (Ref. [1]).

Of the sources of registration error listed in Table 1, there are three sources which have proved to be major problems in air defense systems; they are: (1) position of the sensor with respect to a "system" coordinate origin; (2) alignment of the antennas with respect to a common north reference (i.e., the azimuth offset error); and (3) range offset errors. The other errors may exist in the current radar systems; however, they have not been significant problems in the past. As the radar technology improves, some of the other error sources may become significant factors.

As suggested in Table 1, electronic position-location systems such as GPS or commercial satellite survey systems are available which can locate a position on the earth to within a maximum error of 100 feet (or better). This accuracy is certainly adequate for radar systems in which the standard deviation of the range measurement error is no better than 0.125 nmi. The problem is now to deal with range and azimuth offset errors.

The potential effects of range and azimuth offset errors are illustrated in Figure 1. Registration errors are systematic measurement errors rather than random errors. The figure illustrates the expected or average reports of a common target from two radars, each of which consistently reports (1) a range less than the true range by a fixed amount (the offset) and (2) an azimuth (measured clockwise from north) less than the true azimuth by a fixed offset. For any specific set of measurements, random measurement errors will be superimposed on the bias errors.

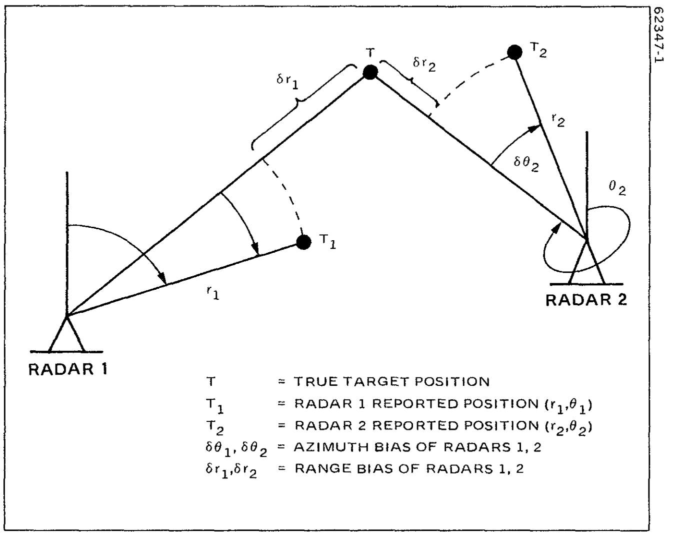

62347-1
T
δr_1
δr_2
T_2
r_2
$\delta\theta_2$
θ_2
T_1
r_1
RADAR 2
RADAR 1
T = TRUE TARGET POSITION
T_1 = RADAR 1 REPORTED POSITION (r_1,θ_1)
T_2 = RADAR 2 REPORTED POSITION (r_2,θ_2)
$\delta\theta_1, \delta\theta_2$ = AZIMUTH BIAS OF RADARS 1, 2
$\delta r_1, \delta r_2$ = RANGE BIAS OF RADARS 1, 2

Figure 1. Range and Azimuth Registration Errors. Registration errors introduce measurement biases into the system; this will result in degraded tracking performance or even in the initiation of multiple tracks for a single target.

The effect of systematic errors is to introduce biases into the estimation process. Therefore, failure to register a multiple radar system adequately can result in varying degrees of performance degradation, depending on the magnitude of the error or bias with respect to the random measurement errors and the tracking gates. The level of degradation ranges from the formation of multiple, redundant tracks for a single aircraft to reduced track accuracy and stability, or simply the loss of the benefits of multiple radar tracking by reducing the system, in effect, to a single radar tracking system.

3. Registration Procedures

System registration may be considered as a two phase process: sensor initialization and relative alignment. The objective of the initial registration procedure is to register, with respect to absolute coordinates, each sensor independently. Once the position of the sensor has been estimated, the range measurements have been calibrated, and an initial alignment with respect to true north has been completed, the procedures for relative alignment of the system sensors can be initiated. The initialization procedures generally are straightforward; the REAL registration problem is the relative alignment of the system sensors.

Techniques for relative registration depend on common targets, preferably targets of convenience rather than controlled flights. Generally, data is collected until a sufficient number of paired reports have been obtained, and then a set of bias corrections are computed. The usual techniques for obtaining the solutions is either to formulate the problem as an ordinary (or unweighted) least-squares estimation (LSE) problem or to rely on simple averaging to remove the random error components. The major limitation of either approach is that each radar report is treated equally when, in fact, the measurement (i.e., observation) errors are a function of both the individual radar parameters and target range.

The least-squares approach is commonly employed in the NATO air defense systems. This approach obtains a relative solution for a subordinate radar with respect to a master radar, which is assumed to be perfect. Since there is no particular reason to believe that the master radar is "perfect", this approach can only verify that the initial alignment is adequate; estimates of non-zero biases merely indicate that there is a registration error. Range offset errors, in particular, are not relative; a bias at the master site cannot be transferred to the subordinate site.

The alternative approach is the simple averaging process which is employed in the US-Canadian Joint Surveillance System (JSS) for North America and in the FAA National Air Space System (NAS) (for enroute air traffic control). The derivation of this technique assumes a symmetric distribution of points about the line joining the two radars. Consequently, the solutions are very sensitive to the actual target distribution. For radars along many political borders, it may not be possible to obtain any data at all from one side of the line joining the two sites.

Given this situation, it is obvious that a new approach to system registration is needed. The basic objective of this research is to develop a technique for registration with the following characteristics:

- Insensitive to target distribution,
- Applicable to fixed site, mobile and airborne sensor systems,
- Provide alternative solution sets depending on the need,
- Provide a quality estimate for the solution set, and
- Be based on a recognized principle of optimality.

4. Bias Estimation

Fisher, et. al, suggest (Ref. [1], p. 17) three alternative approaches; specifically, the generalized linear least-squares estimation (GLSE) technique and two numerical optimization methods, one based on a grid search technique and the other on Powell's method for steepest descent. The GLSE is dismissed for computational reasons, and the grid search approach is dismissed in favor of Powell's method because of slow convergence.

Commercial array processors or special purpose co-processors are now available which are capable of performing the large scale matrix operations required by the GLSE approach (see Ref. [2]). Consequently, the GLSE approach is reconsidered in this paper. The technique developed by Wax in Ref. [3] can be applied to formulate the generalized Gauss-Markov problem.

The approach suggested by Wax is to formulate the difference dP in the reported positions as a function of the set of measured variables Z (i.e., observations) and the set of biases B (i.e., parameters) to be estimated: dP = F(Z,B). Following the usual linearization technique, but with the roles of the actual values and estimators reversed, the vector equation for position difference can be transformed in the classical Gauss-Markov GLSE model (see Ref. [2]): $X*B + E = Y$, where X is a matrix of known parameters, E is the vector of measurement errors, and Y is the measurement vector.

The solution of the GLSE Problem above is simply

$$\tilde{B} = (\mathrm{Cov}) * X^T * V^{-1} * Y$$

where $(\mathrm{Cov}) = (X^T * V^{-1} * X)^{-1}$ is the covariance matrix for the estimate $\tilde{B}$ of B.

The difficult part of the formulation is to develop a representation for the covariance matrix V of the measurement error. However, Fischer, et. al., provide a framework for the derivation in Appendix D of Ref. [1]. The details of the derivation of the solution for two range and two azimuth biases are provided in the appendix of this paper.

In general the GLSE approach requires a capability to perform arithmetic with large matrices. For this application, however, the problem may be greatly simplified using the independence of the measurements, both between radars and over the set of targets. As shown in the appendix, the covariance matrix V for the error term E is a block-diagonal matrix; the dimension of the individual blocks is the same as the cardinality of the set Z for the individual samples. For the registration problem, the measurement set Z contains four (4) independent measurements. Therefore, the covariance matrix is the inverse of a sum of 4x4 matrices, which can be computed easily. (See the appendix.)

5. Numerical Evaluation

During the past year the GLSE approach has been formulated and evaluated at Hughes Aircraft Company. The evaluations have considered both theoretical covariance analyses and simulation analyses for comparison of the GLSE technique with the JSS, NATO and the ordinary LSE techniques. Some of the major results of these evaluations are presented below.

5.1 Covariance Analyses.

The GLSE approach was developed for three distinct solution sets; these were the following:

- Two azimuth offset biases,
- Two range offset biases, and
- Two range and two azimuth offset biases.

In the case of the "two azimuth bias" solution, it is assumed that there is a potential azimuth bias at each of the two radars, which are called the master and subordinate for convenience; it is assumed further that there are no range biases at either of the two radars. For the "two range bias" solution, the analogous assumptions were used. The "two range/two azimuth bias" solution is that derived in detail in the appendix.

The covariance matrices for the three alternative solutions sets were analyzed with respect to the number of samples (that is, targets) used in the solution and the distribution of the targets in the (x,y)-plane. The results of the analysis with respect to the sample size are shown in Figures 3, 4 and 5 for the sensor/target geometries illustrated in Figure 2. For these analyses, the standard deviations of the random range and azimuth measurement errors at both radars were assumed to be 0.125 nmi. in range and 0.18 degree (approximately 3.0 milli-radians) in azimuth. These statistics are typical of modern air defense and air traffic control radars.

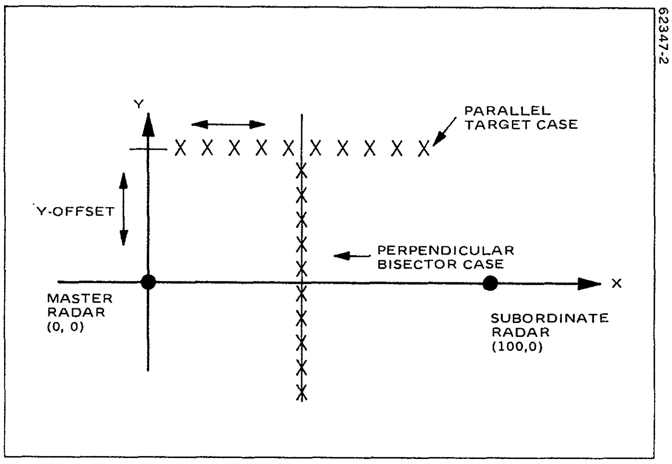


Figure 2. Sensor/Target Geometry. The covariance analyses were conducted for targets distributed symmetrically along the perpendicular bisector of the line segment joining the two radars and for targets distributed along a line parallel to the line segment joining the two radars.

The ratio of the standard deviation of the azimuth bias estimate to the standard deviation of the (random) azimuth measurement error is plotted in Figure 3. From the graph, approximately 125 samples are required in order to obtain a bias-to-measurement ratio of 0.10, which will ensure that any system track inaccuracies are due to the random errors rather than the systematic or bias errors.

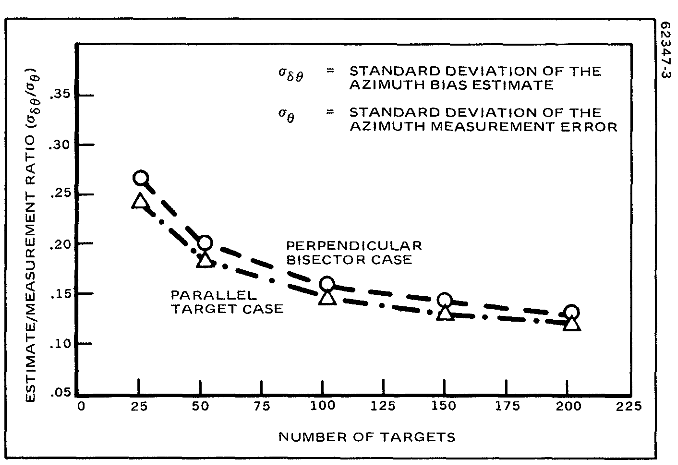


Figure 3. Azimuth Bias Estimation Performance Versus Sample Size. The variance of the azimuth bias estimation error decreases as a function of 1/N, where N is the number of samples.

Similarly, the ratio of the standard deviations of the range bias estimate to the random range measurement error is plotted in Figure 4. From the graph, 100 to 150 samples are required in order to obtain a 5-to-1 improvement ratio. Although this is not as dramatic as the 10-to-1 ratio obtained for the azimuth case, it will be adequate for target tracking. As is the case for the azimuth bias estimate, the variance of the range bias estimate is approximately inversely proportional to the number N of samples.

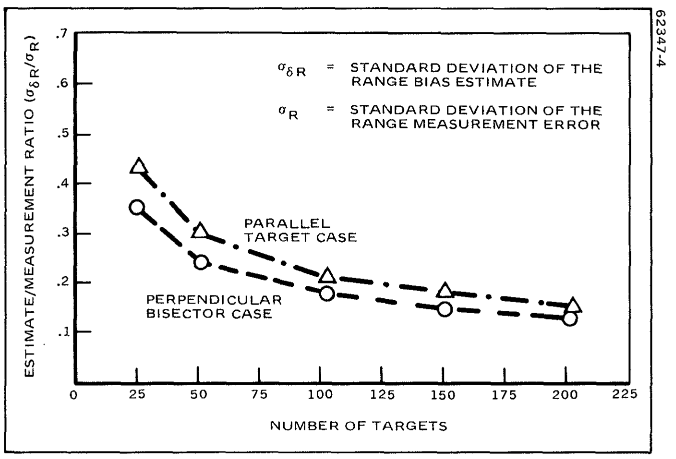


Figure 4. Range Bias Estimation Performance Versus Sample Size. The variance of the range bias estimation error decreases as a function of 1/N, where N is the number of samples.

Finally, the results of the same analyses are shown in Figure 5 for the "two range and two azimuth biases" solution. As was the case for the range-only or the azimuth-only cases, the solutions are relatively insensitive to target distribution. In this case, however, a symmetric distribution along the perpendicular bisector yields the best performance by a factor of at least 2-to-1 (with respect to the parallel target case). If a symmetrically distributed sample cannot be obtained, then 200 or more points may be required in order to obtain satisfactory performance for tracking.

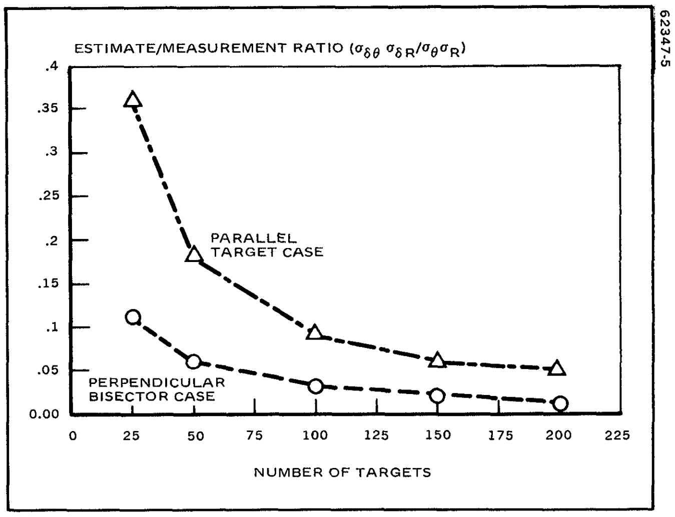


Figure 5. Range and Azimuth Bias Estimation Performance Versus Sample Size. The performance for this case is moderately sensitive to target distribution; however a sample size of 200 will be adequate of tracking in any case.

5.2 Simulation Results

In addition to the covariance analyses described above, simulation analyses were conducted, primarily in order to compare the GLSE algorithms with other techniques (for which covariance estimates are not available). A secondary goal of the analyses were to determine the sensitivity of the current registration assumptions of the techniques. For example, the NATO registration algorithm assumes that there are no range biases at the radars; if there are range biases, then the algorithm will translate them into azimuth biases.

For the simulation data presented in Figures 6 and 7 below, the sensor/target geometry is similar to the "perpendicular bisector case" in the preceding section except that the targets are not symetrically distributed along the bisector. For this analysis, all of the targets are in the first quadrant of the coordinate system.

In Figure 6, six algorithms for estimation of two azimuth biases are compared in terms of the RMS error between the estimated bias and the true bias, which was 0.50 degree. The algorithms included the two relevant versions of the GSLE (or Gauss-Markov) approach discussed in Section 5.1; a trilateration technique which uses only the range measurements from each radar; the JSS method; the NATO (or 407L) method; and an ordinary least-squares algorithm. The NATO algorithm differs from the ordinary LSE algorithm only to the extent that the NATO algorithm solves for an azimuth bias and a position bias of the subordinate radar; the position bias is then translated into an azimuth bias at the master radar.

The results of the analysis are shown in Figure 6, in which the RMS error in the azimuth bias estimates are shown on the ordinate versus an actual range bias on the abscissa. As one would expect, the JSS Algorithm and the GLSE/Gauss-Markov algorithm for the "two azimuth and two range" solution are insensitive to the presence of range biases since both approaches solve for range biases in addition to azimuth biases. The other approaches are derived from the assumption that there are no range biases and, therefore, produce inaccurate azimuth bias estimates when there are range biases, as indicated by the positive slopes for each curve on the graph. If there are no range biases, however, these (azimuth-only) solutions will be somewhat more accurate than the JSS and the GLSE-2 solutions.

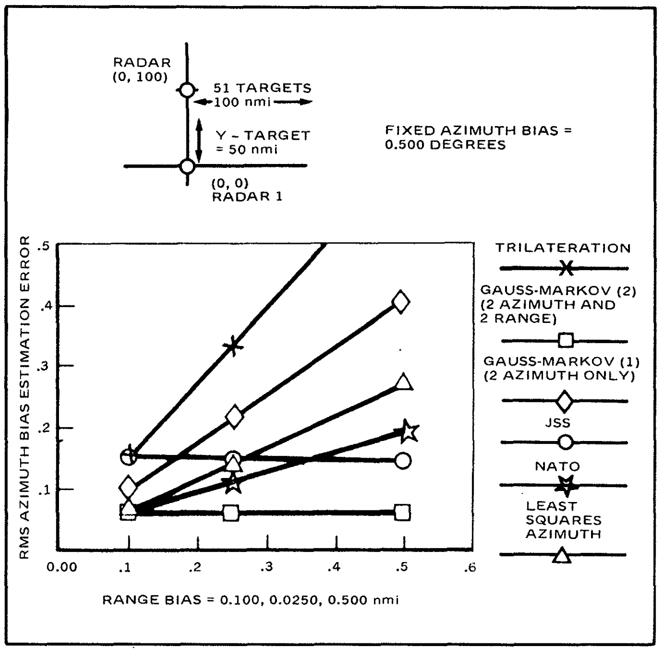


Figure 6. RMS Azimuth Bias Estimation Error versus Range Bias. The JSS and the GSLE-2 algorithms solve for range biases and, therefore, are insensitive to range biased data. The GLSE-2 algorithm can achieve a 50% reduction in estimation error with respect to the JSS approach.

The performance of the range bias estimation techniques is shown in Figure 7 versus an actual azimuth bias. As was the case with azimuth-only solutions, the presence of a bias which is assumed to be non-existent, will degrade performance severely. Also, as before, the GLSE-2 algorithms can achieve a 2-to-1 reduction of the standard deviation of the estimation error with respect to the JSS method.

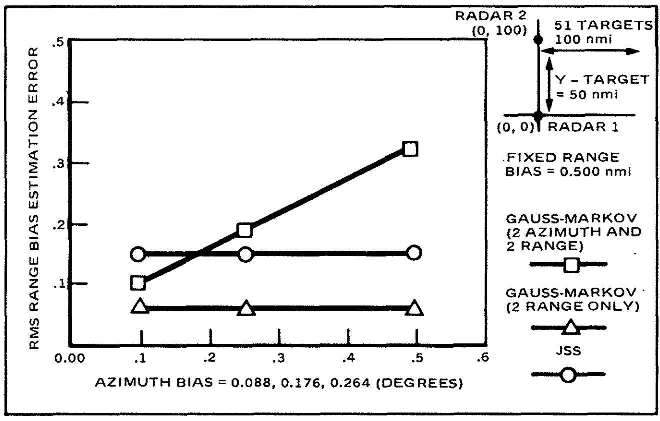


Figure 7. RMS Range Bias Estimation Error Versus Azimuth Bias. The range-only GSLE technique is very sensitive to azimuth biases but is still preferred the JSS method when the magnitude of the azimuth bias is less than 0.20 degree.

Based on the data presented above and extensive analyses which were not included here because of the limited space, the following statements summarize the conclusions of the IR&D work conducted by Hughes Aircraft Company over the past two years. In general, the GLSE approach exhibits modest CPU requirements; the algorithm is certainly practical as an off-line or background capability. More importantly, the GLSE approach is less sensitive to sensor/target geometry than any of the other registration approaches, particularly the JSS approach. In most cases, the GLSE algorithms can achieve satisfactory registration accuracies with 50 to 100 point-pairs rather than the 200 often required by the JSS or NATO algorithms.

References

1. W. L. Fischer, C. E. Muehe and A.G. Cameron, Registration Errors in a Netted Air Surveillance System; MIT Lincoln Laboratory Technical Note 1980-40, 2 Sept. 1980.

2. T. W. Anderson: An Introduction of Multivariate Statistical Analysis; John Wiley & Sons, Inc., 1958.

3. Mati Wax: "Position Location from Sensors with Position Uncertainty," IEEE Trans. on Aerospace and Electronic Systems, Vol. AES-19, No. 5 (Sept., 1983).

Appendix: Mathematical Development

In the following derivation, assume that a master radar R(1) is located at the origin of the coordinate system and that a subordinate radar R(2) is located at coordinates (u,v). For this derivation, it is immaterial which radar is the master and which is the subordinate. Also assume that there are N targets in the intersection of the respective fields of view, denoted by T(1), T(2), ... T(N). (See Figure A.)

The basic problem is to determine the range and azimuth biases at each radar from the measurements of the common targets T(1), T(2), ... T(N). That is, it

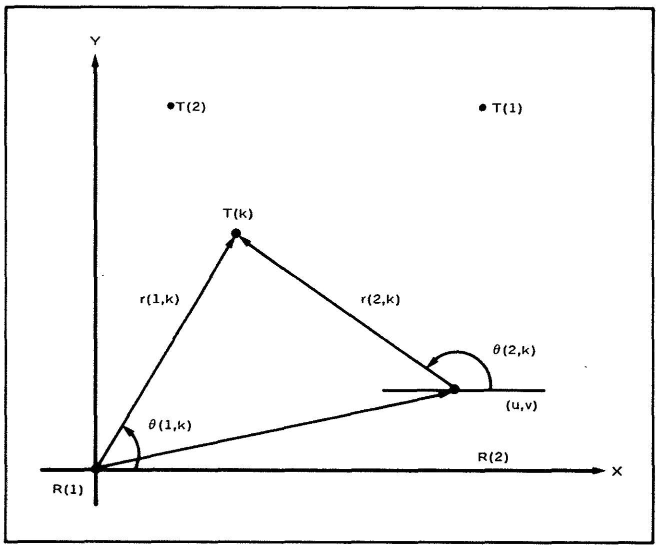


Figure A. Sensor Measurement Geometry. The registration algorithm must determine the system biases for the measurement set X(k) = {r (1, k), θ (1, k), r (2, k), θ (2, k)} .

is necessary to estimate the azimuth biases a(1) and a(2) at R(1) and R(2), respectively, and the range biases b(1) and b(2) at R(1) and R(2). Denote the set of biases by

$$\beta = \{ a(1), a(2), b(1), b(2) \} \qquad (1)$$

For each target T(k), define the set of radar measurements

$$\psi (k) = \{ r(1,k), \theta(1,k), r(2,k) \theta(2,k)\} \qquad (2)$$

where r(1,k), θ(1,k) and r(2,k), θ(2,k) denote the range and azimuth measurements from radar R(1) and radar R(2), respectively.

For each set of measurements, ψ(k), the observations are the separations in the (x,y)-plane of the reported target positions. These are:

$$dx(k) = [r(1,k) + b(1)] \cos [\theta(1,k) + a(1)] - u$$
$$- [r (2,k) + b(2)] \cos [\theta(2,k) + a(2)] \qquad (3)$$

$$dy(k) = [r(1,k) + b(1)] \sin [\theta(1,k) + a(1)] - v$$
$$- [r(2,k) + b(2)] \sin [\theta(2,k) + a(2)] \qquad (4)$$

Equations (3) and (4) above relate the set β of parameters to be estimated to the set of measurements ψ(k) and the vector of observations [dx(k), dy(k)]. However, these functional relationships are non-linear.

In order to apply the Gauss-Markov theory of Generalized Least Squares Estimation (GLSE), it will be necessary to represent the observations as a linear function of the parameters to be estimated, namely β. This can be accomplished by defining a function f as follows:

$$f(\psi(k), \beta) = [dx(k), dy(k)]^T$$

where the superscript T denotes the transposition of the vector (or, later, the matrix). Further, let ψ'(k) and β' denote the actual measurement sets and an initial estimate of β, respectively. Now, Taylor's Theorem can be used to approximate the function f at the true values of ψ(k) and β in terms of the measurements ψ'(k) and the initial estimate β':

$$\begin{aligned} f(\psi(k), \beta) = & f(\psi'(k), \beta') \\ & + \nabla_\beta f(\psi'(k), \beta') (\beta - \beta') \qquad (5) \\ & + \nabla_\psi f(\psi'(k), \beta') [\psi(k) - \psi'(k)] \end{aligned}$$

where

$$F(k) = \nabla_\psi f[\psi'(k), \beta'] \qquad (6)$$

$$= \begin{bmatrix} \frac{\delta\ [dx(k)]}{\delta r(1,k)} & \frac{\delta[dx(k)]}{\delta\theta(1,k)} & \frac{\delta[dx(k)\ ]}{\delta r(2,k)} & \frac{\delta[dx(k)]}{\delta\theta(2,k)} \\ \frac{\delta[dy(k)]}{\delta r(1,k)} & \frac{\delta[dy(k)]}{\delta\theta(1,k)} & \frac{\delta[dy\ (k)]}{\delta r(2,k)} & \frac{\delta[dy(k)]}{\delta\theta(2,k)} \end{bmatrix}$$

and

$$G(k) = \nabla_\beta f(\psi'(k), \beta') \qquad (7)$$

$$= \begin{bmatrix} \frac{\delta[dx(k)]}{\delta a(1)} & \frac{\delta[dx(k)]}{\delta a(2)} & \frac{\delta[dx(k)]}{\delta b(1)} & \frac{\delta[dx\ (k)]}{\delta b(2)} \\ \frac{\delta[dy(k)]}{\delta a(1)} & \frac{\delta[dy(k)]}{\delta a(2)} & \frac{\delta[dy(k)]}{\delta b(1)} & \frac{\delta[dy\ (k)]}{\delta b(2)} \end{bmatrix}$$

If the errors [ψ(k) - ψ'(k)] and (β - β') are sufficiently small that the higher order terms can be neglected, then the approximation in (5) may be regarded as an equality. Also, note that

$$f(\psi(k), \beta) = 0 \qquad (8)$$

by definition; therefore:

$$G(k)\beta + F(k)\, d\psi'(k) = G(k)\, \beta' - f(\psi'(k), \beta') \qquad (9)$$

where dψ'(k) = ψ'(k) - ψ(k). Note that the matrix G(k) is a matrix of known parameters, F(k) dψ'(k) is the error due 'to the measurement noise, and that the terms on the right-hand side of equation (9) now represent the observations.

With all of this notation and the approximation of equation (5), equations (3) and (4) may now be reformulated as the classical Gauss-Markov model of GLSE theory:

$$X\, \beta + \varepsilon = Y \qquad (10)$$

by setting

$$X = [G(1), G(2), \dots G(N)]^T \qquad (11)$$

$$\varepsilon = [F(1)\, d\psi'(1), F(2)\, d\psi'(2), \dots F(N) d\psi'\ (N)]^T \qquad (12)$$

$$Y = [G(1)\, \beta' - f(\psi'(1), \beta'), G(2)\, \beta' - f(c'(2), \beta'), \dots\dots G(N) \dots]^T \qquad (13)$$

Note that X is a 2Nx4 matrix, ε is 2N vector, and that the observation vector is also of dimension 2N.

The last step in this application of the Gauss-Markov model is to develop the covariance Σ_ε matrix for the error vector ε. To this end, define:

$$\Sigma_\varepsilon = E[\varepsilon\varepsilon^T] = \mathrm{diag}\ \{F(k)\ E[d\psi')(d\psi')^T]\ F^T(k)\} \qquad (14)$$

where the notation "diag" indicates a diagonal matrix with the non-zero terms enclosed in the brackets. Note that

$$\Sigma_\psi = E \left[(d\psi')(d\psi')^T\right] = \mathrm{diag} \left[\sigma^2_{R(1)}, \sigma^2_{\theta(1)}, \sigma^2_{R(2)}, \sigma^2_{\theta(2)}\right] (15)$$

Further, note that F(k) is a 2 x 4 matrix and that Σ_ψ is a 4 x 4 matrix; therefore

$$\Sigma_k = F(k)\ \Sigma_\psi\ F^T(k) \qquad (16)$$

is a 2 x 2 matrix. This implies that Σ_ε

is a block-diagonal matrix of the form

$$\Sigma_\varepsilon = \mathrm{diag} \left[\Sigma_1, \Sigma_2, \dots \Sigma_N\right]. \qquad (17)$$

The solution of the Gauss-Makrov equation (10) is simply

$$\hat{\beta} = (X^T\ \Sigma_\varepsilon^{-1}\ X)^{-1}\ X^T\ \Sigma_\varepsilon^{-1}\ Y \qquad (18)$$

where

$$\mathrm{Cov}\ \ (\hat{\beta}) = (X^T\ \Sigma_\varepsilon^{-1}\ X)^{-1}. \qquad (19)$$

Since Σ_ε is a 2NX2N block-diagonal matrix, it follows that

$$X^{T} \Sigma_{\varepsilon}^{-1} X = \sum_{k=1}^{N} G^{T}(k) \Sigma_{k}^{-1} G(k) \qquad (20)$$

where the individual terms of the sum are 4X4 matrices. Similarly

$$X \Sigma_{\varepsilon}^{-1} Y = \sum_{k=1}^{N} G^{T}(k) \Sigma_{k}^{-1} [G(k)\beta' - f(\psi'(k),\beta')] \qquad (21)$$

If the individual radar measurement errors are normally distributed, then $d\psi'(k)$ is a normally distributed vector; and $F(k)\ d\psi'(k)$ is a linear combination of normal variables and is, therefore, normally distributed. Thus ψ is $N(0,\Sigma_{\psi})$.

Equation (18) is the minimum variance solution under any error distribution. For the normal distribution (i.e., $\varepsilon \sim N(0, \Sigma_{\varepsilon})$), $\hat{\beta}$ is also the maximum likelihood solution. By these criteria, $\hat{\beta}$ in (18) is the "best" solution to the problem as defined by equation (10):

$$Y = X\beta + \varepsilon$$

where the error term ε is distributed as $N(0,\Sigma\varepsilon)$, X is a matrix of known parameters, and Y is the vector of observations.

A SYSTEMS THEORY APPROACH TO SURVEILLANCE AND C^{3}I

S. Gardner, F. Polkinghorn

Naval Research Laboratory
Washington, D.C. 20375

R. Dawes

Martingale Research Corporation
Plano, Texas 75075

Summary

A systems theory approach is required in order to establish an effective design methodology for applications in active and passive electromagnetic and acoustic surveillance, electronic warfare and C^{3}I. We concentrate on the noncooperative surveillance environment, where a receiving system may be required to detect, localize, and identify signals of unknown modulation. Several major paradigms are discussed. These include a Hilbert resolution space (HRS) theory of communication, non-convex optimization algorithms based upon simulated annealing and mapping of non-convex optimization algorithms onto nonlinear dynamical system architectures.

1. Introduction

New techniques of generalized demodulation and signal processing are required for effective electronic surveillance of noncooperative targets. The bulk of existing communications theory has been developed assuming a cooperative environment where the transmitter and receiver are jointly designed to meet the requirements of information transfer across a given channel. In a noncooperative environment, a transmitter may be required to put a signal through a channel without subjecting itself to discovery or it's message to interception. Conversely, a surveillance system may be required to detect, identify, and localize signals of unknown modulation in harsh and variable interference through a channel which is unsuitable for communications. A new communications theory is required in order to uniformly represent man-made forms of modulation (intentional and non-intentional) as well as propagation effects such as multipath distortion and fading. Ideally this theory should treat signal modulation, both wanted and unwanted, in an identical manner. In addition, on the assumption that appropriate signal features are available, we require robust methods of non-convex optimization for determination of minimal cost solutions to detection, identification, and localization problems.

In cases of practical interest, a propagation channel may have several favored paths which persist over a period of seconds, or perhaps, minutes. Under these circumstances, a receiver with knowledge of these paths, could, in principle, process a received signal to minimize adverse propagation effects. This form of processing is most effective in active applications, such as radar, where the transmitted signal is 'owned'. However, it is possible to envision passive systems where time-varying channel information is used to update computer-based channel models to separate the effects of propagation (such as multipath and fading) from the transmitter modulation. When so separated the propagation information improves the system locational accuracy and the transmitter information improves the systems identification capability.

In this paper we discuss a systems methodology for surveillance and C^{3}I in terms of several paradigms: a Hilbert resolution space (HRS) communication theory, non-convex optimization based on simulated annealing, and the mapping of optimization algorithms, such as simulated annealing onto dynamical systems architectures. We anticipate that the HRS communications theory paradigm, when fully developed, will provide a representation of signal energy in a multi-dimensional 'feature space'. Given a suitable HRS representation, we expect that the problems of identification and localization can be expressed in terms of non-convex optimization for which simulated annealing methods are appropriate. We discuss HRS communications theory, and non-convex optimization based upon simulated annealing in Sections 2 and 3. In Section 4, we discuss the mapping of non-convex optimization algorithms onto nonlinear dynamical system architectures.

2. Communications Theory in Hilbert Space

HRS communications theory consists of a generalized theory of modulation together with a compatible model for the communications channel. A major advantage of using a common representation for modulation, channel noise, and demodulation processes is simplification in the definition and synthesis of new signal processing algorithms and architectures which do not critically depend upon apriori information regarding transmitter and channel modulations.

The customary approach to surveillance in a noncooperative environment is to repeatedly select a set of modulators or demodulators which have been developed for cooperative applications and vary their parameters (with spectral spreaders, adaptive interference cancellers, correlators, Bessel function evaluators, etc.) until one of them does something useful or recognizable to the signal of interest. If the signal of interest has been previously catalogued, then some preliminary spectral estimation will narrow the field of possibilities; but if it has not been catalogued, then the signal analyst must (generally manually) apply his art and a new box must be designed. This procedure is 'ad-hoc' and can be costly both in terms of money and in terms of development/deployment time that equates to lost opportunities for acquisition of intelligence data .

The lack of unified mathematical methods for communication theory has been a culprit in this state of affairs, and a resolution of this problem has been sought for a long time [2-6]. The models which are presented below were inspired by a paper of Bedrosian [2] and represent an operator generalization of his functional representations. A more complete presentation of the HRS theory is given by Dawes [1].

Consider a "signal" x(t) as a finite-energy function of a time variable. That is, the space of all signals is identified with $L^2(\mathbf{R},\mathbf{C})$, which is the Hilbert space of complex valued Lebesgue square-integrable functions on the real line; Sobolev spaces are physically more appropriate, but the corresponding spectral theory is more complex. A 'message' is defined as any bounded and measurable function m(t) of the time variable (i.e., any member of $L^\infty(\mathbf{R},\mathbf{C})$. It is reasonable not to impose finite energy constraints on abstractions such as "I love you". However, the real reason for putting all messages into $L^\infty(\mathbf{R},\mathbf{C})$ as opposed to some other abstract space is that the Gel'fand theory provides one way to construct modulating operators on the space of signals from members of $L^\infty(\mathbf{R},\mathbf{C})$; although this is not the only potential solution to the problem.

We define a 'modulator' as any causal bounded linear operator on $L^2(\mathbf{R},\mathbf{C})$ whose pseudoinverse is causal or can be made so upon right multiplication by some delay operator. We shall allay the consternation of the reader who objects that FM and other angle modulation techniques are not linear forthwith, and then proceed to discuss some of the implications.

$$\text{Let } x(t) = \begin{cases} \exp(i\,\omega t) & \text{for all } t \text{ in some large interval of interest} \\ 0 & \text{elsewhere} \\ & \text{(to preserve finite energy)} \end{cases} \qquad (2.1)$$

Let m(t) be any message function. Then $\exp(i\,m(t))$ is also a message function, and the operator $P_{\exp(\,i m(t))}$, which multiplies all signals pointwise by $\exp(\,i m(t))$, is a modulator by our definition. It's effect on the "carrier" signal x(t) is

$$\begin{aligned} P_{\exp(\,i m(t))}\, x(t) &= x(t)\exp(\,i m(t)) \\ &= \exp(\,i(\omega t+m(t)) \end{aligned} \qquad (2.2)$$

which is clearly angle modulation with a linear operator. Where it differs from the traditional approach (cf., Bedrosian [2], Papoulis [4], Sakrison [6], Schwartz [7], Voelcker [5], Wozencraft & Jacobs [8]) is that our modulators transform physical signals (e.g., carriers), not messages. The message is incorporated into the modulator via the Gel'fand spectral integral on $L^\infty(\mathbf{R},\mathbf{C}) \dashrightarrow \mathbf{B}(L^2(\mathbf{R},\mathbf{C}))$. In other words (with all due respect to Marshall McLuhan),

The Modulator IS the Message.

From the construction above, the reader should easily be able to construct his favorite modulator, unless he is fond of time base modulators (e.g. PPM). For PPM, it suffices to perform a spectral-like integral **not** on the scalar valued m(t) or $\exp(i\,m(t))$, but on the shift-operator-valued function U(m(t)), where U is defined by

$$U(s)f(t) = f(t-s) \text{ for all real } s,t \quad \text{(right shift for } s>0\text{).} \qquad (2.3)$$

It should not come as a surprise to the mathematically inclined that double sideband modulators (for real-valued messages) are Hermitian operators, whereas angle modulators (for real-valued messages) are unitary operators. And the binary PSK modulators lie right where the modulation theorist expects: They are both Hermitian and unitary.

Unlike the traditional approach to modulation, this formulation has the analytical advantage that it separates the study of modulators from that which is to be modulated. It provides a more abstract home for information than in some space of physical signals, yet it is a home which has an enormous body of analytical structure built up around it. Most known modulation schemes take the form of multiplication operators, yet there is plenty of room for potentially useful alternatives, and there are clearly marked doors showing how to get there. In particular, we will assume that the communication channel can be described with the same model (extended to a space of random signals) in which the messages are constructed by nature rather than by man. And, most importantly in view of our C^3I objectives, it provides the foundation for an organized approach to generalized methods for channel equalization and demodulation in the noncooperative communication environment.

We do not claim to have constructed these methods. At this time, we can only point to some candidate methods which can build on this foundation. It is expected that the HRS approach will ultimately lead to a unified mathematical description of both temperal and spatial modulators applicable to both active and passive surveillance. However a considerable amount of additional work needs to be done.

3. Simulated Annealing for Non-Convex Optimization

In optimizing the detection, localization, identification, and resource allocation functions of a surveillance system, we are faced with the problems of incomplete knowledge of the statistics of critical parameters, missing data and incorrect data. The standard approach to the solution of such a problem is to make a sufficient set of assumptions about the data such that the resulting cost functions have a single minimum. The performance of this convex optimization technique is no better than the validity of these assumptions. The alternative to the convex-by-assumption technique is to incorporate the entire range of assumptions, together with their respective probabilities of being correct, into a single, non-convex cost function. The problem with non-convex cost functions is that they have multiple local minima and the positive identification of a global minimum has historically required vast amounts of computer resources.

Assume, through use of a representation in a suitable high-dimensional feature space, that the multi-dimensional surveillance problem can be reduced to non-convex optimization. We describe a new, potentially highly parallel [9], approach to non-convex optimization based on a stochastic search method called simulated annealing. This analogy with statistical physics was used by Kirkpatrick, to develop a computational method which we call classical simulated annealing (CSA) [11].

CSA is an iterative stochastic search technique which uses the Metropolis Algorithm [12] and the Gibbs-Boltzmann distribution for state generation. Following the analogy with statistical physics, the CSA process consists of first 'melting' the system being optimized at a high effective temperature, then lowering the temperature by stages until the system 'freezes' and no further changes occur. The CSA method is based upon three parts: a 'state-generating function', an 'acceptance function', and a 'cooling schedule'. At each iteration a new state is generated near the current state by the state-generating function. The new state is accepted by the acceptance function depending on its cost relative to the old state. If accepted, the new state becomes the current state for the next iteration. At each temperature, the simulation proceeds long enough for the system to reach an equilibrium condition.

When state generation is based upon the Gibbs-Boltzman distribution, the CSA process is called a 'Boltzmann Machine' [13]. The Boltzmann Machine,because of the Gibbs-Boltzman distribution, requires the temperature during cooling to be inversely proportional to log time; which is computationally slow. To remedy this drawback, since we are not constrained to emulate solid state physics, Szu developed a method of fast simulated annealing (FSA) called the Cauchy Machine [14]. FSA uses the state generating properties of the Cauchy probability distribution to perform simulated annealing with a cooling temperature inversely proportional to time. The Cauchy distribution has a divergent variance which allows fast annealing. Hartley and Szu have shown, using nonergodic theory, that FSA based upon the Cauchy distribution converges to a global optimum [15].

4. Spin Glass Models and Nonlinear Dynamical Systems

An impetus for the development of the simulated annealing method for solution of optimization problems was the close relationship between the behavior of systems with many degrees of freedom in thermal equilibrium at a fixed temperature and the the combinatorial optimization problem of finding the minimum of a given function depending upon many parameters. CSA methods, based upon the Metropolis Algorithm, have been successfully used to find near optimal solutions to non-convex optimization problems such as traveling salesman problems (TSP) and routing of wires on circuit boards [11,16]. The CSA annealing algorithm corresponds to a high temperature regime where, in the case of the TSP problem, the thermodynamic properties (e.g. the average length of a tour) scale differently with the number of cities visited. For the TSP class of problem, a low temperature regime exists where a phase transition can occur [10]. In this regime the thermodynamic properties of the TSP are reminiscent of spin glasses [10, 20].

Since spin glass models can be mapped onto nonlinear dynamical system architectures, it may be possible to solve an important class of non-convex optimization problems with nonlinear dynamical systems. For example, in the neural network analog of a spin glass, bonds between elements located at lattice sites correspond to synaptic connections and spins correspond to neural firing rates. Several non-convex optimization problems, such as the TSP, have been solved using nonlinear dynamical systems (e.g. 'Hopfield' networks [17]). However, at the present time, mapping of non-convex optimization algorithms onto nonlinear dynamical system architectures, remains an art and not a science.

We consider a spin glass model, discussed by Stanley [19], which is based upon the d-dimensional extension of the Heisenberg spin lattice [20]. This model, in the limit of infinite dimensionality, reduces to the Kac-Berlin (K-B) spherical model originated by M. Kac in 1947, and treated rigorously by Berlin and Kac in 1952 [18]. We are motivated by the possibility that the KB model, which is analytically tractable, can be used as a paradigm for mapping a class of non-convex optimization problems, onto nonlinear dynamical systems.

Following Stanley [19], the Hamiltonian for N lattice sites can be written as

$$\mathcal{H}^{S} = - \sum_{<\ell,\, \ell'>} J_{\ell,\ell'}\, S^{d}{}_{\ell} \cdot S^{d}{}_{\ell'} \qquad (4.1)$$

where $S^{d}{}_{\ell} \equiv [\, \sigma_1(\ell) , \sigma_2(\ell) , \sigma_3(\ell) . . . , \sigma_d(\ell)]$ is a d-dimensional vector of length $d^{1/2}$ and $J_{\ell,\ell'}$ represents the coupling between spins at lattice sites ℓ and ℓ' and is assumed to depend upon the distance $|\ell - \ell'|$. Stanley [19] argued that, in the limit N, d –> ∞ the free energy in the model (4.1) is identical to Kac proposed a spherical model of a ferromagnet with a Hamiltonian

$$\mathcal{H}^{KB} = - \sum_{<\ell,\, \ell'>} J_{\ell,\ell'}\, \sigma_{\ell} \sigma_{\ell'} \qquad (4.2)$$

where the spins are continuous scalar variables $-\infty < \sigma_{\ell} < \infty$ subject to the constraint

$$\sum_{\ell} \sigma_{\ell}^{2} = N \ . \qquad (4.3)$$

Kac and Berlin derive a partition function for the Hamiltonian (4.2) which can be used to obtain all thermodynamic functions [20]. Evaluating the partition function by means of a lattice Fourier series, they find that a critical phase transition occurs a temperature where a saddle-point coincides with a branch-point. The saddle point integral is found to be directly related to the lattice Green's function discussed by Montroll and Weiss in connection with Lévy random walks on a d-dimensional lattice [21, 22]. This establishes a direct connection between the theory of Lévy random walks and the KB spherical model. The Lévy random walk theory predicts that a phase transition occurs if and only if the corresponding random walk is transient [20].

The KB spin glass model and the theory of random walks provide new analytical tools for mapping of non-convex optimization problems onto nonlinear dynamical systems architectures. We predict that a class of new optimization algorithms, based upon the d-dimensional Heisenberg model (4.1), will be developed for implementation as nonlinear dynamical systems.

Acknowledgement

The authors gratefully acknowledge the support of the Naval Ocean Systems Center (NOSC) and the Office of Naval Technology (ONT).

References

[1] Dawes R.,"Communications Theory In Hilbert Space", Martingale Research Corporation, Jan. 1986.

[2] Bedrosian,E.,"The Analytic Signal Representation of Modulated Waveforms", Proc. IRE, Vol. 50, October 1952.

[3] Holmes, R. B. ,"Mathematical Foundations of Signal Processing", SIAM Review, Vol. 21, No. 3, 1979.

[4] Papoulis, A. "Random Modulation: A Review", IEEE Trans. ASSP, Vol.31, No. 1, Feb. 1983.

[5a] Voelcker, H. B., "Toward a Unified Theory of Modulation",Part I., Proc. IEEE, Vol. 54, No.3, March 1966.

[5b] Voelcker, H. B., "Toward a Unified Theory of Modulation",Part II.,Proc. IEEE, Vol. 54, No.5, May 1966.

[6] Sakrison,D.J.,"Notes on Analog Communication",Van Nostrand Reinhold, New York, 1970.

[7] Schwartz, M., "Information Transmission, Modulation, and Noise", New York, McGraw Hill, 1980.

[8] Wozencraft, J.M., Jacobs, I. M.,"Principles of Communication Engineering", Wiley, 1965.

[9] Geman, S., Geman, D.,"Stochastic Relaxation, Gibbs Distributions, and the Baysian Restoration of Images", IEEE Pattern Anal. Machine Intel., Vol.PAMI-6, November 1984.

[10] Vannimenus, J., Mezard, M., "On the Statistical Mechanics of Optimization Problems of the Traveling Salesman Type", J. Physique Letters, Vol. 45, No. 24, Dec.1984.

[11] Kirkpatrick S., C. Gellatt, Jr., M. P. Vecchi, "Optimization by Simulated Annealing", Science 220, 671-680 ,1983.

[12] Metropolis, A., Rosenbluth, M., Rosenbluth, A., Teller, A., Teller, E., J. "Equation of State Calculations by Fast Computing Machines", J.Chem. Phys.21, 1087,1953.

[13] Sejnowski, T., Hinton, G., "Separating Figure from Ground with a Boltzmann Machine", in Arbib, M.A. & Hansen A. R. eds., "Vision, Brain and Cooperative Computation", MIT Press, Cambridge, 1985.

[14] Szu,H.,Hartley, R.,"Fast Simulated Annealing with Cauchy Probability Densities", preprint (submitted for publication).

[15] Hartley, R., Szu, H., "Generalization and Analysis of the Simulated Annealing Algorithm", preprint (submitted for publication).

[16] Hopfield, J., Tank, D., "Neural Computation of Decisions in Optimization Problems", Biol. Cybern. 52, 141-152 ,1985.

[17] Hopfield,J.,"Neural Networks and Physical Systems with Spontaneous Emergent Collective Computational Abilities",Proc. Nat. Acad. Sci.79,2554-2558,1982.

[18] Berlin, T., Kac,M.,"The Spherical Model of a Ferromagnet", Physical Review, Vol. 86, No. 6, 1952.

[19] Stanley, H.," Spherical Model as the Limit of Infinite Spin Dimensionality", Physical Review, Vol. 176, No. 2, 1968.

[20] Joyce, G., "Critical Properties of the Spherical Model", Chap. 10, Domb, C., Green, M., "PhaseTransitions and Critical Phenomena", Vol. 2., Academic, New York, 1972.

[21] Montroll, E., Weiss, G., "Random Walks on Lattices. II", J. Math. Phys., Vol.6, No. 2, Feb. 1965.

[22] Schlesinger, M., Klafter, J., "Lévy Walks Versus Lévy Flights", in "On Growth and Form",Stanley, H., Ostrowsky, N. eds., Martinus Nijhoff, 1986.

SOME TECHNIQUES FOR PRIORITIZED PREEMPTIVE SCHEDULING

Lui Sha[1], John P. Lehoczky[2], Ragunathan Rajkumar[3]

[1]Department of Computer Science
[2]Department of Statistics
[3]Department of Electrical and Computer Engineering
Carnegie-Mellon University

Abstract

Most existing real-time control systems use *ad hoc* static priority scheduling methods. This is in spite of the fact that the rate monotonic scheduling algorithm was proved to be the optimal static priority scheduling algorithm for periodic tasks over a decade ago. This lack of use is, in part, because a direct application of this algorithm leads to a number of practical problems which have not been fully addressed in the literature. In this paper, we give a comprehensive treatment of a number of practical problems associated with the use of the rate monotonic algorithm. We review the methods to handle aperiodic tasks and present a new approach to stabilize the rate monotonic algorithm in the presence of transient processor overloads. Finally, we investigate the degradation effects of cycle-stealing in real-time systems. Effective solutions are proposed to minimize these effects. New results also clearly establish the importance of using an integrated approach to schedule both the processor and data I/O activities.

1. Introduction

Scheduling hard real-time tasks is an important topic in real-time systems. Both preemptive scheduling algorithms[1, 2, 3, 4, 5, 6] and non-preemptive scheduling algorithms[7, 8, 9] are active areas of research. In the context of a uni-processor preemptive scheduling environment, the rate monotonic and earliest deadline scheduling algorithms were proven to be optimal static priority and dynamic priority scheduling algorithms respectively more than a decade ago[1]. It was also proven that the least slack time algorithm is optimal[5].

In spite of the optimality property, neither of these algorithms is widely used in practice. This is, in part, because a direct application of these algorithms leads to many practical problems which have not been fully addressed in the literature. Among them, the problem of stochastic task execution times and the associated problem of scheduling stability are of particular importance. In many applications, the task execution times are stochastic, and the worst case execution time is much larger than the average execution time. To ensure that the processor never becomes overloaded would lead to a very low degree of average processor utilization. In other words, in order to have a reasonable average processor utilization, scheduling algorithms must be able to handle occasional transient overloads. We consider a scheduling algorithm to be stable if it can guarantee the deadlines of a set of critical tasks even if the processor is overloaded, as long as this set of critical tasks, by itself, can be scheduled by this algorithm under worst case conditions. Unfortunately, when a set of tasks is scheduled by the earliest deadline algorithm (or the least slack time algorithm), deadlines will be missed in an unpredictable fashion, should the processor experience a transient overload.[1] This is especially ironic, because the time at which a processor overload develops is usually the moment at which the physical system under control experiences some great difficulty. In the case of the rate monotonic algorithm, the tasks with longer periods miss their deadlines when a processor experiences a transient overload. However, a task with a longer period can actually be the most important task to the mission. Therefore, the solution to the stability problem is crucial in the application of the rate monotonic scheduling algorithm to real-time control systems.

Another important practical problem which has received little attention in the literature is the integration of processor and data I/O scheduling. Data I/O using DMA results in the problem of "cycle-stealing" where I/O activities "steal" cycles from the processor causing delays. These delays result in tasks being slowed down and deadlines can be missed. Section 3 presents results that demonstrate that any type of cycle-stealing policy on a single system bus to memory leads to performance degradation. We also present a dynamic arbitration scheme that is consistent with the scheduling algorithm used for each of the subtasks. Memory interleaving is required to eliminate the loss of cycles and studies indicate the appropriate number of memory banks to be used. These results also demonstrate that the common practice of assigning fixed priorities to bus interface

This work was sponsored in part by the Office of Naval Research under contract N00014-84-K-0734, in part by Rome Air Development Center under contract F30602-85-C-0274 and in part by the Federal Systems Division of IBM Corporation under University Agreement YA-278067, which support the Archons decentralized computer system project at Carnegie-Mellon University, E. Douglas Jensen, Director.

[1] For a treatment of the stability problem using dynamic priority scheduling method, readers are referred to[10].

units for performing DMA, which approximates FIFO, is inconsistent with the rate monotonic or other well known real-time processor scheduling algorithms.

In this paper, we give a comprehensive treatment of some commonly encountered problems in the context of using the rate monotonic algorithm for prioritized preemptive scheduling. We show that the rate monotonic algorithm is not only very easy to implement but also very versatile. It can be easily made stable in the presence of stochastic execution times and transient processor overloads. In Section 2, we address the problem of stochastic task execution times and stability in scheduling. In Section 3, we address the problems associated with integrated processor and data I/O scheduling. Finally, Section 4 presents the conclusion.

2. Stochastic Task Execution Times and Scheduling Stability

2.1. Background

In many applications, the task execution times are often stochastic, and the worst case execution time can be significantly larger than the average execution time. For example, the computation time of a periodic process which monitors and controls the temperature of some device can be small if the temperature is normal. However, an abnormally high temperature reading can trigger an additional burst of computation. When we create a periodic task to serve a class of aperiodic tasks as described above, the resulting periodic task also has a stochastic execution time. When the processor load is stochastic, ensuring that the processor never becomes overloaded can lead to a very low average processor utilization.

In the context of managing task stochastic execution times, we consider a scheduling algorithm to be stable if any given set of critical tasks is guaranteed to meet all their deadlines even if the processor is overloaded and it is impossible to schedule all the tasks. Of course, the set of critical tasks, by itself, must be schedulable under worst case conditions. Obviously, it is desirable to have scheduling algorithms which are not only optimal but also stable. This will provide the highest degree of average processor utilization and guarantee meeting the deadlines of all the critical tasks even if a transient overload develops. However, a direct application of either the rate monotonic or the earliest deadline algorithm results only in optimal but unstable scheduling, because neither algorithm addresses the problem of transient overloads. If the rate monotonic algorithm is used and a transient overload develops, tasks with longer periods will miss their deadlines. This may not be desirable, because a task with a longer period could be most critical to the mission. If the earliest deadline or the least slack time algorithm is used and a transient overload develops, then it is impossible to predict a priori which task's deadline will be missed. As a result, many existing real-time systems provide a priority task dispatcher and let application programmers solve the stability problem on a case by case basis.

2.2. Existing Methods for the Stability Problem

A common approach to solving the stability problem is to first assign scheduling priorities to tasks according to their importance for the mission. A fixed priority scheduler developed in this way is stable in the sense that the execution of a task will not be compromised by less important ones. However, the processor utilization can be poor, because an important task can have a relatively long deadline. When it is assigned a high priority, less important tasks with shorter deadlines can be forced to miss their deadlines unnecessarily.

For example, suppose that tasks $\mathbf{T}_1$ and $\mathbf{T}_2$ are periodic with periods 100 and 10 respectively. Both of them are initiated at $t = 0$, and task $\mathbf{T}_1$ is more important than task $\mathbf{T}_2$. Assume that task $\mathbf{T}_1$ requires 10 units of execution time and its first deadline is at $t = 100$, while task $\mathbf{T}_2$ needs 1 unit of execution time with its first deadline at $t = 10$. In this case, a simple scheduling algorithm such as the rate monotonic algorithm will successfully schedule both tasks. However, if task $\mathbf{T}_1$ is assigned higher scheduling priority, task $\mathbf{T}_2$ will miss its deadline unnecessarily even though the total processor utilization is only 0.2. Of course, the situation would be far better if task $\mathbf{T}_2$ were more important than task $\mathbf{T}_1$. In this case, assigning tasks according to their mission importance results in the optimal static priority schedule: a rate monotonic schedule. In practice, situations are often somewhere in between the worst and the best cases. Typically, after assigning tasks priorities according to their relative importance, the processor load is tested. If either the deadlines of some critical tasks cannot be ensured or the processor utilization is too low, iterations of task priority adjustment and code optimization are performed until both the timing requirements of critical tasks are ensured and a reasonable average processor utilization is achieved.

Another common approach used to deal with the stability problem is to create a set of time division multiplex (TDM) slots and then hand-pack all the important tasks into them. This is typically done in the context of a cyclical executive, which usually uses a few frequencies[11]. The fastest cycle is usually called the major cycle and the slower ones are called minor cycles. The major cycle is assigned the highest priority. Given the highest priority, a major cycle with period P will be regularly given 1 slot every P units of time. This, in effect, creates a virtual processor with processing bandwidth 1/P. The period of the major cycle is determined by two factors. First, period P must be short enough so that it can accommodate the high frequency periodic tasks. Second, the major cycle must also accommodate tasks which have lower frequencies but are critical to the mission, since the major cycle has the highest priority. A handcrafted table is then constructed to schedule both the high frequency tasks and the critical tasks over the virtual processor. The construction of the scheduling table often takes many iterations over the adjustment of the period of the major cycle, the modification of the scheduling table and the optimization of the code of certain tasks.

2.3. The Period Transformation Method

The common problem with the two trial and error approaches described in Section 3.2 is that the scheduling is not performed according to a clearly specified algorithm. As a result, the timing behavior of the resulting schedule is often very difficult to understand and modify. A simple method to ensure a high degree of schedulability and to ensure the stability during transient loads is to perform a period transformation. We illustrate this idea by the following example. Suppose that we have two tasks, $T_1 : (P_1 = 12, C_1 = 4, C_1^+ = 7)$ and $T_2 : (P_2 = 22, C_2 = 10, C_2^+ = 14)$, where P_i, C_i and C_i^+ are task i's period, the average case and the worst case computation times respectively. The average case and worst case utilizations of these two tasks are 0.79 and 1.2 respectively. We also assume that both of these two tasks are initiated at time $t = 0$. Since the Liu and Layland bound for two tasks is $2(2^{1/2}-1) = 0.82$, these two tasks are schedulable by the rate monotonic in the average case but cannot be scheduled in the worst case by any algorithm. Now assume that task T_2 is more important than task T_1 and that we want to guarantee the deadline of task T_2 even in the worst case condition. It is tempting to elevate the priority of task T_2 directly. Unfortunately, this will force task T_1 to miss its deadline even in the average case. If the priority of task T_2 is elevated higher than that of task T_1, then task T_2 takes the first 10 time slots. Task T_1 takes the next 4 time slots and consequently task T_1 misses its deadline at $t = 12$. However, this problem can be solved by transforming task T_2 to $T_2^* : (P_2 = 11, C_2 = 5, C_2^+ = 7)$. The transformed task T_2^* has a period 11 with at most 7 units of computation time allowed in each period. Since tasks T_1 and T_2^* are scheduled using the rate monotonic scheduling algorithm and their total average utilization is 0.82, which is less than the Liu and Layland bound for 2 tasks, they are schedulable in the average case. In addition, in the worst case, task T_2^* will still meet its deadline, because its priority is higher than task T_1.

Instead of transforming task T_2's period to a shorter one, it is possible to lengthen the period of task T_1, provided that the application in question permits the postponement of T_1's deadlines. For example, we can decompose task T_1 into two tasks $(P_{1,1} = 24, C_{1,1} = 4, C_{1,1}^+ = 7)$ and $(P_{1,2} = 24, C_{1,2} = 4, C_{1,2}^+ = 7)$. In addition, these two new tasks must be separated by a phase of 12. That is, task $T_{1,1}$ initiates at times, 0, 24, 48, ... etc, while task $T_{1,2}$ initiates at times 12, 36, 60, ... etc. Note that in this arrangement, a job which arrives at time t has its deadline at $(t + 24)$ rather than $(t + 12)$. This results in a deadline postponement of a single period. From an implementation point of view, this is equivalent to scheduling task T_1 at the priority level associated with period 24 rather than that associated with period 12, and to postponing the deadline of each job in task T_i to the end of the next period rather than the end of the current period. From the discussion in Section 2, we know that deadline postponement also has the additional advantage of improving the scheduling bound, if it is permitted by the application in question. In summary, there are two ways to transform a period. First, we can shorten a period P_i to P_i/k and restrict the maximal computation time for each period to be C_i^+/k. Second, we can "lengthen" the period of a task P_i to kP_i, if permitted. This corresponds to scheduling this task at the priority level associated with kP_i. In addition, the deadline of a job of task T_i, which arrives at time t, is postponed to $(t + kP_i)$ from $(t + P_i)$.

A systematic procedure to perform period transformation is as follows. First, we identify the set of critical tasks, which are schedulable by the rate monotonic algorithm under the worst case condition. From an implementation point of view, the specification of the critical task set can be made quite flexible. First, we can refine the definition of mission phase and define the associated set of critical tasks, so that for any given mission phase, we have a small and well defined set of critical tasks. If the critical task set of a particular mission phase becomes too large, we can consider specifying one part of a task as critical. For example, suppose that in a tracking application the position information of a ship is calculated periodically. It may be acceptable to miss the deadline of a single period but not two consecutive ones. In this case, one can decompose the computation into two tasks. For example, suppose that the original task has a period of 5 units. We can create two tasks with period 10. One starts at time $t = 0$, and the other starts at time $t = 5$. Of the two tasks, only one of them is designated as a critical task.

Second, given that the task set is partitioned into a critical set and a non-critical set, we order both the critical set of tasks and the non-critical set of tasks by the rate monotonic scheduling algorithm. Next, we denote the longest period of the critical set to be $P_{c,max}$ and the shortest period in the non-critical set to be $P_{n,min}$. If $P_{c,max} < P_{n,min}$, then no period transformation is needed. If $P_{c,max} = P_{n,min}$, then we break the tie by assigning $P_{c,max}$ a higher priority, and there is again no need for period transformation.

Third, suppose that $P_{c,max} > P_{n,min}$. We now attempt to include the task corresponding to $P_{n,min}$, $T_{n,min}$, into the critical set. If the worst case total utilization of task $T_{n,min}$ and of the critical tasks is less than the Liu and Layland bound, $n(2^{1/n}-1)$, then we take task $T_{n,min}$ out from the non-critical task set and put it into the critical set.[2] We repeat this procedure until one of the following two conditions is true. First, suppose that in the current partition, $P_{c,max} \leq P_{n,min}$. In this case, no further period transformation is needed.

[2] Alternatively, we can simulate the rate monotonic scheduling under the worst case condition. This is more time consuming but has the potential to permit a larger critical task set, because the Liu and Layland bound is a worst case calculation for any task set of size n, not for a particular task set.

For example, suppose that we have three tasks, T_1, T_2, T_3 with $P_1 < P_2 < P_3$ and that task T_2 is the critical one. In this case, we have $(P_{c,max} = P_2) > (P_{n,min} = P_1)$. Suppose that task T_1 and T_2 can be scheduled by the rate monotonic algorithm. We include task T_1 in the critical set, and now the critical set becomes $\{T_1, T_2\}$. Furthermore, in the current partition we have $(P_{c,max} = P_2) < (P_{n,min} = P_3)$. Thus, no additional period transformation is needed. Of course, it is possible that $P_{c,max}$ remains longer than the current $P_{n,\ min}$ and that $T_{n,min}$ cannot be included in the critical set. In this case, we go to the next step.

Fourth, given that $P_{c,max} > P_{n,min}$ and task $T_{n,min}$ cannot be included in the critical set, we now try to lengthen the periods of non-critical tasks whose deadlines are shorter than $P_{c,max}$, because deadline postponement, if permitted, increases the scheduling potential. We lengthen the period of $P_{n,min}$, if permitted, until it becomes longer than that of $P_{c,max}$. We then take the new $P_{n,min}$ and repeat this procedure until one of the following two conditions is true. First, suppose that in the current partition, we have $P_{c,max} \leq P_{n,min}$. In this case, no more period transformation is needed. However, it is possible that the current $P_{n,min}$ is not permitted to be lengthened to the extent that it becomes longer than $P_{c,max}$. In this case, we go to the next step.

Finally, given that we cannot transform $P_{n,min}$ to the point that $P_{c,max} < P_{n,min}$, we lengthen $P_{n,min}$, if permitted, to the extent that a new longer $P_{n,min}$ is obtained. For example, suppose the current $P_{n,min}$ is 100 and the next shortest period in the non-critical task set is 150. We lengthen $P_{n,m}$ to 200 and now the new $P_{n,min}$ is 150. We repeat this procedure until the longest possible $P_{n,min}$ is obtained. At this point we shorten the length of each of the critical tasks whose period is longer than $P_{n,min}$ until its transformed period is shorter than or equal to $P_{n,min}$.

In summary, to ensure a high degree of average processor utilization and to guarantee meeting the deadlines of all the critical tasks, we first define a set of critical tasks which, by themselves, can be scheduled by the rate monotonic algorithm even in the worst case situation. Next, we use the period transformation method to ensure that the longest period in the critical set is less than or equal to the shortest period in the non-critical set. Finally, we assign priorities according to the rate monotonic algorithm. If the periods of a critical and of a non-critical task are equal, we break the tie by assigning higher priority to the critical task. This ensures that the deadlines of all the tasks can be met as long as the total work load is schedulable by the rate monotonic algorithm and that the deadlines of the critical tasks will always be met even if overload develops. Having addressed the issue of task stochastic execution times and the associated problem of stability, we now turn to the problem of integrated processor and data I/O scheduling in the next section.

3. Integrated Processor and I/O Scheduling

3.1. Background

In contrast to the rather well developed theory of scheduling periodic tasks in a processor or on a bus, little work has appeared on the important problem of integrating the scheduling of tasks in processors with their data I/O. In fact, many practitioners often use FIFO to schedule data I/O to and from bus interface units or other I/O devices, independent of the method used in scheduling the processor. In addition, when tasks have processing as well as I/O requirements, the memory subsystem is of paramount importance. If all memory accesses take place over a single bus, the processor would be delayed when I/O is in progress and vice-versa. A common scheme for single bus configurations is to allow DMA devices to "steal" cycles from the processor.

In this section, we demonstrate the degrading effects of cycle-stealing in real-time systems and study solutions to the problem of minimizing these effects. In these studies, we assume a simple task model where each task is periodic and has three subtasks executed in sequence namely input DMA, processing and output DMA. We also assume that the deadline for each job is always at the end of its period. It is also assumed that there is a single DMA controller. Under these conditions, the processor can be scheduled independently and DMA subtasks are scheduled together using the same scheduling algorithm. However, if a DMA subtask is currently executing, a processor subtask is delayed and vice-versa. We model the delays encountered by these tasks using the Skinner-Asher model[12]. This model is based on the probability that a processor or device requests access to a particular memory module and the probability that it gains immediate access in the case of contention. A redefinition of the probabilities can simulate multiple memory schemes and contention resolution policies.

3.2. Algorithm Selection

We use the rate-monotonic scheduling algorithm to demonstrate the effects of cycle-stealing. Since the scheduling takes place at two stages, namely in the processor and in the DMA controller, we refer to this scheduling policy as RR. In addition, since contention takes place between DMA and processor subtasks, we need to adopt a resolution policy to resolve this contention. Such a conflict resolution policy is usually provided in hardware and allows only one access to be made during a single memory cycle. This resolution policy can range from higher priority to one or the other device or fair resolution (FIFO). We represent this policy as the ratio of requests serviced in favor of DMA accesses to that of processor accesses. The policies, namely DMA : CPU accesses, that we use in our studies are: 0 : 100, 25 : 75, 50 : 50, 75 : 25 and 100 : 0. The policy 0 : 100 represents "CPU-burst" mode, 50 : 50 represents "fair" resolution and 100 : 0 represents "DMA-burst" mode with the other two representing variations inbetween.

In these studies, we present the concept of dynamic bus arbitration for memory accesses. In existing systems, the hardware support for conflict resolution is such that priorities are attributed to memory requests depending upon the physical source of the requests. For example, I/O devices are usually accorded higher priority than the processor. This represents a fixed-priority scheme for scheduling the bus and has been the traditional vehicle of bus access implementation schemes (for instance, daisy-chaining). However, in dynamic bus arbitration, the priority at which accesses are made is based only upon the importance of the task being executed and the physical source of the requests is immaterial. Thus, if RR scheduling is used, the task with the shorter period will have higher priority irrespective of whether it is executing its DMA subtask or its processing subtask. In such a situation, all subtasks of a task are scheduled in an integrated fashion using the rate-monotonic scheduling algorithm. On the contrary, the resolution policies mentioned above lead to deviations from rate-monotonic scheduling. For instance, if the fair 50 : 50 resolution policy is used, a task with a longer period can steal cycles from a task with a shorter period and hence with higher priority. This represents some form of priority inversion and is not strictly in accordance with the scheduling approach decided by the rate-monotonic algorithm. Similar cases arise for other fixed-priority resolution schemes as well.

Another parameter that directly affects memory contention is the amount of processor memory accesses. If the processor is executing a tight loop within a cache, or executing multiply and floating point operations, few memory requests will be generated by the processor. We scan the entire spectrum of possible memory access ratios in our studies and choose uniform processor access ratios of 0%, 25%, 50%, 75% and 100%. We assume that DMA is buffered and hence I/O subtasks request memory access every cycle when executing.

This study focused on finding the breakdown utilization points as follows. We begin with a task set $\{(C_1, P_1), \ldots (C_n, P_n)\}$ and systematically scale up the computation requirement of each task by increasing the value of a weight, w, so that the resulting task set $\{(wC_1, P_1), \ldots (wC_n, P_n)\}$ just has a task missing its deadline. The resulting utilization level is the breakdown point for the given task set. However, the importance of I/O scheduling depends in part on the amount of I/O associated with the tasks. If tasks are "CPU bound" and have relatively small amounts of I/O, then I/O scheduling will be relatively unimportant. Similarly, if tasks are "I/O bound", then I/O scheduling will be of much greater importance. It is, therefore, useful to introduce the notions of "I/O bound" and "CPU bound" into the task set generation process. We accomplish this by specifying a doublet of relative utilizations for each of the CPU and DMA phases; for example, (1, 5).

3.3. Results

In our first study, we made runs for 5 task sets: (1, 5), (3, 5), (5, 5), (7, 5) and (9, 5). Different resolution policies and processor memory access ratios were used for each task set and the the performance was averaged over many task sets.

The results of this study are shown in Fig. 3-4. For example, in Figure 3-4.a, the task set is compute bound with the DMA : CPU requirements ratio being 1 : 5. The dashed line which indicates the CPU burst mode does better than dynamic bus arbitration (represented by triangles) with the latter performing at least as well as any other resolution policy. Again in Figure 3-4.e, where the task set is I/O bound, the DMA burst mode (represented by squares) performs better than other resolution policies by as much as 5% with dynamic bus arbitration performing as well as any other resolution policy. In Figures 3-4.b to 3-4.d, the situation is somewhere inbetween, with the resolution policy that favors the dominating phase (subtask) performing well until the CPU memory access ratio tends towards 1. Also for any resolution policy, as the CPU memory access ratio increases, potential for memory conflict increases with resultant delays and decrease in the bottleneck utilization thresholds. For example, in Figure 3-4.c when dynamic bus arbitration is used, the utilization threshold drops from 93% to 60% as the CPU memory access ratio increases from 0% to 100%.

The graph indicates that though other resolution policies can do well for some CPU memory access ratios, only the bus arbitration policy performs consistently well for any ratio. We also saw that for any given type of task set, the arbitration policy while not offering the best performance under a given set of conditions provides a reasonably good performance that is surpassed for the most part only by the policy that favors the dominating phase. However, a policy that favors the dominating phase cannot be preferred due to the inherent instability that it provides to scheduling. It would mean that the resolution policy would have to be reset in hardware every time a new type of task set is to be run and would be highly undesirable. For example, in Fig. 3-4.e for a CPU memory access ratio of 0.5, the DMA-burst mode does better than dynamic bus arbitration by about 8% for the task set is I/O bound while for an evenly balanced task set as in Figure 3-4.c, it performs worse by 5%.

Another significant result from this study is that a resolution policy that favors the dominating phase can, indeed, do better (not necessarily always) than dynamic bus arbitration which represents the rate-monotonic algorithm. Hence, the rate-monotonic algorithm is not optimal with multiple subtasks when the subtasks interfere. This is due to the fact that a resolution policy favoring the dominating phase attempts to maximize the utilization of this bottleneck. We refer to this as its "optimistic" scheduling approach: it tries to keep the bottleneck phase always busy and in the process, deadlines are met. In other words, little laxity can be "stolen" from the dominating phase and hence is assigned higher priority. However, this approach does not always work due to its inability to incorporate the significance of deadlines (as seen by the rate-monotonic algorithm) in its prioritizing policies. By scheduling the dominating phase, it pays no regard to impending deadlines and can miss them. Hence, generally, for lower CPU memory access ratios, the dominating phase policy performs better than all other policies but as the ratio tends towards the worst-case, the performance of this policy becomes worse than other policies. This is particularly evident in the I/O bound task set of Figure 3-4.d where at a CPU access ratio of 1.0, the dominating phase policy, in fact, fares worse than dynamic bus arbitration. It can be shown that the same situation is true for deadline and least slack-time scheduling algorithms as well.

The above study shows that as the CPU memory access ratio increases and the DMA : CPU ratio becomes more even (see Fig. 3-4.c), the utilization threshold can drop to levels below 40%. The primary cause of this effect is that the bus gets saturated and servicing of memory requests is delayed with a resultant penalty in system throughput. Hence, any attempts to alleviate the memory bottleneck should aim at maximizing the number of memory requests that can be serviced during a memory cycle. Memory interleaving is an effective solution towards this goal and allows multiple memory banks to be accessed simultaneously. Ideally, the number of memory banks should be large enough so that there is negligible delay in servicing requests from multiple sources. However, additional memory banks incur extra cost and the variation of performance with the number of banks is the focus of our next study.

In this study, we use RR scheduling as before but we use the DMA-burst resolution policy which reflects several existing schemes where I/O devices are assigned higher priority than the processor. However, in order to minimize the contention of memory accesses, memory banks are interleaved such that requests to distinct modules can be serviced simultaneously. We made runs for all types of task sets. For each task set, RR scheduling was used with DMA accesses always having higher priority while the number of memory banks was varied from 1 to 32 and the CPU memory access ratio was varied as before. In this study, we assume that all memory banks are accessed with uniform probability which assumes lo-bit interleaving ([13]). The average performance for all these task sets is given in Fig. 3-5.

Fig. 3-5 shows that as the number of memory banks increases, the uitlization threshold rises exponentially to that obtainable under ideal conditions as assumed in the studies presented in the previous section. In the extreme case when the processor memory access ratio is nearly zero, no contentions take place and additional banks serve no purpose. For the worst-case CPU memory access ratio of 1.0, ie. the processor requests a memory access every cycle, the average performance shows that while 8 banks are needed to obtain near-maximal performance, 4 banks yield only about 3% less than with 8 banks on the average and would be a good choice for the number of banks both in terms of utilization and economy. The inclusion of 4 banks can lead to a 30% performance enhancement and the the inclusion of a single additional bank can offer 25% enhancement relative to a single bank!

3.4. Significance of I/O Scheduling

Many practitioners often use FIFO to schedule data I/O to and from bus interface units or other devices, independent of the method used in scheduling the processor. We now conduct a simulation study to compare the performance of the rate monotonic algorithm with that of other methods. We now assume that loss of cycles to DMA devices is made negligible by the provision of interleaved memory banks. The general practice of assigning channel processor priorities according to the average speed of the devices it is connected to approximates a FIFO algorithm. The deadline scheduling algorithm is an optimal dynamic priority scheduling algorithm. Hence, we choose the following set of algorithms in our study: the dual stage FIFO (denoted as FF), FIFO for DMA and rate-monotonic for the processor (FR), the dual stage rate-monotonic (RR), and the dual stage deadline scheduling (DD). Finally, we use at both stages a refined version of the earliest deadline scheduling algorithm called the ***propagated deadline*** scheduling algorithm which is based upon the latest starting time for each phase (denoted as PP). We repeat the above experiment with a single deadline postponement where the deadline of the current job is the end of its next period instead of its current period. The filled portion of each bar in the graphs indicates the utilization without any deadline postponement and the empty portion indicates the breakdwon utilization with a single deadline postponement.

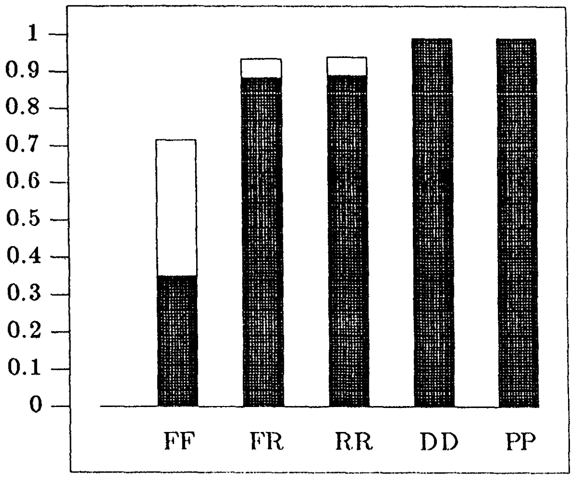


Figure 3-1: Breakdown Utilization for 10 tasks
(i) Total DMA : Computation = 1 5

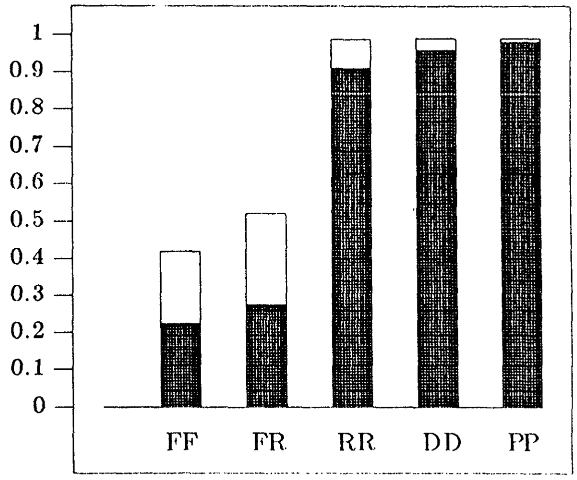


Figure 3-2: Breakdown Utilization for 10 tasks
(i) Total DMA : Computation = 5 5

The results for three types of task sets are presented in Figs. 3-1 through 3-3. The figures are typical of the behavior observed. FIFO scheduling for I/O is highly erratic. It is possible that such an algorithm could achieve a decent breakdown utilization (equivalent to RRR or RR) for one phasing but for another to miss deadlines at a very low utilization. Clearly I/O must be scheduled just as the CPU if its utilization is at all significant. The important observation is that a simple algorithm such as RRR or RR achieves an excellent breakdown utilization.

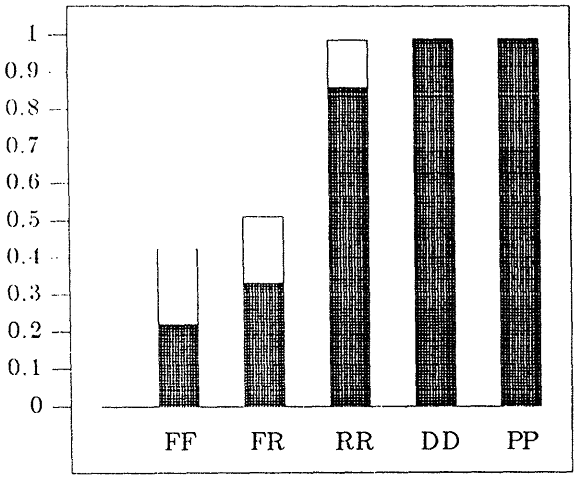


Figure 3-3: Breakdown Utilization for 10 tasks
(i) Total DMA : Computation = 9 5

To sharpen our understanding of the relative behavior of these algorithms, we performed another simulation experiment. This consisted of generating ten sets of ten tasks using (7,5) relative utilizations. Each of the ten task sets was then scheduled by the five algorithms under consideration. We present the results in Table 1.

CASE	FF	FR	RR	DD	PP
1	.301	.387	.851	.979	.995
2	.231	.256	.920	.977	.992
3	.218	.290	.968	.989	.996
4	.388	.441	.828	.981	.997
5	.223	.353	.861	.972	.998
6	.582	.748	807	.978	.996
7	.429	.467	.849	.986	.992
8	.121	.164	.912	.967	.991
9	.114	.114	.898	.978	.985
10	.247	.387	.861	.950	.975

Table 1: (7, 5) Relative Utilization for Interfering DMA Subtasks

The table clearly illustrate the drawbacks from using an algorithm such as FIFO and the benefits of using the rate monotonic algorithm. One can observe that the breakdown utilizations for FFF and FRF (FF and FR) have a wide variance. This is in keeping with our experience. It is possible for FFF and FRF (FF and FR) to do adequately (rarely) and miserably (more commonly). These studies clearly demonstrate that the common practice of using the FIFO discipline in I/O device scheduling is very undesirable and suggest that there is a serious need for the redesign of I/O controller architectures. Another important observation is that the rate monotonic algorithm used at each of the scheduling stages on average ensures that all deadlines will be met even if the bottleneck utilization is well above 80%.

4. Conclusion

In this paper, we have investigated the problem of scheduling stability and of integrated I/O scheduling in the context of using the rate monotonic algorithm. The rate-monotonic algorithm can easily be made stable in the presence of stochastic task execution times and transient processor overload by the period transformation algorithm developed in this paper. Another significant result from this study is that in the presence of cycle-stealing, there is *no* single contention resolution policy that performs consistently well over all types of task sets. We have also presented the concept of a dynamically arbitrating bus which partially ameliorates this situation by yielding consistently good performance, though not necessarily the best under a given set of conditions. The performance degradation effects of cycle-stealing can be eliminated by interleaving multiple memory banks and good cost/performance tradeoffs can be obtained for a particular system by appropriately choosing the number of memory banks. Moreover, we demonstrate that forcing data I/O to be scheduled in a FIFO fashion often leads to very poor scheduling results unless the tasks are strongly CPU bounded. Since the hardware modification needed to support rate monotonic scheduling is straightforward, we suggest that real-time system designers seriously consider this issue, especially in the context of a distributed system where the bus interface unit of each computer must serve data associated with tasks that have different periods and deadlines.

References

1. Liu, C. L. and Layland J. W., "Scheduling Algorithms for Multiprogramming in a Hard Real Time Environment", *JACM*, Vol. 20 (1)1973, pp. 46 - 61.

2. Mok A. K. and Dertouzos, M. L., "Multiprocessor Scheduling in a Hard Real-Time Environment", *Proceeding, Seventh Texas Conference on Computer Systems,*Nov. 1978.

3. Leung, J. Y. and Merrill M. L., "A Note on Preemptive Scheduling of Periodic, Real Time Tasks", *Information Processing Letters*, Vol. 11 (3)Nov. 1980, pp. 115 - 118.

4. Lawler, E. L., "Scheduling Periodically Occurring Tasks on Multiprocessors", *Information Processing Letters*, Vol. 12 (1)February, 1981, pp. 9 - 12.

5. Mok, A. K., *Fundamental Design Problems of*

Distributed Systems For The Hard Real Time Environment, PhD dissertation, M.I.T., 1983.

6 Ramamrithan K. and Stankovic J. A., "Dynamic Task Scheduling in Hard Real-Time Distributed Systems", *IEEE Computer*, July 1984.

7. Leinbaugh, D. W., "Guaranteed Response Time in a Hard Real-Time Environment", *IEEE Transaction on Software Engineering*,Jan. 1980.

8. Leinbaugh, D. W. and Yamini M., "Guaranteed Response Time in a Distributed Hard Real-Time Environment", *Proceeding, Real-Time Systems Symposium*,Dec. 1982.

9. Zhao, W., Ramamritham, K., and Stankovic, J. A., "Scheduling Tasks with Resource Requirements in Hard Real-Time Systems", *IEEE Transaction on Software Engineering*,April 1985.

10 Sha, L. and Lehoczky, J. P., "Stablized Deadline Scheduling Algorithms --- Scheduling Hard Real-Time Tasks with Stochastic Execution Times", Tech. report, Department of Computer Science, Carnegie-Mellon University, 1986.

11. Carlow, G. D., "Architecture of the Space Shuttle Primary Avionics Software System", *Communications of the ACM*,September 1984.

12. Skinner,C.E., and Asher,J.R., "Effects of Storage Contention on System Performance", *IBM Systems Journal*, Vol. 41969, pp. 319-333.

13. Gorsline,G.W., *Computer Organization: Hardware and Software*, Addison-Wesley, Reading, MA, 1984, ch. 4.

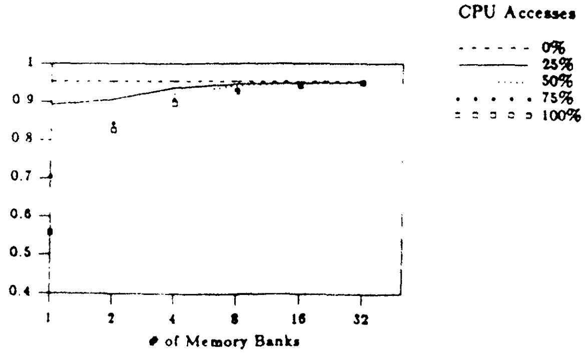


Figure 3-5: Average Breakdown Utilization for RR for all types of task sets. DMA-burst policy is used.

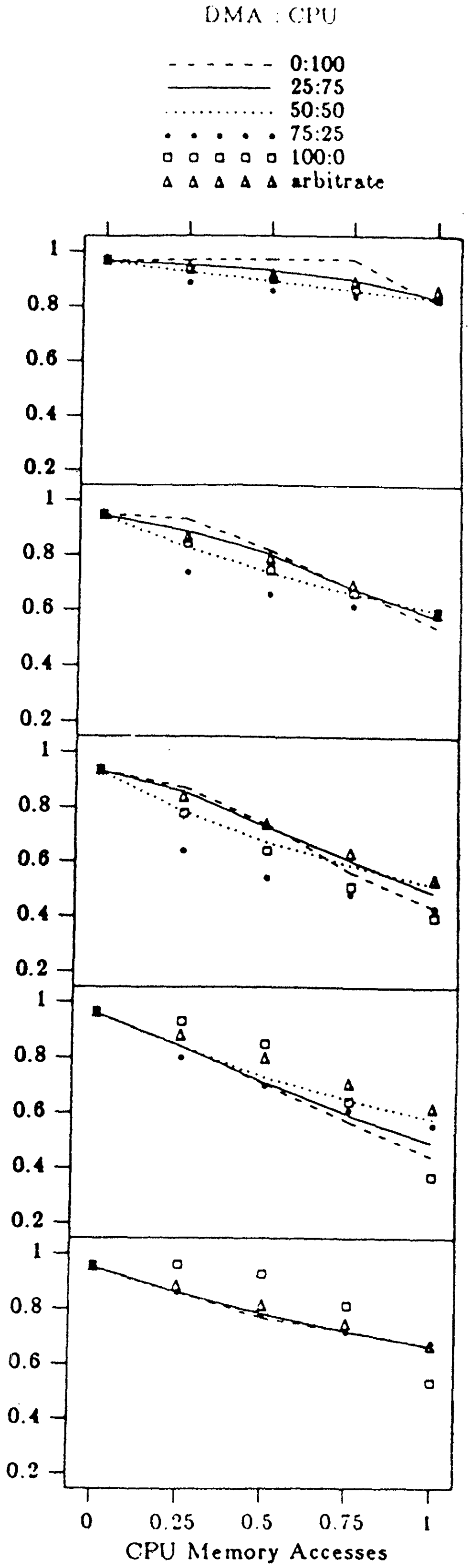


Figure 3-4: Breakdown Utilization for RR with one memory bank. DMA : CPU ratios are (a) 1 : 5 (b) 3 : 5 (c) 5 : 5 (d) 7 : 5 (e) 9 : 5

A PERSPECTIVE ON MOE'S FOR WIDE AREA SURVEILLANCE

Leonard O. Sweet

Naval Research Laboratory
4555 Overlook Avenue, S.W.
Washington, DC 20375-5000

1. Introduction

A distributed wide area surveillance (WAS) system as shown in Fig. 1 consists of sensors, processing/fusion nodes, and communication links whose function is to obtain a WAS product and deliver it to the user with sufficient speed and accuracy so that the product can improve operational decisions. There is a need to be able to make informed decisions on how to specify system performance, what type of systems to buy, and how many to buy. In other words, a process is required that can quantitatively support a system acquisition strategy.

The required process demands a method for evaluating the contribution of the various systems to the overall WAS performance, which requires that meaningful measures of effectiveness or MOE's be developed. All too often, system (sensors, correlator/trackers) MOE's are considered independently without regard to their interaction or their contribution to overall operational effectiveness. What is often not appreciated, is the importance and difficulty in agreeing on which MOE's to use and how to use them commensurately so that the MOE's will ultimately couple into an evaluation of the Navy's capability to fight since that is the fundamental reason for providing the WAS product.

It is desirable to define a hierarchy of MOE's as is illustrated in Fig. 2 and recognize that different models or methodologies will be used to obtain MOE's for each of the three levels (sensing, fusion, and operations). Although each level can be evaluated separatedly against a set of MOE's and useful insights and information obtained as interim steps, it is only by modeling the end to end process that acceptable answers that will support a system acquisition strategy can be obtained.

This paper will define candidate MOE's for each level and discuss their relationship to overall WAS effectiveness. We will take a top down approach starting with operational MOE's followed by fusion MOE's and finally sensor MOE's. This is a logical sequence since lower level MOE's need only be considered if they have a significant impact on a higher level MOE.

2. Operational MOE's

The operational MOE's associated with wide area surveillance can be grouped into two general categories, those impacting fleet defense and those supporting our offensive operations. In either case, the MOE's are expressed in terms of changes in operationally related outcomes as a result of the WAS product.

The evaluation of the dependence of operational outcome to the quality of the WAS product requires the use of an engagement simulation or model. Either knowledgeable men-in-the-loop must be used to make operational decisions based on the WAS product and a given scenario, or their decisions must be capable of being emulated within the simulation.

Candidate MOE's follow.

Fleet Defense

a. Aircraft Destroyed. The change in the number of attacking bombers destroyed. It is obtained as the outcome of the outer air battle (OAB).

b. Missiles Reaching the Inner Defense Zone (IDZ). The change in the number of missiles reaching the IDZ.

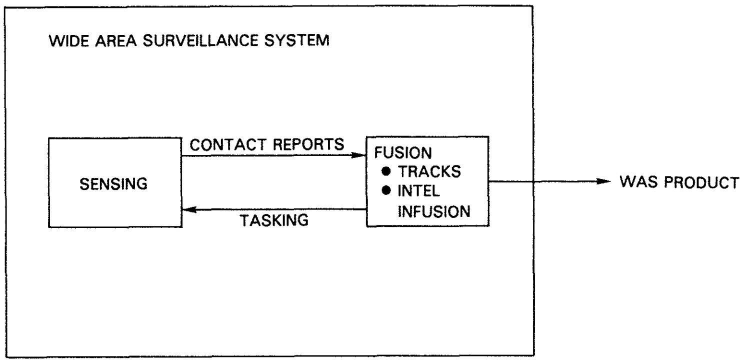


Fig. 1 — Wide area surveillance system

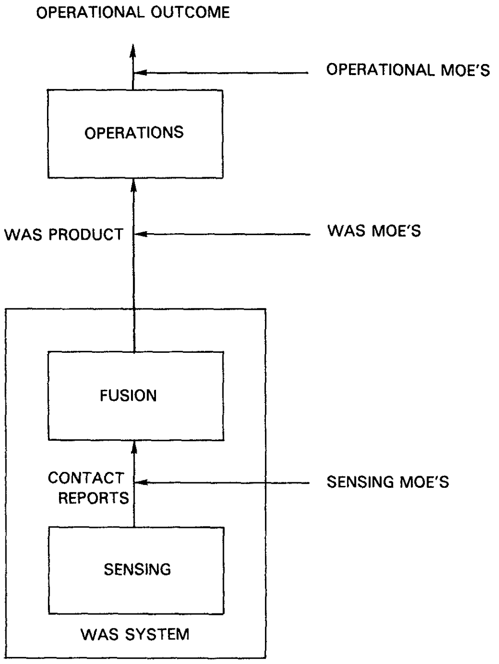


Fig. 2 — Hierarchy of MOE's

c. Missile Hits on Our Ships. The change in the number of missiles that actually hit our ships. To determine this, one has to consider the effects of area and point defense hard kill (AEGIS and CIWS for example) and area and terminal EW (SLQ-32, SLQ-17, SRBOC, ALQ-99 for example).

d. Damage to Our Ships. The change in the damage to our ships as a result of the change in missile hits. To obtain this data, one must be able to characterize ship vulnerability to missile hits.

Offensive Strikes

a. Missiles on Target. The change in the number of missiles (Harpoon, Tomahawk) that hit hostile ships. The determination of this MOE requires modeling the target designation, target acquisition, and homing functions for the missiles.

b. Missiles on Friendlies/Neutrals. The change in the number of missiles that hit friendly or neutral ships.

c. Damage. The change in the damage done to hostile ships (sunk or partially or totally disabled). To determine this MOE, one must be able to relate ship damage to missile hits.

3. Fusion MOE's

If one looked at the fusion process (i.e. correlating contact reports and creating tracks) as a black box, MOE's that establish the performance of that black box can be defined in terms of its relevant response to its contact reports input. MOE's relating to fusion algorithms or hardware implementation are not necessarily the MOE's of the fused output although these factors will certainly affect the output MOE's. It's analogous to building a radar with MOE's of detection range and surveillance volume; MOE's that relate to radar peak power and antenna gain affect but are not the desired MOE's of detection range and surveillance volume.

Given the preceding philosophy, a candidate list of fusion MOE's follows.

a. Probability of a Platform Report. The probability of reporting those platforms that have been detected by one or more sensors.

b. Probability of a False Report. The probability of reporting a false track or position.

c. Accuracy of Platform Location. The accuracy to which a track or position is held.

d. HULTEC Capability. The capability of characterizing the platform by function, type or hull number.

e. Timeliness. The time between the first contact report and the reporting of that contact by the fusion center.

f. Probability of Target ID. The probability of correctly identifying a reported platform.

g. Cost. The sum of acquisition and operating costs.

h. Availability. The probability of the WAS product being available to the user when desired.

4. Sensor MOE's

The MOE's associated with sensing depend upon the type of sensor (radar, ESM, EO and the platform (RPV, aircraft, satellite, shore based). The following candidate list of MOE's is suggested.

a. Relevant Area Surveyed. The area that a sensor/platform can cover that is also common to a defined region relative to the battlegroup. Such a defined region might, for example, be a circle of 1500 nmi radius centered on the battle group. Where the platform has motion with limits much larger than the sensor range, the MOE will, to first order, be independent of the sensor design parameters and radar cross section. There will, however, be a strong dependence on these data for fixed radar sensors.

If one wants to cover a particular area without regarding other factors, this MOE will provide a first cut relative comparison as to how many sensors would be required as a function of sensor/platform.

b. Probability of Detection. The probability of detection by a given sensor under a set of operationally relevant conditions.

c. Rate of Relevant Area Surveyed. The relevant area surveyed per unit time (through detection). This MOE will provide a relative comparison of the time required to survey a given relevant area for different radar/platforms, i.e. SBR, ROTHR, RPV.

d. Error. The accuracy of a given detection. This will mostly be geo location but it can also reflect the probability of a false alarm or the probability of a false emitter/platform identification for an ESM sensor.

e: HULTEC Capability. The capability of the sensor to characterize the platform, i.e. function, type, hull number.

f. Signature Detectability. The type of target signature that is capable of being detected (reflecting, RF emitting, IR emitting) and for passive sensing, the frequency coverage of the sensor.

g. Availability. The probability that a sensor will be available when required (including environmental factors affecting sensor performance, i.e., clouds, ducting, etc.).

h. Cost. The sum of procurement costs and operating costs that the deployment of the asset will entail.

i. Cost to Defeat. The relative cost required to defeat a given sensor/platform. This MOE will reflect how much the enemy has to spend to defeat a sensor relative to what we have to spend to deploy it. Clearly, ratios much less than unity will be an argument against deployment of such sensor/platforms.

An average MOE of different sensor mixes can be obtained if one can characterize or model the different sensors and determine the relationship to individual properly weighted MOE's. The procedure is useful in bounding the range of inputs to the next higher level of evaluation, i.e. fusion, but can not be used to evaluate whether or not a given sensor mix should be acquired. That determination requires an assessment of the contribution of the sensor mix to battle group operations.

5. Use of MOE's

Figure 3 summarizes how MOE's are used. There is a set of inputs and one uses analysis, and modeling and simulation to determine outcomes which are then evaluated against an agreed on set of weighted MOE's to obtain an average measure of effectiveness for a given input set.

One should realistically not expect to obtain absolute answers by using MOE's. For one, MOE's to some extent exist in the eye of the beholder and are, therefore, somewhat arbitrary. One could achieve a consensus on MOE's among a group of knowledgeable people but a second group will probably arrive at a different set of MOE's.

Another reason why absolute answers should not be expected is that simulations are not precise. Many simulations use men-in-the-loop and the results obtained will depend upon the operator's proficiency. Additionally, systems can not always be accurately characterized and, in many cases, there will be neither the time nor the funding to obtain a statistically adequate number of runs.

Never the less, the results obtained can be used to guide a WAS acquisition strategy if one looks for relative effectiveness values, trends, and robust solutions.

6. Summary

In wide area surveillance, sensors on platforms detect and provide contact reports which are fused to form a WAS product. This product is used by an operational commander to improve his decision making so that he can inflict greater damage on an enemy and sustain lower losses.

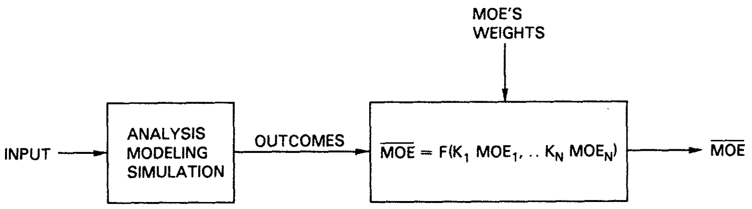
A hierarchy of MOE's was defined and, in principle, an evaluation of relative performance can be obtained for any level if the transfer function relating inputs to agreed on MOE's can be analyzed, modeled, or simulated.

The recommended procedure would be to use a top down approach in order to bound the problem early on to limit the number of cases considered. As an example, one would limit parametric excursions in the WAS product to those values that resulted in meaningful changes in operational effectiveness. Similarly, excursions in contact reports as a result of sensor mix variations would be limited to ranges that produced WAS product variations that have been previously determined to be necessary to cause significant changes in operational effectiveness.

The characterization of sensing and operations is reasonably straightforward; various models and simulations exist. The same can not be said for fusion; it is a complex process and involves men-in-the-loop as well as immature algorithm development. A common misconception when considering fusion is that correlation tracker MOE's are also fusion MOE's. This is not the case since we are concerned with measuring the fusion process against a given scenario and not evaluating the different subsystems that make up the fusion system.

Although a first order "relative goodness" of sensing and fusion can be obtained if those functions can be adequately modeled, the sequential connection of all of the models in an over all end to end model or simulation is required to make the evaluation relevant. The end to end configuration is conceptual; the models of the three functions need not all run in the same real time nor do they have to be colocated. All that is required is that the outputs of one model be the inputs to the next model in the sequence, and the models use the same scenario.

The"bottom line" is that the performance of the different systems that make up a wide area surveillance system must ultimately be evaluated by considering both their contribution to the WAS product *and* the WAS product's contribution to battlegroup operations. It makes no sense, for example, to acquire or improve sensors unless that acquisition or improvement will improve the WAS product *and* the WAS product will increase the Fleet's operational capability.



- DON'T LOOK FOR ABSOLUTES
- DO LOOK FOR TRENDS, RELATIVE VALUES, ROBUST SOLUTIONS

Fig. 3 — Use of MOE's

ON THE DESIGN OF DISTRIBUTED ORGANIZATIONAL STRUCTURES*

Pascal A. Remy
Alexander H. Levis
Victoria Y.-Y. Jin

Laboratory for Information and Decision Systems
Massachusetts Institute of Technology, Cambridge, MA 02139

ABSTRACT

The problem of designing human decisionmaking organizations is formulated as an organizational form problem with special structure. Petri Nets are used to represent the organizational form. An algorithmic procedure, suitable for computer-aided design, is presented and the specific algorithms that it includes are developed. The approach reduces the dimensionality of the problem to a tractable level.

INTRODUCTION

There are two basic problems in organizational design: the problem of organizational form and the problem of organizational control. Most of the theoretical developments in decision and control theory have addressed the latter problem: given an organizational structure, determine the decision rules or strategies that optimize some performance criterion. The former problem has been addressed only indirectly, i.e., given an organizational form, evaluate its performance according to some criteria and then change in some ad hoc manner the organizational form until a satisfactory structure has been obtained. The reason for this approach is that the general organizational form problem becomes computationally infeasible, even for a small number of organizational units.

In this paper, the organizational form problem is posed for a well defined class of organizations - those that have fixed structure and can be represented by acyclical directed graphs. These structures represent distributed decisionmaking organizations performing well defined tasks under specified rules of operation. Such organizations have been modeled and analyzed in a series of papers [1-4]. The basic unit of the models is the interacting decisionmaker with bounded rationality. The set of interactions will be generalized in Section 2 to allow not only for information sharing and command inputs, but also several forms of result sharing between decisionmakers. While this generalization increases the dimensionality of the design problem, it also allows for more realistic models of actual organizational interactions.

The mathematical formulation of the problem is based on the Petri Net description of the organizational structure. Furthermore, the dimensionality of the combinatorial problem is reduced by utilizing the notion of information paths within the organization. A number of new concepts are introduced that bound the problem to the search for alternative organizational forms from within the set of feasible structures only. The introduction of structural constraints, which characterize the class of organizations under consideration, and of user constraints that are application specific, lead to an algorithmic approach that is implementable on a personal computer. The mathematical model of the organization is described in the second section. In the third section, the various constraints are introduced. In the fourth section, the algorithm is described, while in the fifth a nontrivial example is presented.

MATHEMATICAL MODEL

The single interacting decisionmaker is modeled as having four stages or actions, the situation assessment (SA) stage, the information fusion (IF) stage, the command interpretation (CI) stage, and the response selection (RS) stage. In the SA stage, external inputs -- data from the environment or other members of the organization are processed to determine the situation assessment. This information is transmitted to the IF stage where it is fused with situation assessments communicated by other organization members. The resulting revised situation assessment is used to select a response in the response selection stage. The responses can be restricted by commands received by the CI stage that precedes the RS stage. An individual decisionmaker could receive inputs therefore at the SA stage, the IF stage, and the CI stage. It can produce outputs only by the SA stage and the RS stage. The exchange of information between the situation assessment and the information fusion stages of different decisionmakers constitute <u>information sharing</u> among them. On the other hand, what is being transmitted from the response selection stage of one decisionmaker (DM) to the IF stage of another could be the decision made by the first DM; in this case, the interaction is of the <u>result sharing</u> type. If the transmission is from the RS stage of one to the CI stage of another, then the former is issuing a command to the latter. This interaction imposes a hierarchical relationship between decisionmakers, - one is a commander, the other is a subordinate - while the other interactions don't.

The use of Petri Nets for the modeling of decisionmaking organizations was presented in [3] and exploited in [4]. Petri Nets [5] are bipartite directed multigraphs. The two types of nodes are places, denoted by circles and representing signals or conditions, and transitions, denoted by bars and representing processes or events. Places can be connected by links only to transitions, and transitions can be connected only to places. The links are directed. Tokens are used to indicate when conditions are met - tokens are shown in the corresponding place nodes. When all the input places

*This work was carried out at the MIT Laboratory for Information and Decision Systems with partial support by the U.S. Office of Naval Research under Contract No. N00014-84-K-0519 and by the Joint Directors of Laboratories under Contract No. ONR/N00014-85-K-0782.

to a transition contain tokens, then the transition is said to be enabled and it can then fire. Properties of Petri Nets are the subject of current research, e.g., references [5] - [8].

Figure 1 shows the Petri Net model of the single interacting decisionmaker. The DM can receive inputs (u) only at the SA, IF, and CI stages and produce outputs (y) only by the SA and RS stages, as stated earlier.

The allowable interactions between two decisionmakers are shown in Figure 2. For clarity, only the interactions from DM^i to DM^j are shown. The interactions from DM^j to DM^i are identical. The superscripts i or j denote the decisionmaker; the pair of superscripts ij indicates a link from DM^i to DM^j. Consider the general case of an organization consisting of N decisionmakers, a single input place, and a single output place. The last two are not really restrictions; for example, multiple sources can be represented by a single place and a transition that partitions the input and distributes it to the input places of the appropriate organization members. The organizational structure, as depicted by the Petri Net, can be expressed in terms of two vectors and four matrices. The elements of these vectors and matrices can take the value of zero or of one; if zero, then there is no connection, if one, then there is.

The interaction between the organization and the external source (input) is represented by an N-dimensional vector $\underline{e}$ with elements e^i. The output from the RS stage to the external environment is represented by the N-dimensional vector $\underline{s}$ with elements s^i.

The information flow from the SA stage of DM^i to the IF stage of DM^j is denoted by F^{ij}. Since each DM can share situation assessment information with the other N-1 DMs, the matrix F is N x N, but with the diagonal elements identically equal to zero.

Similarly, the links between the RS stage of a DM and the SA stage of the others are represented by the matrix G; the links from the RS stage to the IF stage by the matrix H; and the links from the RS stage to the CI stage by the matrix C. These three matrices are also N x N and their diagonal elements are identically equal to zero.

Therefore

$$\underline{e} = [e^i],\ \underline{s} = [s^i] \qquad 1 \leq i \leq N\ ,\ 1 \leq j \leq N$$

$$F = [F^{ij}],\ G = [G^{ij}],\ H = [H^{ij}],\ C = [C^{ij}]$$

$$F^{ii} = G^{ii} = H^{ii} = C^{ii} \equiv 0,\ \text{all } i$$

There are, altogether, 2^m possible combinations of different vectors $\underline{e}$, $\underline{s}$ and matrices F, G, H, and C, where $m = 4N^2 - 2N$. For a five member organization (N=5), m is equal to 90 and the number of alternatives is 2^{90}. Fortunately, many of these are not valid organizational forms and need not be considered. In the next section, the allowable combinations will be restricted by defining a set of structural constraints.

CONSTRAINTS

Four different structural constraints are formulated that apply to all organizational forms being considered.

R_1 The structure should have no loops.

R_{2a} The structure should be connected, i.e., there should be at least one undirected path between any two nodes in the structure.

R_{2b} A directed path should exist from the source to every node of the structure and a directed path should exist from any node to the output node.

R_3 There can be at most one link from the RS stage of a DM to each one of the other DMs, i.e., for each i and j, only one of the triplet (G^{ij}, H^{ij}, C^{ij}) can be nonzero.

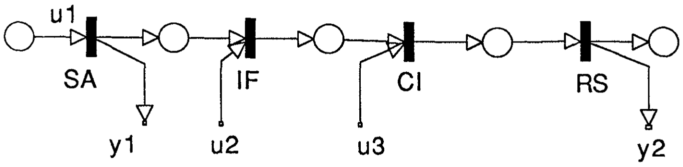


Figure 1. Aggregated Model of Interacting Decisionmaker

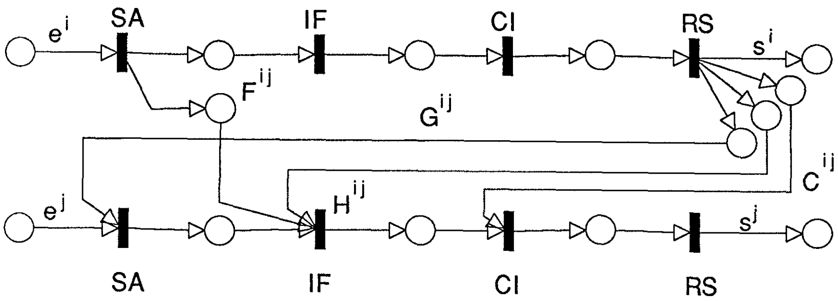


Figure 2. Modeled Interactions Between Two Decisionmakers

R_4 Information fusion can take place only at the IF and CI stages, consequently, the SA stage of each DM can have only one input from outside of the DM.

The set of structural constraints is defined as

$$R_s = \{R_1, R_{2a}, R_{2b}, R_3, R_4\}$$

The first constraint allows acyclical organizations only. The second and third define connectivity as it pertains to this problem; it eliminates structures that do not represent a single organization. The last two reflect the meaning of the four-stage decisionmaking model.

In addition to these constraints, the organization designer may introduce additional ones that reflect the specific application he is considering. For example, there may be a hierarchical relationship between the decisionmakers that must be maintained in the organizational structure. Then, the appropriate 0s and 1s will be placed in the arrays {e,s,F,G,H,C} thus restricting even further the organizational design problem solution. Lastly, to accommodate some very specific kind of interactions, the organization designer may imput links between the decisionmakers that are not modeled by the arrays mentioned above. Those links are however fixed and therefore do not increase the dimensionality of the design problem. They will be referred to as special constraints. Let all these constraints be denoted by R_u.

A Petri Net whose structure can be modeled by the four matrices and two vectors {F,G,H,C} and $\{\underline{e},\underline{s}\}$, respectively, will be called a Well Defined Net (WDN). A WDN that fulfills the structural constraints R_s and the designer's contraints will be called a Feasible organization form.

The notion of a subnet of a well defined net (WDN) can be defined as follows: Let W be a WDN specified by the set of arrays $\{\underline{e},\underline{s},F,G,H,C\}$. Let W' be a second WDN specified by the set $\{\underline{e}',\underline{s}',F',G',H',C'\}$. Then W' is a subnet of W if and only if

$$\underline{e}' \leq e \;,\quad \underline{s}' \leq \underline{s} \;,\quad D' \leq F$$

$$G' \leq G \;,\quad H' \leq H \;,\quad C' \leq C$$

where the inequality between arrays means that

$$(A' \leq A) \iff (\forall\, i \;,\; \forall\, j \quad A'_{ij} \leq A_{ij}).$$

Therefore, W' is a subnet of W if any interaction in W' (i.e., a 1 in any of the arrays $\underline{e}',\underline{s}',F',G',H',C'$) is also an interaction in W (i.e., a 1 in the corresponding array of W). The union of two subnets W_1 and W_2 is a new net that contains all the interactions that appear in either W_1 or W_2 or both.

DESIGN ALGORITHM

Let R be the set of contraints $R_s \cup R_u$. The design problem is to determine all the Feasible Organizational Forms, $\Phi(R)$, i.e., all the WDNs that satisfy the set of contraints R. The approach is based on defining and constructing two subsets of feasible organizational forms: the maximally connected organizations and the minimally connected organizations.

A Feasible Organizational form is a Maximally Connected Organization (MAXO) if and only if it is not possible to add a single link without violating the constraint set R. The set of MAXOs will be denoted by $\Phi_{max}(R)$.

A Feasible Organizational form is a Minimally Connected Organization (MINO) if and only if it is not possible to remove a single link without violating the constraint set R. The set of MINOs is denoted by $\Phi_{min}(R)$.

Consider now the designer's constraints R_u. The well defined nets that satisfy the constraints R_u are denoted by the set $\Omega(R_u)$. For a given number of decisionmakers, the maximally connected net associated with the set of constraints R_u is obtained by replacing all the undetermined elements of $\{\underline{e},\underline{s},F,G,H,C\}$ with 1s. This particular net is denoted by $\tilde{\Omega}(R_u)$. Therefore, by construction, $\tilde{\Omega}(R_u)$ is unique.

Proposition 1: Any feasible organization $\Phi(R)$ is a subnet of $\tilde{\Omega}(R_u)$.

Since any element of $\Phi(R)$ must satisfy the set of constraints R_u and since $\tilde{\Omega}(R_u)$ is the MAXO with respect to R_u, the elements of $\Phi(R)$ must be subnets of $\tilde{\Omega}(R_u)$.

Since $\tilde{\Omega}(R_u)$ is a Petri Net, it has an associated incidence (or flow) matrix Δ, [4]. The rows of the incidence matrix represent the places, while the columns represent the transitions. A -1 in position Δ_{ij} indicates that there is a directed link from place i to transition j; a +1 indicates a directed link from transition j to place i, while a 0 indicates the absence of a directed link between place i and transition j.

An integer vector $\underline{q}$ is an s-invariant of $\tilde{\Omega}(R_u)$ if and only if

$$\Delta'\underline{q} = 0$$

A simple information path of $\tilde{\Omega}(R_u)$ is a minimal support s-invariant of $\tilde{\Omega}(R_u)$ that includes the source node (source place) (for details, see [4]). This simple path is a directed path without loops from the source of the net to the sink.

Proposition 2: Any well defined net that satisfies the constraints R_u and the connectivity constraint R_{2b} is a union of simple paths of $\Omega_{max}(R_u)$.

Proof: If a WDN $\mathbb{T}$ satisfies the constraint set R_u, then it is a subnet of $\tilde{\Omega}(R_u)$, by the definition of $\tilde{\Omega}(R_u)$. Constraint R_{2b} implies that every node of $\mathbb{T}$ is included in at least one simple path since there is a path from the source to the node and a path from the node to the output node. Therefore, $\mathbb{T}$ is a union of simple paths of $\tilde{\Omega}(R_u)$.

Corollary: Any feasible organization Φ is a union of simple paths of $\tilde{\Omega}(R_u)$.

Let $Sp(R_u)$ be the set of all simple paths of $\tilde{\Omega}(R_u)$, i.e.,

$$USp(R_u) = \{sp_1, sp_2, \ldots, sp_r\}$$

and let $U\underset{\sim}{S}p(R_u)$ denote the set of all unions of simple paths of $\tilde{\Omega}(R_u)$. From now on, only WDNs that are elements of $U\tilde{S}p$ need be considered.

The procedure described so far can be summarized by a sequence of four steps.

Step 1: Given the set of contraints R_u, define the set of arrays $\{\underline{e},\underline{s},F,G,H,C\}$ that satisfy these constraints.

Step 2: Construct the maximally connected net $\tilde{\Omega}(R_u)$ by replacing with 1s all the undetermined elements in the six arrays.

Step 3: Find all the simple paths of $\tilde{\Omega}(R_u)$ using the algorithm developed by Jin [9] or the algorithm of Martinez and Silva [10] which generates all minimal support s-invariants of a general Petri Net using linear algebra tools. An improved version of this algorithm has been proposed by Toudic [11].

Step 4: Construct the set of all unions of simple paths of $\tilde{\Omega}(R_u)$.

From the corollary, the set $\{\Phi\}$ is a subset of $USp(R_u)$. Consequently, the number of feasible organizational forms is bounded by 2^r. The dimensionality of the problem is still too large. One more step is needed to reduce the computational effort.

Proposition 3: Let $\tilde{\Pi}$ be a WDN that is a union of simple paths of $\tilde{\Omega}(R_u)$. Then Π is a feasible organization form, i.e., $\Pi \varepsilon \{\Phi\}$, if and only if, (a) there is at least one MINO which is a subnet of Π, and (b) Π is the subnet of at least one MAXO.

The MAXOs and MINOs can be thought of a the "boundaries" of the set $\{\Phi\}$. The next step is to find a procedure for constructing the MAXOs and the MINOs corresponding to the constraint set R. Since Π is a subset of $USp(R_u)$, it follows that Φ_{min} is a subset of $USp(R_u)$. Then, one can scan all the elements of USp and select those that satisfy the constraint set R.

To guide the search for MINOs, the links of a net are divided into two categories: fixed and free links. Fixed links refer to requirements that cannot be transgressed and correspond to the 1 entries in the constraint matrices $\{\underline{e},\underline{s},F,C,G,H\}$ or to the special constraints. Free links correspond to the unspecified elements of the above mentioned matrices: they may or may not be present. Any feasible organization must include all fixed links. Associated with the fixed links are places - therefore, these places are also fixed and must be present in the organization. An index, hd(p) is associated with each fixed place p: it is the number of simple paths containing the place p.

Clearly, if hd(p) = 1, the only simple path going through the place p has to be included in all MINOs. It is therefore useless to consider elements of $Sp(R_u)$ that do not contain this specific simple path. The scanning of the set $USp(R_u)$ is done by taking advantage of the insight brought by the index hd.

The search process starts by picking from among the fixed places the one with the smallest index hd; this place is denoted as p_{min} (if there are several such places, one of them is selected, arbitrarily).

Then, one by one, all the simple paths, sp, going through the place p_{min} are considered. For each of them, the remaining fixed places are searched for the one with the smallest index hd. The procedure is repeated until there are no more fixed places.

At each step, an element of $USp(R_u)$ is found and checked against the constraints: if they are violated the scanning stops and returns to the previous step.

If the number of remaining fixed places is zero and if the structural constraints are not violated, a MINO has been found.

Whenever a MINO or an element of $USp(R_u)$ violating the structural constraints is found, it is eliminated from the subsequent scanning.

To determine the MAXOs, a similar procedure is used but instead of building the subnets by taking the union of simple paths, the scanning starts from the net $\tilde{\Omega}(R_u)$. Subnets are constructed by removing paths until a feasible form is found. Therefore, the fifth and sixth steps of the algorithm are:

Step 5: Search the set USp to find the minimally connected organizations.

Step 6: Search the set USp to find the maximally connected organizations.

Implicit in Steps 5 and 6 is the ability to test efficiently whether constraints R are satisfied. Indeed, if the interconnection matrix (see Ref. [4]) for the net $\tilde{\Omega}(R_u)$ is constructed, then the checking for the constraints R reduces to simple tests on the elements of the interconnection matrix.

APPLICATION

The procedure is illustrated in this paper for the case of a five person organization modeling the ship control party of a submarine. This organization, as it currently exists, has been modeled and analyzed by Weingaertner [12] and is represented in its Petri Net form in Figure 3. At the top of the hierarchy is the Officer of the Deck (DM1) with responsibility for all ship control matters pertaining to the conduct of the submarine's mission. He receives information both from the external environment and from the Diving Officer of the Watch (DM2). He issues command to DM2.

The Diving Officer of the Watch is responsible for the bulk of the control decision process. He receives information from and sends commands to the remaining members of the organization: the Chief of the Watch (DM3), the Lee Helm (DM4), and the Helm (DM5). The decisionmakers DM3, DM4 and DM5 can be considered the sensors and the actuators of the organization. They received information from the external environment (ship control panels,...) and can act on the external environment (stern planes, fairwater planes,...).

The boldface links of Figure 3 represent the fixed links of the organization. They denote the explicit hierarchical structure existing between the members of the organization.

The design problem is to consider alternative feasible organizational forms that could possibly have better performance measures than the actual one. Figure 4 shows the matrices $\underline{e}$, $\underline{s}$, F, G, H, C used in the design of alternative organizational forms for the problem under consideration.

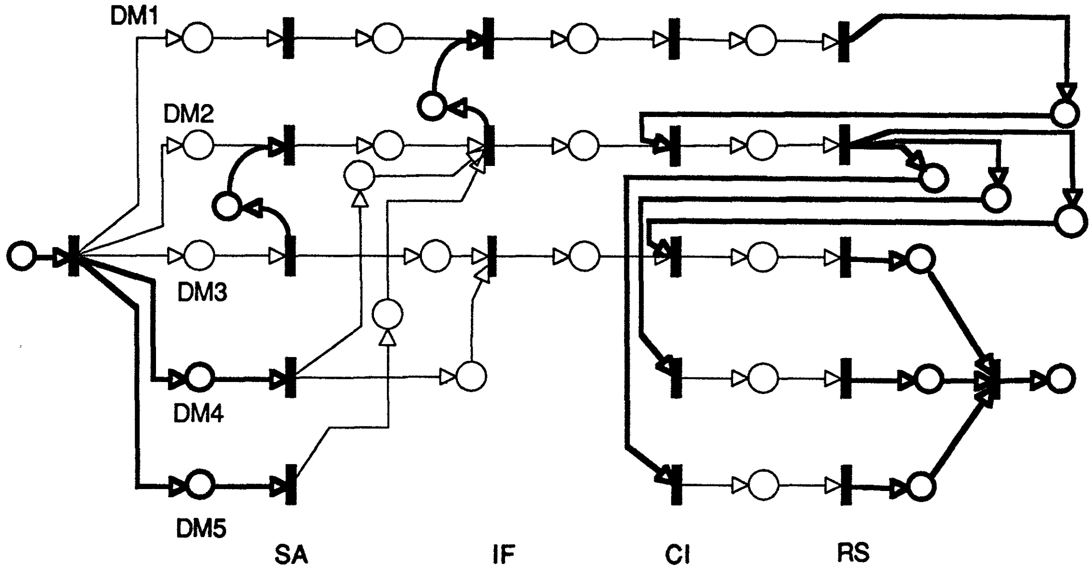
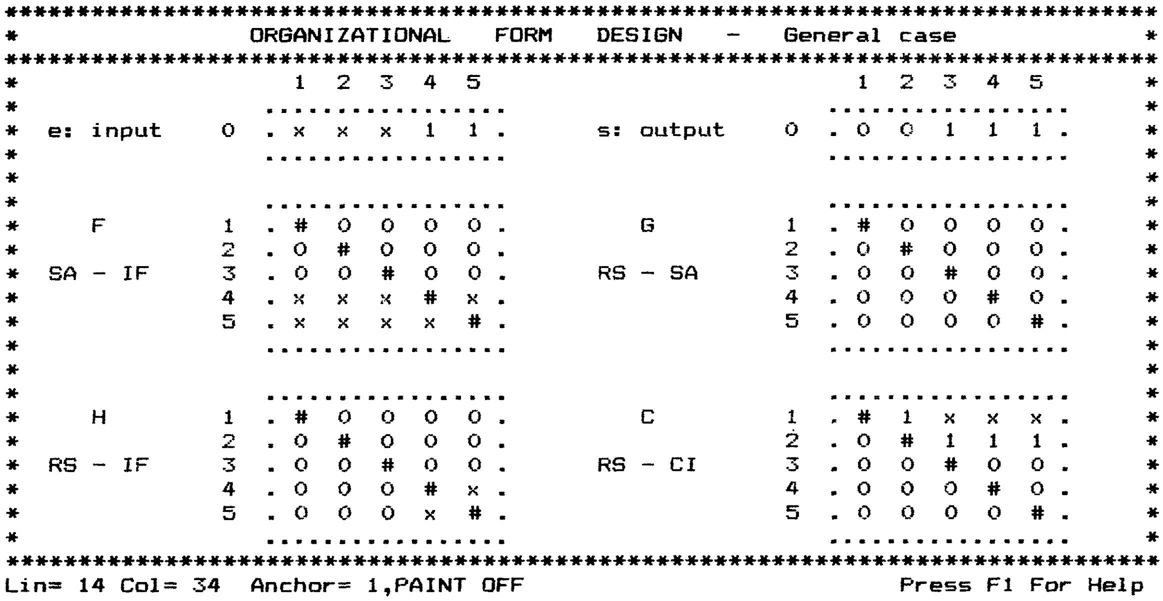


Figure 3. Present Organization



```
**********************************************************************
*              ORGANIZATIONAL   FORM   DESIGN   -   General case     *
**********************************************************************
*                   1  2  3  4  5                    1  2  3  4  5   *
*                 .................                ................. *
*  e: input    0  . x  x  x  1  1 .    s: output  0 . 0  0  1  1  1 .*
*                 .................                ................. *
*                                                                    *
*                 .................                ................. *
*     F        1  . #  0  0  0  0 .       G       1 . #  0  0  0  0 .*
*              2  . 0  #  0  0  0 .               2 . 0  #  0  0  0 .*
*  SA - IF     3  . 0  0  #  0  0 .    RS - SA    3 . 0  0  #  0  0 .*
*              4  . x  x  x  #  x .               4 . 0  0  0  #  0 .*
*              5  . x  x  x  x  # .               5 . 0  0  0  0  # .*
*                 .................                ................. *
*                                                                    *
*                 .................                ................. *
*     H        1  . #  0  0  0  0 .       C       1 . #  1  x  x  x .*
*              2  . 0  #  0  0  0 .               2 . 0  #  1  1  1 .*
*  RS - IF     3  . 0  0  #  0  0 .    RS - CI    3 . 0  0  #  0  0 .*
*              4  . 0  0  0  #  x .               4 . 0  0  0  #  0 .*
*              5  . 0  0  0  x  # .               5 . 0  0  0  0  # .*
*                 .................                ................. *
**********************************************************************
Lin= 14 Col= 34  Anchor= 1,PAINT OFF                Press F1 For Help
```

Figure 4. Simplified Representation of the Screen

A computer-aided design procedure has been implemented on an IBM PC/AT with 512K RAM and a 20 MB hard disk drive. The six arrays for organizations with up to 5 members are shown graphically on the color monitor and the user can interact with them. A simplified printout of the screen can be obtained (Fig. 4). The symbol # denotes that no link can exist at this location. A 0 indicates the choice that no link be at that location, a 1 that a link must exist at that location, and an x indicates that the choice is open: the x's represent the degrees of freedom in the design.

The 1's in Figure 4 represent the fixed links of Figure 3. The x's represent all allowable interactions. Figure 5 is a graphical representation of the well defined net represented by the arrays in Figure 4. Indeed, the WDN shows all the interactions allowed by the organization designer. The fixed contraints are presented by boldface links. There are 101 simple paths in the universal net, as determined independently by both the Jin [9] and the Martinez and Silva [10] algorithms.

The algorithm presented in this paper produces 25 MINOs and 2 MAXOs. For this problem, it took 3 minutes, using Jin's algorithm to find the invariants to determine the complete solution. When Martinez and Silva's algorithm is used to find invariants, the same run takes 7 minutes. One MINO and one MAXO are reproduced in Figures 6 and 7 respectively. They have been selected so that the original organization of Fig 3 is located between them. As expected, the original organization is one specific solution of the design problem. Alternative solutions can be analyzed (see, for example, [12]) to determine preferred designs.

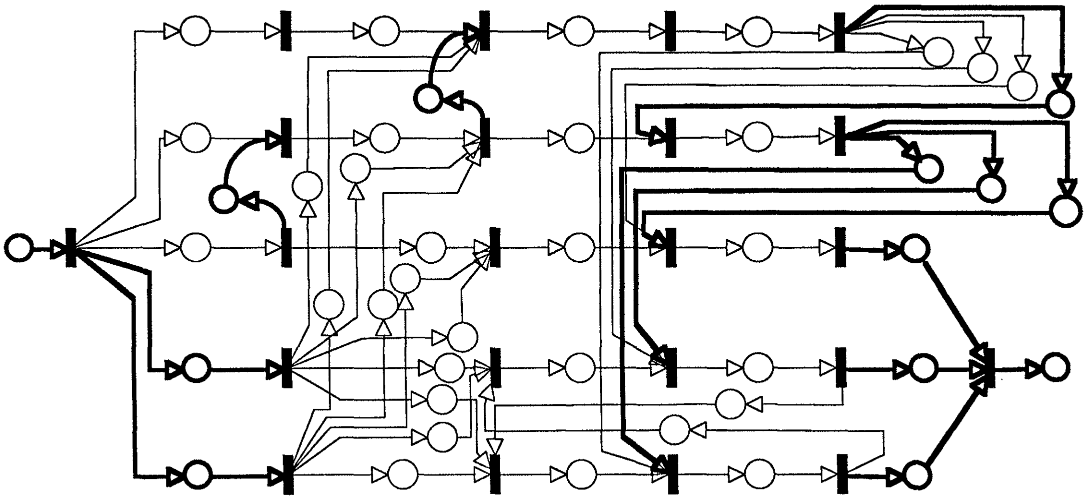


Figure 5. Well Defined Net of Example

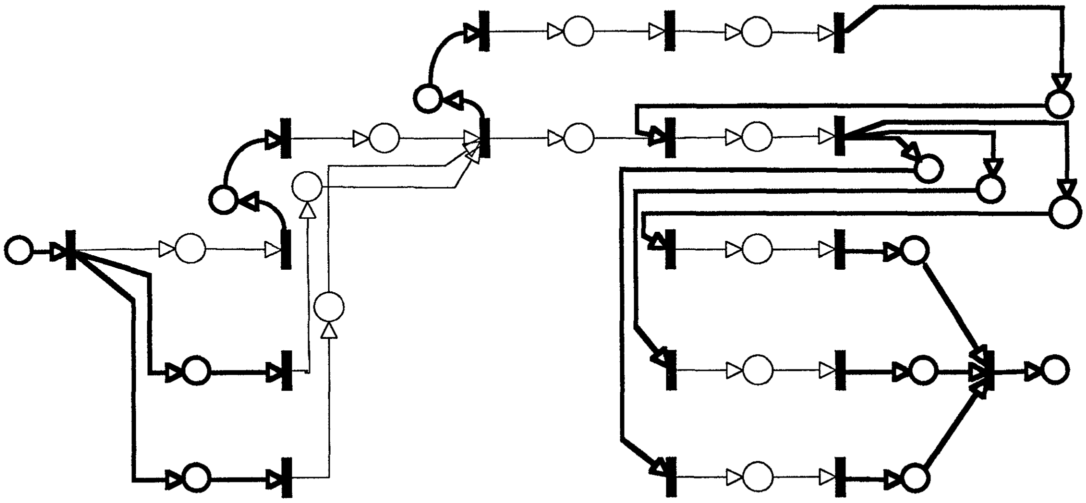


Figure 6. One of the 25 MINOs

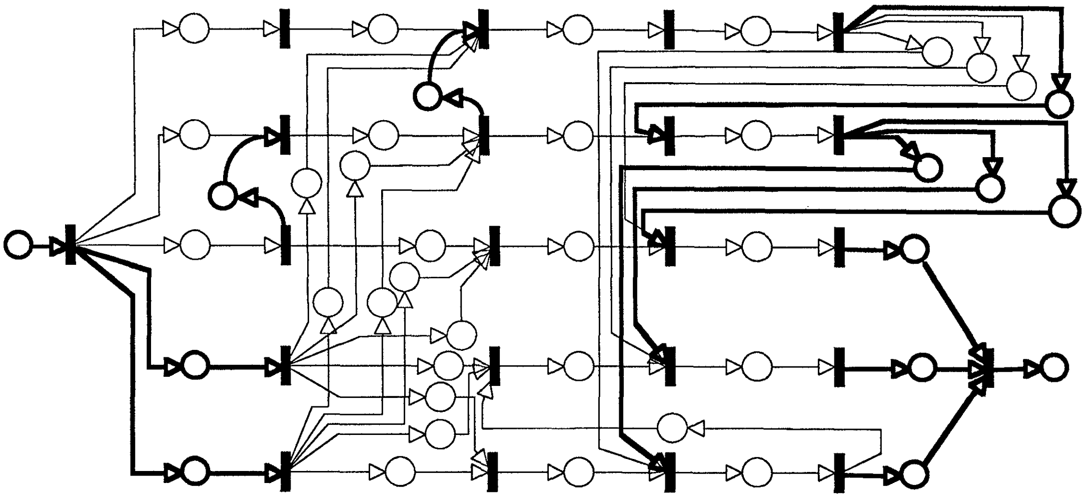


Figure 7. One of the 2 MAXOs

CONCLUSION

The organizational from problem has been described and a mathematical formulation based on Petri Nets has been presented. An algorithm that reduces the problem to a tractable level has been introduced that takes into account the special structure of human decisionmaking organizations. A preliminary implementation of the algorithm on a microcomputer is described.

REFERENCES

[1] K. L. Boettcher and A. H. Levis, "Modeling and Analysis of Teams of Interacting Decisionmakers with Bounded Rationality," Automatica, Vol. 19, No. 5, November 1983.

[2] A. H. Levis, "Information Processing and Decisionmaking Organizations: A Mathematical Description," Large Scale Systems, Vol. 7, 1984.

[3] D. Tabak and A. H. Levis, "Petri Net Representation of Decisions Models," IEEE Trans. on Systems, Man, and Cybernetics, Vol. SMC-15, No. 6, November/December 1985.

[4] V. Y.-Y. Jin, A. H. Levis, and P. Remy, "Delays in Acyclical Distributed Decisionmaking Organizations," Proc. 4th IFAC/IFORS Symposium on Large Scale Systems, Zurich, Switzerland, August 1986.

[5] J. L. Peterson, Petri Net Theory and the Modeling of Systems, Prentice-Hall, Englewood Cliffs, NJ, 1981.

[6] G.W. Brams, Reseaux de Petri: Theorie et Pratique, Masson,Paris, 1983.

[7] W. Brauer, Ed., Net Theory and Applications, Proceedings of the Advanced Course on General Net Theory of Processes and Systems, Hamburg 1979, Springer-Verlag #84, Berlin, 1980.

[8] G. Rozenberg, Advances in Petri Nets 1984, Springer-Verlag, Berlin, 1985.

[9] V. Jin, "Computation of Delays in Acyclical Distributed Decisionmaking Organizations," LIDS-TH-1459, MS Thesis, Laboratory for Information and Decision Systems, MIT, May 1985.

[10] J. Martinez and M. Silva, "A Simple and Fast Algorithm to Obtain all Invariants of a Generalized Petri Net," in Application and Theory of Petri Nets, Giraud C. and Reisig W., Eds., Springer-Verlag #52, Berlin, 1980.

[11] J. M. Toudic, Algorithmes d'analyse Structurelle des Reseaux de Petri, These 3eme Cycle, Universite Paris VI, 1981.

[12] S. Weingaertner, "A Model of Submarine Emergency Decisionmaking and Decision Aiding," MS Thesis, Department of Mechanical Engineering, MIT, 1986.

PERFORMANCE MEASURES FOR THE EVALUATION OF C^3 SYSTEM CONFIGURATIONS

Donald R. Edmonds

Nichols Research Corporation
McLean, Virginia

ABSTRACT

This paper develops basic concepts for the evaluation of C^3 system configuration options based on block and Petri net diagrams, system characteristics, performance requirements and performance measures. Five performance measures are developed: connectivity, operability, survivability, timeliness and information utility. The methodology is applied to a simple illustrative example.

1. INTRODUCTION

The purpose of this paper is to develop some basic concepts for the evaluation of command, control and communications (C^3) system configurations or architecture options on a quantitative basis. A typical military C^3 system is composed of sensors, communication links, information processors, command centers, weapons and human decisionmakers distributed over many locations or platforms. Moreover, the command and control (C^2) process encompasses such diverse functions as compiling information on threats, force performance and damage; assessing the status of enemy and friendly forces from the information compiled; and directing offensive and defensive operations through the use of weapons. A formidable problem for the C^3 design analyst is the choice of meaningful performance measures and a way to relate these measures to system configuration.

Karam and Levis [1,2] have developed a general approach to configuration evaluation based on two routes of analysis, namely a system route and a mission route. In the system route system characteristics are related to performance measures. In the mission route a utility function is defined based on the importance of each performance measure and the error in performance measures. Assuming the performance measures to be subject to probabilistic variation, an expected value of mission utility is computed and system configurations are ranked on the basis of the expected value.

In this paper the work on the system analysis route is extended to include five performance measures which are general enough to apply to most C^3 systems: connectivity, operability, survivability, timeliness and information utility. It is shown how these measures can be derived from two basic diagrams: a block diagram and a Petri net diagram. A block diagram indicates the physical organization of individual components and the information flow between them. Each block represents a component which performs one or more tasks.

A Petri net diagram is essentially a flow chart which portrays the time sequence of the C^2 tasks performed. Tabac and Levis [3] have shown that the task sequence in C^3 systems can be described by five basic symbols in Petri net diagrams. These symbols are circle node, bar node, multiple input place, multiple output place and a demultiplexer (representing a decision switch). These symbols can indicate the sequence of tasks, decision alternatives and mutual backup of tasks by various components. Using these concepts, any decisionmaking structure can be modeled by a Petri net diagram.

Section 2 presents the overall evaluation methodology in more detail. Section 3 presents the performance measure relationships and their connection with the two types of diagrams. In Section 4 an outline is given for computing the expected value of mission utility. Section 5 summarizes the methodology.

2. OVERALL METHODOLOGY

Figure 1 provides an overview of the methodology as extended from [2]. The context (or scenario) establishes the relevant factors regarding the military operational situation under which the system will operate. The context influences two independent routes of analysis, namely the system analysis route shown at the top of Figure 1 and the mission analysis route shown at the bottom of Figure 1.

In the system route, system configurations in terms of Petri net (task flow) diagrams and block diagrams delineate the time dynamics of the data processing (tasks) and the information exchange between components of the C^3 system. System characteristics (such as probability of component survival) are used to compute values of performance measures $\underline{x}$ (such as probability of system survivability).

In the mission route, importance factors for the performance measures are also based on the context of the military operational situation. The importance factors range say from zero to 10 with 10 being the most imporant weighting. The imporance factors are used to determine a utility function for the values taken on by the performance measures [1]. A typical utility function is:

$$u(\underline{x}) = 1 - \frac{(\underline{1}-\underline{x})'F'F(\underline{1}-\underline{x})}{\underline{1}'F'F\underline{1}} \qquad \text{eq(1)}$$

where the matrix F represents normalized importance factors for the performance measures (columns) for each evaluator (row).

The utility function and the performance measures are combined to determine a scalar value called expected mission utility E(u) given by:

$$E(u) = \int_{L_S} f(\underline{x})\ u(\underline{x})\ d\underline{x} \qquad \text{eq(2)}$$

where L_s is the multidimensional range of variation in performance measures. The range of variation in performance measures is determined by designating a range of variation in the values of the system characteristics.

Once E(u) is computed for all system

configurations under study, these configurations can be ranked on the value of E(u). Since u($\underline{x}$) is normalized, values of E(u) will range from zero to one with one being the desirable value.

3. SYSTEM ANALYSIS

C^3 system analysis as described in this paper consists of three steps, namely drawing block and Petri net diagrams, specifying system characteristics and performance requirements, and computing performance measures. These steps are described below using a simple example.

3.1 Block and Petri Net Diagrams

System analysis begins with the drawing of diagrams to depict system configuration and system operation [4]. Figure 2 shows the block and Petri net diagrams for a simple example of an Army C^3 system configuration. In Figure 2a blocks 11 and 21 represent two individual sensors (e.g., forward observers) which send information to an Intelligence/Electronic Warfare (IEW) fusion center (block 2). The processed information from IEW is sent to either of two Operations (G3) elements for tactical interpretation. Finally, either of the Operations elements (block 13 or 23) transfers information to a Commanding Officer (block 4) for decisionmaking.

Whereas the task sequence can sometimes be deciphered from simple block diagrams, the diagrams generally do not contain all the ingredients necessary to clearly depict task sequence. For example, from Figure 2a it is not clear whether both blocks 11 and 21 must report information to block 2 prior to block 2 performing its task. The other alternative is that block 2 will execute its task upon receiving information from either block 11 or block 21. Likewise, it is not clear whether both blocks 13 and 23 must provide information prior to block 4 executing its task. The other alternative is that block 4 will execute its task upon receiving information from either block 13 or 23.

Petri net diagrams resolve the questions concerning task sequence. In Figure 2b the tasks performed by blocks in the block diagram are depicted by short vertical bars known as transformations (TR). The oval represents a decision switch indicating that information from TR2 can be sent either to TR13 or TR23. TRs 13 and 23 are said to back each other up.

From a standpoint of temporal dynamics, the multiple inputs directly into TR2 imply that TR11 and TR21 perform tasks in parallel. In addition, multiple inputs represent an "and" condition in the sense that inputs from both TR11 and TR21 have to be received at TR2 before TR2 will begin processing. After fusion is completed by TR2, data is sent to either TR13 or TR23 for tactical interpretation. The oval in the Petri net diagram represents a switch with two positions: data flow to TR13 or data flow to TR23. Thus, there are two possible paths which the operation of the system can take. The circle in the Petri net diagram after TR13 and TR23 represents an "or" condition in the sense that TR4 will activate upon completion of either TR13 or TR23.

Other possibilities in drawing the Petri net diagram are shown in Figure 3 where Figure 3a, 3b and 3c represent "or" and "or", "or" and "and", and "and" and "and" situations, respectively, regarding TRs11,21 and TRs13,23. In the latter two cases, the decision switch has been eliminated since inputs from both TR13 and TR23 are required for TR4 to activate.

These conventions of drawing diagrams help to distinguish aspects of system configuration relevant to various performance measures. Their value will become evident in the next two subsections when Figure 2 will be used as an example to introduce computational features involved in the performance measures.

3.2 Performance Measures

Five performance measures are considered in this paper: survivability, operability, timeliness, connectivity and informantion utility. These measures are described in general at first. The general discussion is then followed by a mathematical presentation.

Survivability refers to the ability to resist physical destruction (hardkill) of components/sites performing tasks, leading to a continuation of task flow. Survivability is the probability of the system maintaining continuity of task flow among blocks as a function of block survival. When blocks are collocated (physically adjacent), they act as one component and possess a single value of probability of survival for the set of collocated blocks. (This assumes that a bomb or artillery projectile would knock out the entire collocated facility.)

Operability and timeliness are determined by the Petri net diagram. Operability refers to the chances of resisting electronic interference, deception or human errors (softkill) producing disruptions to satisfactory performance of tasks which lead to unsatisfactory continuation of task flow. Operability differs from survivability in that operability accounts for all tasks being satisfactorily completed, whereas survivability accounts for all blocks physically surviving. A distinction is that a single block may execute more than one task. Likewise, multiple blocks may be collocated (resulting in a single probability of hardkill per collocated set of blocks).

Timeliness refers to the chances that a set of required sequential tasks are accomplished within a specified time window (normally a performance requirement based on enemy repsonse time). Since there may be alternative paths of task accomplishment by a system, the critical path (longest time path) is used as the basis of evaluation. Any other path will have a shorter time than the longest path. Thus, in essence timeliness is based on the most pessimistic time.

Connectivity and information utility are determined by the block diagram. Connectivity refers to the linking between components in the block diagram. Connectivity is defined as the ratio of the number of direct connections between blocks over the total possible direct connections. Whereas operability takes into account task sequence, connectivity is a static measure portraying simply the "richness" of interconnections between blocks.

Information utility refers to the commonality of information sent by one block and wanted at another block. It is defined as the ratio of the total number of information items both sent by a sender and wanted by a receiver over all information items and averaged over all block combinations that are transferring information. At present no significance is given to the importance of one information item compared to another although refinements are possible.

3.3 Quantification of Performance Measures

In addition to block and Petri net diagrams, system characteristics are the basic (primitive) variables used as input to performance measure evaluation. Each performance measure is dependent on a set of system characteristics. This section will introduce the relationships between system characteristics and associated performance measures. Throughout the section the example introduced in Figure 2 will be used for purposes of illustrating calculation of the performance measures. The example in Figure 2 will be denoted as the Core example.

3.3.1 Connectivity. Connectivity C may be formulated as the ratio of the number of (direct) connections betwen blocks (components) as compared to the total possible number of (direct) connections. Connectivity is given by:

$$C = \frac{K}{\sum_{i=1}^{N} (N-i)} \qquad eq(3)$$

where the system characteristics K and N are:

K = number of connections between blocks

N = number of blocks

The value of C can vary between zero and one with one being the desired value. In Figure 2a of the Core example the value of K is 7 and N is 6, giving C a value of 0.47.

Some readers may be somewhat concerned about the simplistic nature of the denominator of equation 3 because it accounts for <u>all</u> combinations of blocks and not all blocks need to be connnected for a C^3 system to operate. More refined definitions which account for the importance of various connections are possible but left to future work. Nevertheless, the current definition is an important measure for C^3 systems because multiple paths for mutual backup (affecting survivability and operability) cannot occur without the system possessing a high degree of connectivity.

3.3.2 <u>Operability</u>. Let x_i be the state of accomplishment of a required task i where:

$$x_i = \begin{cases} 1, & \text{task i is successfully accomplished} \\ 0, & \text{task i is unsuccessfully accomplished} \end{cases}$$

Through reliability theory, the accomplishment of a set $\underline{x}$ of tasks can be stated as:

$$\phi(\underline{x}) = \begin{cases} 1, & \text{mission accomplished} \\ 0, & \text{mission unaccomplished} \end{cases}$$

where the relationship $\phi(\underline{x})$ is called a structure function. The probability of system operability P_o is given by:

$$P_o = P[\phi(\underline{x})=1] = E[\phi(\underline{x})]$$
$$= f[p_o(x_1), p_o(x_2), \ldots, p_o(x_n)] \quad eq(4)$$

where $p_o(x_1)$, $p_o(x_2)$, etc. are probabilities of task accomplishment. The probabilities $q_o(x_1)=1-p_o(x_1)$, etc. would be probabilities of softkill.

Simple procedures for deriving the structure function and computing P_o have been developed [5,6]. The steps consist of:

1. Finding all paths of task leading to mission accomplishment in the Petri net diagram
2. Identifying the minimum paths
3. Iteratively developing the structure function
4. Taking the expected value of the structure function

As an example consider the Core example in Figure 2. The task paths associated with this figure are:

11,21,2,13,4

and

11,21,2,23,4

These two paths also happen to be minimum paths. If another path 11,21,2,13,23,4 also existed, it would not be a minimum path because at least one other shorter (less tasks) path would exist.

The structure function is determined by developing combinations of the minimum paths by the laws of set theory. For our example:

$$\phi(\underline{x}) = x_{11}x_{21}x_2x_{13}x_4 + x_{11}x_{21}x_2x_{23}x_4$$
$$- x_{11}x_{21}x_2x_{13}x_{23}x_4 \quad eq(5)$$

If another minimum task path such as 11,21,2,13,5 existed, the next iteration of the structure functipn would be:

$$\phi(\underline{x}) = x_{11}x_{21}x_2x_{13}x_5 + x_{11}x_{21}x_2x_{13}x_4 + x_{11}x_{21}x_2x_{23}x_4$$
$$- x_{11}x_{21}x_2x_{13}x_{23}x_4 - x_{11}x_{21}x_2x_{13}x_4x_5$$
$$- x_{11}x_{21}x_2x_{13}x_{23}x_4x_5 + x_{11}x_{21}x_2x_{13}x_{23}x_4x_5$$

Taking the expected value of the structure function in equation 5 gives:

$$P_o = p_o(x_{11})p_o(x_{21})p_o(x_2)p_o(x_{13})p_o(x_4)$$
$$+ p_o(x_{11})p_o(x_{21})p_o(x_2)p_o(x_{23})p_o(x_4)$$
$$- p_o(x_{11})p_o(x_{21})p_o(x_2)p_o(x_{13})p_o(x_{23})p_o(x_4) \quad eq(6)$$

If all tasks had the same probability of success, i.e. if $p_o(x_i) = p_o$ for all i, then equation 6 reduces to:

$$P_o = 2p_o^5 - p_o^6 \quad eq(7)$$

3.3.3 <u>Survivability</u>. Let y_i be the state of a component i performing a required task where:

$$y_i = \begin{cases} 1, & \text{component i is active} \\ 0, & \text{component i is killed} \end{cases}$$

Through reliability theory, the survival of a system composed of a set $\underline{y}$ of components performing tasks can be stated as:

$$\phi(\underline{y}) = \begin{cases} 1, & \text{system survives} \\ 0, & \text{system is killed} \end{cases}$$

where the relationship $\phi(\underline{y})$ is a structure function. The probability of system survivability P_s is given by:

$$P_s = P[\phi(\underline{y})=1] = E[\phi(\underline{y})]$$
$$= f[p_s(y_1), p_s(y_2), \ldots, p_s(y_n)] \quad eq(8)$$

where $p_s(y_1)$, $p_s(y_2)$, etc. are probabilities of survival for individual components performing tasks. The quantities $q_s(y_1) = 1 - p_s(y_1)$, etc. would be probabilities of hardkill.

The simple procedures for deriving the structure function and computing P_o given earlier for operablity can also be used to compute P_s. The primary distinction is that now we are interested in the sequence of components used in the accomplishment of sequential tasks instead of the tasks themselves.

Again let us consider the Core example in Figure 2. Since there is a one-to-one relationship between tasks and components in this example, the relationship for P_s is the same as P_o, with of course $p_s(y_1)$ substituted for $p_o(x_1)$, etc. However, a condition that sometimes influences system survivability is the location of components. Let us suppose that blocks 13 and 4 are collocated so that destruction of one block also assures destruction of the other. The component paths are then:

11,21,2,13

and

11,21,2,23,13

Note in the first path 11,21,2,13 that no mention is made of block 4 because it is now considered collocated with block 13. In the second path 11,21,2,23,13 block 13 has been substituted for block 4 for the same reason. Since the second path includes the first path, there is only one minimum path, namely 11,21,2,13.

The survivability structure function for the system including collocation is:

$$\phi(\underline{y}) = y_{11}y_{21}y_2y_{13} \qquad \text{eq(9)}$$

and the probability of system survivability is:

$$P_s = p_s(y_{11})p_s(y_{21})p_s(y_2)p_s(y_{13}) \qquad \text{eq(10)}$$

If all the components have the same probability of survival, i.e. if $p_s(y_i) = p_s$ for all i, then:

$$P_s = p_s^4 \qquad \text{eq(11)}$$

The form of the relationships for operability and survivability (equations 6 and 10) differ because of the collocation condition introduced in the Core example. However, other factors can also make the relationships differ. For example, if any component in the Core example executed more than one task, the multiple tasks would result in a new relationship other than equation 6.

3.3.4 Timeliness. The performance measure timeliness is associated with the longest expected (time) task path in the Petri net diagram. Timeliness is defined as the probability that the critical path time is less than the time window specified for mission completion (which can be dependent on the enemy's response time). The theory behind PERT analysis is used to derive a measure of timeliness [7].

Let the random variable t_i be the time to complete task i, i = 1, 2, ..., n for a task path in the Petri net diagram. It is assumed that t_i is beta distributed and thus dependent on three parameters: a low (optimistic) time a_i, the mode (most likely time) m_i, and a high (pessimistic) time b_i. These times are system characteristics. The expected value $E(t_i)$ for task i is given by:

$$E(t_i) = (1/3)[2m_i + (1/2)\ (a_i + b_i)] \qquad \text{eq(12)}$$

and the variance σ_i^2 is:

$$\sigma_i^2 = [(1/6)\ (b_i - a_i)]^2 \qquad \text{eq(13)}$$

Let the total time t on a path in the Petri net diagram be the sum of the task times on the path, i.e.

$$t = \sum_{i=1}^{n} t_i \qquad \text{eq(14)}$$

It is assumed that the times t_i are statistically independent. Based on the central limit theorem, it is also assumed that t has a normal distribution with expected time E(t) and variance $\sigma^2{}_t$ equal to the sum of the $E(t_i)$ values and $\sigma^2{}_i$ values, respectively, on each path. The critical path is the path in the Petri net diagram with the largest expected value E(t) of total time t. Timeliness may then be defined as the probability P(T) of completing the tasks on the critical path before a specified time window T. Timeliness is given by:

$$P(T) = P(t < T) \qquad \text{eq(15)}$$

where t is normally distributed and T = required window for mission completion.

For example, in Figure 2b of the Core example suppose all the tasks except TR21 and 13 have values of $a_i = 1$, $m_i = 3$, and $b_i = 5$ hours. For TR21 and 13 let $a_i = 1$, $m_i = 2$, and $b_i = 3$ hours. Using equations 12 and 13, then the tasks 11, 2, 23, and 4 have an expected time of $E(t_i) = 3$ hours and a variance of $\sigma^2{}_i = 4/9$ hour^2. Likewise, the tasks 21 and 13 have expected time of $E(t_i) = 2$ hours and a variance of $\sigma_i^2 = 1/9$ hour^2.

Now the tasks 11 and 21 are executed in parallel so that the critical path will be based on the task with the larger value of $E(t_i)$, namely task 11. There are, therefore, two possible paths which are the critical path, namely 11,2,13,4 and 11,2,23,4. For the task path 11,2,13,4 the value of expected total time is 11 hours, whereas for the task path 11,2,23,4 the value of the expected total time is 12 hours. Thus, the latter is the critical path. Now suppose that the window T = 13 hours. For values of E(t) = 12 and $\sigma^2{}_t = 16/9$, use of the normal distribution tables leads to $P(t<T) = 0.77$.

3.3.5 Information Utility. Information utility may be considered in terms of the overlap between the information a sender sends and the information a receiver wants. For two particular blocks in a system let:

K_{si} = set of information items sent by block i

and

K_{wj} = set of information items wanted by block j

Then the number of information items in the common set of information items sent by block i and wanted by block j is:

$$N_{ij} = g(K_{si} \cap K_{wj}) \qquad \text{eq(16)}$$

where g is a count of the common set items. Define:

N_i = number of information items in set K_{si}

and

N_j = number of information items in set K_{wj}

Then information utility I_{ij} for blocks i and j is given by:

$$I_{ij} = \frac{N_{ij}}{N_{ij} + N_j + N_{ij}} \qquad \text{eq(17)}$$

It is the fraction of common information items out of all information items. The overall measure of information utility I can be defined as the average over all combinations of i and j, namely:

$$I = \sum_{i \neq j} \sum \frac{I_{ij}}{M} \qquad \text{eq(18)}$$

where

M = number of component combinations in the system that are transferring information

The value of I can vary between zero and one with one being the desired value.

The Core example in Figure 2 can again be used as an illustration. Figure 4 shows the transfer of information items associated with the block diagram of Figure 2a. The letters a through h indicate individual information items. Each line connection between blocks shows the information items sent (to the left) and information items wanted (to the right). These items do not necessarily correspond as shown between blocks 2 and 13 and blocks 2 and 23. In the case of blocks 2 and 13, e and f are sent while e, f and g are wanted so that:

K_{s2} = e,f $\qquad N_2 = 2$

K_{w13} = e,f,g $\qquad N_{13} = 3$

$K_{s2} \cap K_{w13}$ = e,f $\qquad N_{2,13} = 2$

Therefore

$$I_{2,13} = \frac{2}{2+3-2} = \frac{2}{3}$$

By the same procedures:

$$I_{11,2} = 1 \qquad I_{13,4} = 1$$

$$I_{21,2} = 1 \qquad I_{23,4} = 1$$

$$I_{2,23} = 1/2$$

The overall measure of information utility is:

$$I = (1 + 1 + 2/3 + 1/2 + 1 + 1)/6 = 0.86$$

3.4 System Analysis Synopsis

Analysis begins with the drawing of the block and Petri net diagrams. The first diagram is important to delineating how a system is physically organized and the second is important to delineating how a system operates. Lack of either diagram can cause untold problems in system analysis. Associated with each diagram is a set of system characteristics which acts as the bridge between the diagrams and performance measures. The following is a listing of the system characteristics and diagrams which influence each performance measure.

Perf Measure	Sys Char	Diagram
Connectivity	K, N	block
Operability	$p_o(x_i)$	Petri
Survivability	$p_s(y_i)$	Petri, block
Timeliness	a_i, m_i, b_i	Petri
Information Utility	N_i, N_j, N_{ij}, M	block

These performance measures are not independent and, in general, tradeoffs exist between them. For example, while P_o and P_s tend to increase with additions of multiple backup task paths, timeliness tends to decrease due to encountering longer critical time paths, i.e., paths with larger E(t). Thus, there is generally a tradeoff of P_o and P_s with P(T) for various system configurations. Similarly, even when values of hardkill and softkill are equal, differences in P_o and P_s will exist due to collocation of blocks and multiple tasking. As a result, one configuration will rarely dominate in terms of all the performance measures. Therefore, use of the expected mission utility E(u) is necessary to resolve conflicting values of performance measures associated with various system configurations.

4. OTHER INPUT DATA

In order to carry out the full analysis of each problem, besides the block and Petri net diagrams, we have to specify the density function f($\underline{x}$) for the performance measures. In the absence of more precise knowledge, we assume that f($\underline{x}$) is uniformly distributed over a range $\underline{x}_a$ to $\underline{x}_b$ determined by specifying values of the following system characteristics namely:

1. Upper and lower values of K
2. Upper and lower values of p_o
3. Upper and lower values of p_s
4. Values of a_i, m_i, and b_i for each transformation
5. Upper and lower values of N_i, N_j, and N_{ij} for each block combination

In addition, we need upper and lower values of the time window T. From the overall data we are able to:

1. Compute upper and lower values of connectivity C
2. Derive the relationships for P_o and P_s
3. Determine upper and lower values of P_o and P_s
4. Determine the critical (time) paths associated with the Petri net diagrams
5. Compute the upper and lower values of P(T)
6. Compute upper and lower values of information utility I

From this information we are able to compute E(u) in equation 2. Complete illustrations of these calculations and the eventual results of ranking design options are given in Edmonds [8].

5.0 SUMMARY

This paper extends existing system evaluation methodology by illustrating the development of five C^3 system performance measures from block and Petri net diagrams. It is essential that the main issues regarding system configuration be captured in a set of block and Petri net diagrams. As the example demonstrates, such diagrams display the differences in system configurations in terms that are readily analyzed. Without such diagrams we cannot become specific regarding component placement, information transfer, task sequence or mutual backup. It is essential that we start with such diagrams to delineate the differences between options.

The raw data required by this analysis include the following categories:

1. Block and Petri net diagrams
2. System characteristics (viz., K, N, $p_o(x_i)$, $p_s(y_i)$, a_i, m_i, b_i, N_{ij}, N_i, N_j, and M)
3. Performance requirements (viz., T)
4. Primitives that reflect the utilities of evaluators (viz., F)

Prioritization of system configurations is normally carried out by examining the effect of the first two categories of primitives while holding the last two categories constant.

The Core example was used to give a detailed demonstration of the mathematical aspects of the methodology. It is expected that the methodology will be helpful in ongoing C^3 projects such as the development of the US Army Maneuver Control System.

6.0 REFERENCES

[1] J.G. Karam, Effectiveness Analysis of Evolving Systems, LIDS-TH-1431, Laboratory for Information and Decision Systems, MIT, January 1985.

[2] J.G. Karam and A.H. Levis, "Effectiveness Assessment of the METANET Demonstration," Proceedings of the 7th MIT/ONR Workship on C^3 Systems, LIDS-R-1437, Massachusetts Institute of Technology, December 1984.

[3] D. Tabak and A.H. Levis, "Petri Net Representation of Decision Models," Proceedings of the 7th MIT/ONR Workshop on C^3 Systems, LIDS-R-1437, Massachusetts Institute of Technology, December 1984.

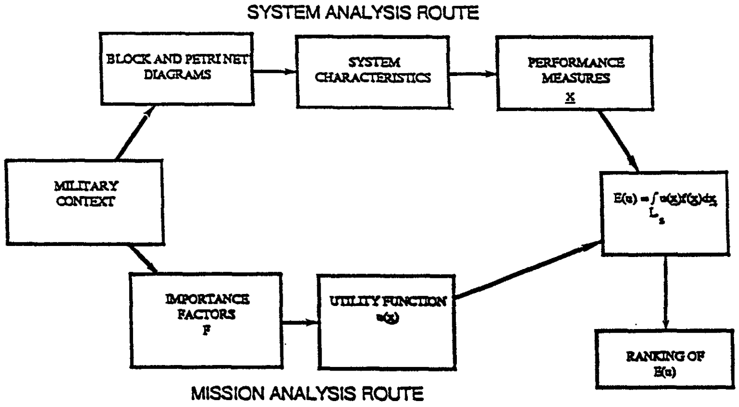
[4] J. Bick, D. Edmonds, W. Hise, J. Hannan, G.Miller, P. Palmore, E. Preston, M. Simmons, F.Snyder, B. Truett, D. Wolfe, Specification for the Maneuver Control Functional Segment Subsystem (U), Volume I, WP-84W00004-01, The MITRE Corporation, 1 February 1984.

[5] D.R. Edmonds, "Evaluating C^3 System Survivability Based on Reliability and Network Analysis Theory," Proceedings of the 5th MIT/ONR Workshop on C^3 Systems, LIDS-R-1267, Massachusetts Institute of Technology, December 1982.

[6] D.R. Edmonds, G. Emami and W.B. Hise,Computational Methodology for Evaluating ADP Operational Survivability, MTR-81W00152, The MITRE Corporation, September 1981.

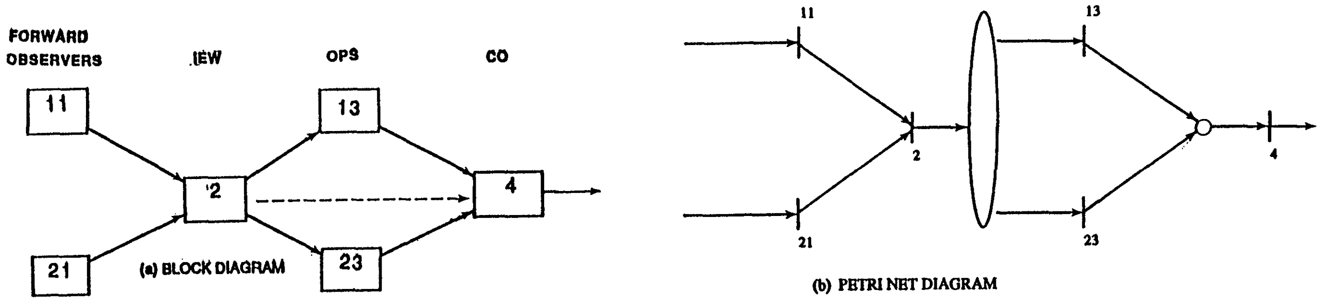
[7] F.S. Hillier and G.J. Lieberman, Operations Research, Second Edition, Holden-Day, Inc., San Francisco, 1974.

[8] D.R. Edmonds, Methodology for the Effectiveness Evaluation of Alternative C^3 System Configurations, MTR-85W00261, The MITRE Corporation, October 1985.



SYSTEM ANALYSIS ROUTE
BLOCK AND PETRI NET DIAGRAMS
SYSTEM CHARACTERISTICS
PERFORMANCE MEASURES $\underline{x}$
MILITARY CONTEXT
$E(u) = \int_{L_s} u(\underline{x})f(\underline{x})d\underline{x}$
IMPORTANCE FACTORS F
UTILITY FUNCTION $u(\underline{x})$
RANKING OF E(u)
MISSION ANALYSIS ROUTE

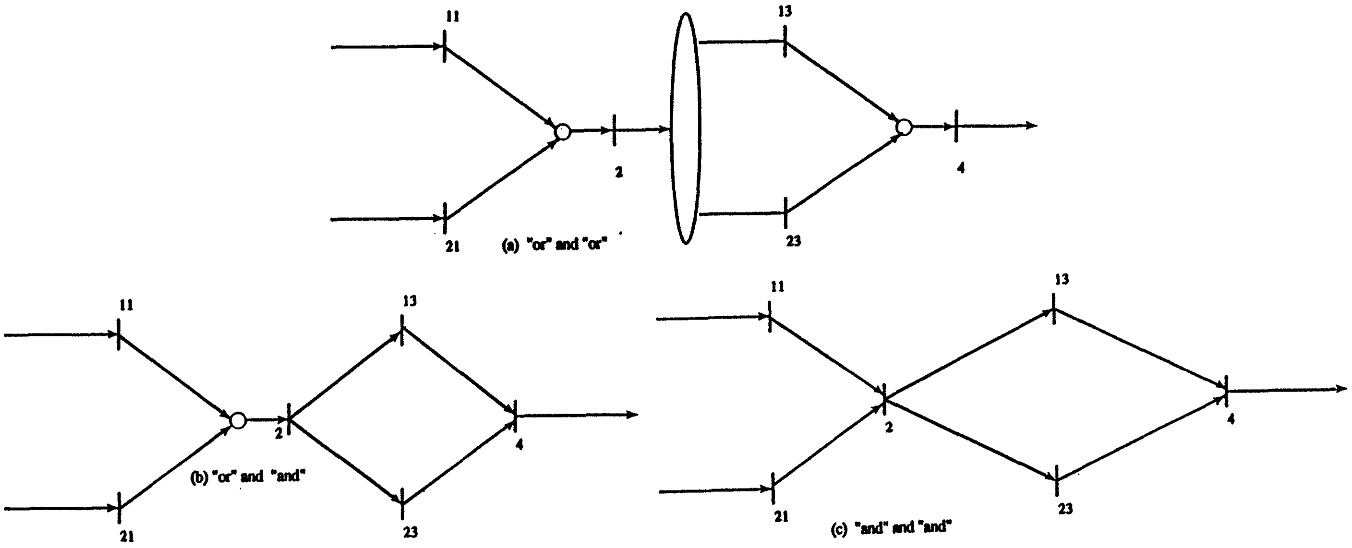
FIGURE 1
OVERVIEW OF THE METHODOLOGY



(a) BLOCK DIAGRAM

(b) PETRI NET DIAGRAM

FIGURE 2
DIAGRAMS FOR CORE EXAMPLE



(a) "or" and "or"

(b) "or" and "and"

(c) "and" and "and"

FIGURE 3
OTHER POSSIBLE PETRI NET DIAGRAMS

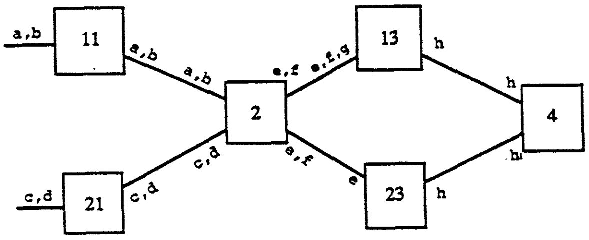


FIGURE 4
TRANSFER OF INFORMATION ITEMS IN CORE EXAMPLE

A REPORT ON A PROJECT ON MULTIPLE CRITERIA DECISION MAKING, 1986

Stanley Zionts

School of Management

State University of New York at Buffalo

Abstract

The purpose of this paper is to present an update on a project on Multiple Criteria Decision Making, a project which began in the early to mid-seventies. Jyrki Wallenius and I developed a method for solving multiple objective linear programming problems. Our work has continued and grown in four directions:

1. Multiple Objective Linear Programming
2. Multiple Objective Integer Programming
3. Choosing among Discrete Alternatives involving MultipleCriteria
4. Multiple Decision-Maker Multicriteria Decision Making

In this paper we describe the current status of each branch and emphasize recent developments.

1. Introduction

The purpose of this paper is to present an update on a project on multiple criteria decision making begun by Jyrki Wallenius and myself over ten years ago at the European Institute for Advanced Studies in Management in Brussels. The project started as a way of finding a multiple objective linear programming method that would work better than those tested by Wallenius' (1975a,b). Wallenius and I did a substantial amount of work on the problem and came up with such a method (Zionts and Wallenius, 1976). Wallenius' (1975a) thesis, one of the first outputs of that project, comprises a rather significant piece of research in the multiple criteria area. Since that time our work has continued to evolve. Wallenius and I have worked together on a great deal of it; some of it has involved students and other faculty colleagues. In presenting this update, I shall make every effort to accurately attribute (and reference) each piece of research to the appropriate person(s). Though I have tried not to omit any references or acknowledgments, or both, I apologize in advance for any inadvertent omissions.

2. The Background of Our Approaches

Our methods all involve the use of pairwise comparisons and tradeoff evaluations by a decision maker who chooses between selected pairs of alternatives and/or evaluates tradeoffs. His choices reveal a preference to which we locally fit a linear function (and in some instances a quadratic or higher order function). The use of a linear function is not meant to imply the decision maker's underlying utility function is linear. In many (perhaps most) cases it is not. Further, since the linear function is not unique and we may find different functions for different problems with the same decision maker (even if he is acting in a consistent manner with a well-behaved utility function), we downplay the importance of the function we identify. Rather than use this function as a utility function, we use it to identify good (and hopefully optimal) alternatives, and present these to the decision maker in helping him to make a decision. Our approach is in contrast to the utility assessment models which assess the utility function directly by an interview process, come up with a utility function, and then rank order the alternatives for further consideration by the decision maker. The latter methods, developed and maintained by Keeney, Raiffa among others (see for example, Keeney and Raiffa, 1976), come up with a utility function that could conceivably be transferred from one decision situation to another. Though our function could be transferred from one situation to another, that is not our intention; we have no evidence to suggest that such a procedure is even worthwhile for our methodological framework.

Our work has four major branches:

1. A multiobjective linear programming method that assumes an underlying unknown pseudoconcave utility function.

2. A multiobjective integer linear programming method that assumes an underlying unknown quasiconcave utility function.

3. A multiobjective method for choosing among discrete alternatives. Here we assume an underlying unknown quasiconcave utility function.

4. A multiperson, multiobjective method for handling problems of type 1 and type 3.

We will describe each of the branches and attempt to highlight what we consider to be the most interesting developments. In this section we have introduced and overviewed what we present in this paper. In section three, we briefly review our original method. In section four, we overview recent results in the four branches of our research and explore how they build on the earlier work. This includes both the theory we have developed and what practical experience we have had to date. We then draw conclusions.

3. Review of our Original Multiple Objective Linear Programming Model

Our method is a method for multiple objective linear programming which uses weights. This development is based on Zionts and Wallenius (1976). In our framework a numerical weight (arbitrary initially though generally chosen equal) is chosen for each objective. Then each objective is multiplied by its weight, and all of the weighted objectives are then summed. The resulting composite objective is a proxy for a utility function. (The manager need not be aware of the combination process.) Using the composite objective, we solve the corresponding linear programming problem. The solution to that problem, an efficient or nondominated solution is presented to the decision maker in terms of the levels of each objective achieved. Then the decision maker is offered some trades from that solution, again only in terms of the marginal changes to the objectives. The trades take the form, "Are you willing to reduce objective 1 by so much in return for an increase in objective 2 by a certain amount, an increase in objective 3 by a certain amount, and so on?" The decision maker is asked to respond either yes, no, or "I don't know" to the proposed trade. The method then develops a new set of weights consistent with the responses obtained, and a corresponding new solution. The process is then repeated, until a presumably "best" solution is found.

The above version of the method is valid for linear utility functions. However, the method is extended to allow for the maximization of a general but unspecified concave function of objectives. The changes to the method from that described above are modest. First, where possible the trades are presented in terms of scenarios, e.g., "Which do you prefer, alternative A or alternative B?" Second, each new nondominated extreme point solution to the problem is compared with the old, and either the new solution, or one preferred to the old solution is used for the next iteration. Finally, the procedure terminates with a neighborhood that contains the optimal solution. Experience with the method has been good. With an many as seven objectives on moderate-sized linear programming problems (about 300 constraints) the maximum number of solutions is about ten, and the maximum number of questions is under 100.

4. Recent Work on Our Methods

In this section we consider the methods in the order outlined in Section 2. We do this in a series of subsections, one for each method.

4.1 The Multiple Objective Linear Programming Method

Our earliest computer codes incorporated only the linear version of our method. To implement the concave and then the pseudoconcave extensions of the method we made several changes to the method. First we partitioned the questions to be asked of the decision maker into six groups. The first three groups consist of questions that are efficient with respect to old responses; the second three groups consist of questions that are efficient, but not with respect to old responses. Within each set of three groups we have a partition of efficient questions. The first group of efficient questions are those that lead to distinctly different solution vectors of objective functions. Those questions are asked as scenarios, i.e., "Which do you prefer, solution A or solution B?" Operationally, distinctly different solutions are not well defined. We define the term in a working context to mean some specified minimum difference in at least one criterion. The second group of efficient questions include those that lead to solutions that are not distinctly different. We present those questions as tradeoffs "If you are at solution A, would you like to decrease the first objective by so much in return for increasing the second objective by so much, etc.?" The third group of efficient questions are those corresponding to distinctly different solutions that were not preferred to the reference solution by the decision maker. These are presented to the decision maker again, but this time as tradeoffs. The decision maker proceeds through the sequence of questions. Whenever a group of questions is completed and the decision maker has liked a tradeoff or an alternative, a new set of weights (consistent with responses) is generated and the corresponding solution that maximizes the weighted objective function is found. The procedure continues from that solution. If the decision maker does not prefer any alternative to the reference solution (and does not like any tradeoff), then the reference solution is optimal. If the decision maker likes one or more tradeoffs, and if an extreme point solution preferred to the reference solution cannot be found, we know that there are solutions preferred to the reference solution. To find them we cannot restrict ourselves to corner point solutions, and some other procedure must be used. This presentation is of necessity brief; some steps have been simplified for exposition. For more details on these changes see Zionts and Wallenius (1983).

Deshpande (1980) has developed an approach for finding optimal solutions when the procedure terminates at an extreme point solution that is not optimal. Deshpande's procedure begins at the termination point of the above procedure, with the tradeoff vectors liked by the decision maker, takes their (vector) sum, and has the decision maker engage in a binary search over the feasible range in the facet of the convex polyhedron of the solution space. If the most preferred solution is at the end point of a range, the procedure tries moving to an adjacent facet. Otherwise, it chooses an orthogonal direction on the facet. The procedure continues until an optimal solution is found. Deshpande's procedure is rather cumbersome, and does not look very promising. It has not been extensively tested. In some almost completed work, Steven Breslawski, a Ph.D. student, and I are investigating how close the best extreme point found is to true optimal solution for a class (or several classes) or assumed nonlinear utility functions. Our contention is (and early results show) that the solutions are generally close. Of course, we have defined close in an operational manner. Steve has also carried out extensive tests on the different types of questions asked of a decision maker in the procedure. Several of the question types are not very fruitful in practice, and Steve has heuristic procedures that ignore some of these. Our results are that using such a procedure, an optimal solution is obtained even though an optimal solution cannot be guaranteed.

Should the user not be satisfied with the best extreme point found, we also have a simple procedure for finding a sequence of solutions to the best extreme point solution found. The intent of this procedure is to improve upon the best extreme point solution found rather than to necessarily find an optimal solution. The efficacy of this procedure has yet to be tested. sessions. Brelawski is also exploring the use of cone dominance (see Section 4.3) and the effect of a most-consistent set of weights (see the section on integer programming) on the linear programming problem. Breslawski has been able to construct the constraint set bounding the set of cone dominated solutions. This involves writing the cone of dominated solutions, and performing simplex pivots on it. The result is a set of polyhedra that makes up the constraint set. By dropping constraints as part of an algorithm, he finds the desired constraint set.

As far as the application of our method is concerned, we have programmed the method and have used it in several different forms. We and various organizations have prepared and adapted programs to solve different problems. Our most current program is a FORTRAN mainframe program that uses Marsten's XMP (1979) package for the linear programming routines. This code is being used for the tests we are conducting. It is available for use with the XMP package, and may be adapted for use with other linear programming packages.

Many problems have been solved with variations of the method. After solving a number of small problems for which a linear utility function was assumed, we worked on a long-range planning problem for S. A. Cockerill, a large integrated Belgian steel company. The problem involved a time-phased investment model with four objectives, 143 constraints, and 248 variables. See Wallenius and Zionts (1976) for further information. Our method has also been used by the Philips Company in Eindhoven, The Netherlands to solve a strategic management problem involving seven objectives. A form of the general concave method has been used for national economic planning in Finland. Four objectives were used (for more information, see Wallenius, Wallenius, and Vartia (1978). In addition, another rather large problem has been solved in various forms by several decision makers at the Brookhaven National Laboratory and at the United States Department of Energy in Washington. That model is an energy planning model with six objectives and several hundred constraints (for more information, see Zionts and Deshpande (1978)).

The computational requirements for this method involve essentially one linear programming solution for each setting or revision of weights. The maximum number of setting for each revision of weights has always been less than ten in our applications. The total number of questions asked of the decision maker has always been less than 100, and generally less than 50.

Professors Pekka Korhonen of the Helsinki School of Economics and Jyrki Wallenius of the University of Jyvaskyla together with their colleague J. Laakso have developed an approach that combines our approach with some others and uses computer graphics. The approach begins by finding a nondominated solution. Next an improving direction is determined by interaction with the decision maker. Part of the Zionts-Wallenius procedure may be used for this purpose. Then the improving direction is projected onto the efficient frontier to form an efficient curve which is presented to the decision maker by several superimposed objective function graphs on a computer monitor. By using a cursor, the decision maker moves along the efficient curve and identifies a most-preferred point. If a new point is found along the curve, we find a new improving direction. If not, we construct a convex cone containing a number of nondominated feasible directions from the current point. Each is considered as an improving direction. If none leads to a new solution via the preceding steps, then the current solution is optimal. Otherwise, the procedure continues. The method may be thought of as a combination of the Zionts-Wallenius (1976) method as well as the Geoffrion, Dyer, and Feinberg (1972) and Wierzbicki (1980) methods. The method's use of computer graphics is particularly interesting. For further information see Korhonen and Laakso (1985, 1986).

4.2 The Multiple Objective Integer Linear Programming Method

Shortly after Wallenius and I published our initial paper (Zionts and Wallenius, 1976), I proposed an extension of our procedure for solving multiple criteria integer programming problems (Zionts, 1977). I was not very optimistic regarding the approach, but I felt that the idea was nonetheless interesting and worth reporting. Not long thereafter a Ph.D. student in Industrial Engineering, SUNYAB, Bernardo Villarreal began to work with Mark Karwan, a professor in the Industrial Engineering Department and me. He proposed doing a thesis on multiple criteria integer programming. The thesis (Villarreal, 1979) developed several methods including an improved version of what I had proposed. In extensive testing, Villarreal had found that, although the methods had done well for small problems, the method did not appear to have promise for problems of reasonable size. Another reference on the method is Villarreal, Karwan, and Zionts (1980). The procedure uses a branch-and-bqund approach after first solving the corresponding noninteger linear programming problem. The procedure is like the standard branch-and-bound method, except that it uses some special approximations in the branch-and-bound process.

Shortly after Villarreal completed his thesis, we began to rethink a few ideas developed therein. We developed several improvements to the method. Our work was evolutionary in that once we made an improvement and it seemed worthwhile under test conditions, we incorporated it into our procedure. It is possible that we may be missing even better options in the design of our method. However, the results obtained were sufficiently positive in our judgment as to delay a more systematic study.

Based on our original work (Villarreal, Karwan, and Zionts (1980), and Villarreal (1979), we envisioned two improvements to the method of Villerreal's thesis:

1. Eliminating response constraints on weights that become redundant.

2. Finding a "most consistent" or "middle most" set of weights rather than any consistent set of weights.

We shall now consider both of these in detail.

4.2.1 Eliminating Redundant Constraints

Constraints on weights are generated by decision-maker responses and are used for:

a) determining which tradeoff questions are efficient;
b) determining a feasible set of weights;
c) determining whether a decision-maker's response to a comparison of two solutions can be inferred from previous responses.

Because the set of constraints on the weights grows with the number of responses, and because the feasible region shrinks, we believed that a number of constraints became redundant. Although it is not possible to predict what fraction (or number) of constraints are redundant in general, we know for certain that with two objectives, there can be at most two nonredundant constraints. (By normalizing the weights without loss of generality using for example $\lambda_1 + \lambda_2 = 1$, we may express all constraints in terms of one λ, e.g., λ_1. Our weight space is therefore unidimensional, and we may have at most two nonredundant constraints: an upper and a lower bound on λ_1.) With three objectives, there is no limit on the number of nonredundant constraints. Nonetheless, we did believe that a substantial portion of the constraints for more than two objectives became redundant. Accordingly, we altered our computer program. After each constraint was added to the set of constraints on weights, we use the Zionts-Wallenius (1982) method for identifying redundant constraints to eliminate whichever constraint or constraints have become redundant.

4.2.2 Finding a Most-Consistent Set of Weights

In our multicriteria integer programming procedure we need to find a new set of feasible weights whenever the decision maker likes an efficient tradeoff offered by the procedure. Previously, we found an arbitrary solution to the set of inequalities on the weights using the dual simplex method. The resulting set of weights, an extreme point of the feasible region of the λ-space to be sure, was generally quite close to the previous set of weights. As a result, the new solution or node in the branch and found procedure was "close" in terms of objective function value to the old one. We proposed changing the procedure to find a most-consistent or middle-most set of weights by maximizing the minimum slack of the constraints on the weights. The idea of choosing a most-consistent or middle-most set of weights is analogous to using a binary search procedure in a single dimensional search. The questions generated thereby are intended to decrease the set of feasible weights quickly.

The results of these simple changes were very good. We ran two sample sets of 0 - 1 multicriteria linear programming problems. The times to solve problems having two objectives, four constraints, and twenty variables decreased from 57.7 seconds to 10.8 seconds of CPU time; similarly, the times to solve a problem having three objectives; four constraints, and ten variables decreased from 23.7 seconds to 8.6 seconds (of CPU time). A further improvement was to use various heuristics to identify a good initial integer solution. The empirical results of these improvements were to further reduce CPU times by an additional factor of three. We also examined such questions as the relation between computation time and various problem parameters and the effect on problem solution times of the initial set of weights. With relatively minor changes in our approach, we have brought our approach to the threshold of computational feasibility. For further information, see Karwan, Zionts, Villarreal, and Ramesh (1985).

In a more recent dissertation, R. Ramesh (1985), working together with Karwan and me, has made considerable strides in further improving and developing the methodology. First, Ramesh has experimented with several different branch-and-bound strategies. Second, he has worked extensively with cone dominance in eliminating solutions and branches in integer programming (see next section). In addition, he has done extensive work with respect to the bicriteria problem. Research papers based on the dissertation are now available. Most of Ramesh's empirical work was with the bicriteria problem; we are now working on the more general problem.

4.3 A Multicriteria Method for Choosing Among Discrete Alternatives

About the time that we published our first article on the Multiple Objective Linear Programming problem, I had an informal conversation with a colleague not at all familiar with multiple criteria models. He asked whether the linear programming approach could be used to solve discrete alternative problems -- for example, the choice of a house by a prospective buyer. My first reaction was that he did not understand the difference between a linear programming problem and a problem of choice among a set of alternatives. To this day, I am still not sure whether my initial reaction was correct. However, on reflection I saw that the approach could indeed be used. That ideas evolved into an approach published in Zionts (1981).

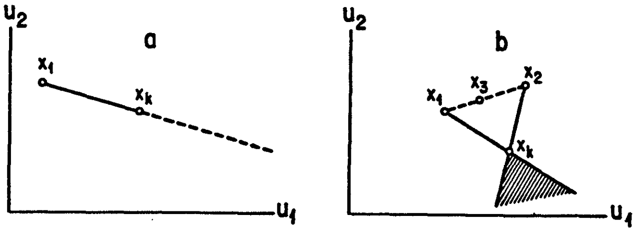
That paper contained an approach modeled after the linear programming method. We begin with an arbitrary set of weights and choose the alternative that maximizes the weighted sum of objectives. We then determine (using the convex hull of all alternatives) the set of adjacent efficient solutions to this solution. We then ask the decision maker whether he prefers the current reference alternative to each of the adjacent efficient solutions. If he prefers the current reference alternative or cannot answer for certain in every case, we terminate the process.

If he prefers at least one of the adjacent solutions, we find a new set of weights consistent with his responses and a new reference alternative that maximizes that weighted sum of objectives. We then ask the decision maker whether or not he prefers the new reference alternative to a preferred solution adjacent to the old reference alternative. If so, we repeat the above process using the new reference alternative in place of the old. Otherwise, we use one of the preferred adjacent efficient solutions in place of the old reference alternative solution. We repeat until no adjacent extreme point solution is preferred to the current reference alternative. In presenting the final subset of alternatives to the decision maker, we order them in terms of the final set of weights found.

Some early applications were made to about four or five different decision problems, each involving a decision maker in a choice situation. All involved a very small number of alternatives (less than fifteen); so the value of the method was not clear, although in each case the method seemed to do well. Zahid Khairullah (1981) in his doctoral thesis did some exhaustive test comparisons of this and other methods.

In a sequel paper (Korhonen, Wallenius, and Zionts, 1984) we provided several improvements over the previous method. First, we weakened the assumption of the underlying utility function to be quasiconcave and increasing. Second, we use a convex cone based on decision-maker choices to eliminate some of the alternatives.

The idea of elimination of alternatives based on convex cones is simple, yet powerful. Consider the following diagram:



We construct a cone from a set of alternatives in the following way: Given two (or more) alternatives: x_1 $x_2,\ldots$, and x_k. Let x_k be the least preferred of the set of alternatives. A cone is constructed proceeding from x_k away from x_1 (and $x_2,\ldots$ and convex combinations thereof). Any alternatives in the cone or dominated by points in the cone may be shown to be less preferred to x_k and may, therefore, be eliminated. By reference to the above diagram this includes any points on or dominated by the cross-hatched line or the shaded cone. The cone dominance concept has proven to be very powerful; we are using it in other branches of our work. We also use some of the developments from other aspects of our work (such as most-consistent multipliers) for the discrete alternative problems.

Murat Koksalan (1984) (See Koksalan, Karwan, and Zionts (1983, 1984, 1986)) made extensive use of cone dominance and solved randomly-generated problems to test variations of the discrete alternative methods. We have found that two-point cones (e.g., the cone involving points x_1 and x_k in the above diagram) seem to work better than cones involving more than two points, although further investigation of this is continuing. We have also used quadratic approximations to the utility function to accelerate the solution process. Finally, we have worked with both cardinal and ordinal criteria. A heuristic for reducing the number of questions asked of the decision maker works well in the case of ordinal criteria. In a relatively refined version of the method that considers both cardinal and ordinal both cardinal and ordinal criteria, generally fewer than twenty pairwise comparisons are asked of the decision maker to identify an optimal solution.

In some follow-up research, H. W. Chung has extended some of the concepts developed by Koksalan and others. One of his most significant developments is the generalization of the function used to locally approximate the utility function. Chung has developed a way of choosing among various nonlinear functions. He also has made extensive use of dominance cones. He also has developed a clever way of making use of "I don't know" responses that promises to be of use in all branches of our work. He tries to treat the "I don't know" response as a response of perfect indifference, but relaxes that as necessary to obtain a feasible set of weights. Chung's dissertation should be completed in the near future.

We have worked with the Greater Buffalo Board of Realtors, the organization of real estate agencies in Buffalo to apply the above method. In that work we are developing our method for use in helping prospective home buyers in the choice of a house. Prospective buyers indicate their criteria and constraints and then choose among pairs of houses. Upon completion of the question session, a number of houses are presented for the home buyer to consider. Then as appropriate they may use the method to generate additional alternatives.

4.4 A Multiple Decision Maker, Multicriteria Model

The fourth problem in the area on which we have worked is a multiple criteria problem in which there are two or more decision makers. This problem is extremely difficult compared to the earlier problems because of the lack of problem resolution if the different members of the group cannot reach an agreement. Our approach (Korhonen, Wallenius, Zionts, 1980) considers both the multiobjective linear programming as well as the multiobjective discrete alternative problem. Both are based on our earlier methods. The procedures work similarly. First each member of the group uses the method by himself to identify his most preferred solution. Then we work with the group to find a group solution. Suppose that λ^k represents the vector of weights for member k of the group. To start the group process, we compute $\bar{\lambda} = \Sigma\lambda^k/d$ (the average of the members' weights) where d is the number of members of the group. Using $\bar{\lambda}$ as a weighting vector we find the corresponding efficient solution (called the reference solution). We then identify efficient solutions adjacent to it and ask the group to choose between the reference solution and an adjacent solution. We use a procedure similar to the corresponding single decision-maker procedure to find a sequence of better solutions. So long as the group members are able to agree on what constitutes an improved solution, the procedure works well. If they are unable to agree on a solution, the procedure does not work. (For more information see Korhonen, Wallenius, and Zionts, 1980.) The procedure has been used in several situations with students at Purdue University and at the University of Jyvaskyla, Finland. See Korhonen, Moskowitz, Wallenius, and Zionts (1986). The situation involved a labor-management negotiation problem where students representing labor and students representing management had to come up with a mutually satisfactory labor contract. We experimented in this study to find out whether our structured approach based on the discrete alternative method seemed to be better than an unstructured form of bargaining. In every instance each group used both forms of bargaining. In the first set of experiments (at Purdue), the structured approach seemed to do slightly better than the unstructured approach, although the results were not significantly different. Further, there seemed to be a learning effect; that is, whichever method was used second was usually preferred. An improved set of instructions for the methods were used for the second study at the University of Jyvaskyla, Finland. The results were a bit more conclusive. There the structured approach was found superior to the unstructured approach. For more information see Korhonen, Moskowitz, Wallenius, and Zionts (1986). More work will be undertaken in the multiple decision maker model; we believe it to be an extremely important problem.

Conclusion

In this paper I have briefly summarized our recent progress in multiple criteria decision making. Work is continuing along four major directions: a linear programming method; an integer programming method; a discrete alternative method; and a multiple decision maker method. Even though we have worked on this project for several years, we continue to be excited and challenged by the problems that remain. The problems stimulate us to overcome them. The field is a challenging one, one that contains many rich areas for further research.

REFERENCES

1. Deshpande, D., 1980, "Investigations in Multiple Objective Linear Programming-Theory and an Application," Unpublished Doctoral Dissertation, School of Management, State University of New York at Buffalo.

2. Geoffrion, A. M., Dyer, J. S., and Feinberg, A., 1972, "An Interactive Approach for Multicriterion Optimization, with an Application to the Operation of an Academic Department," Management Science, 19, pp. 357-368.

3. Karwan, M. H., Zionts, S., Villarreal, B., and Ramesh, R., 1985, "An Improved Interactive Multicriteria Integer Programming Algorithm," in Haimes, Y. Y., and Chankong, V., Decision Making with Multiple Objectives, Proceedings Cleveland Ohio, 1984, Vol. 242, Lecture notes in Economics and Mathematical Systems, Springer-Verlag, Berlin, pp. 261-271.

4. Keeney, R. L. and Raiffa, H., 1976, Decisions with Multiple Objectives: Preferences and Value Tradeoffs, John Wiley and Sons, New York.

5. Khairullah, Z., 1981, "A Study of Algorithms for Multicriteria Decision Making," Unpublished Doctoral Dissertation, School of Management, State University of New York at Buffalo.

6. Koksalan, M. M., 1984, "Multiple Criteria Decision Making with Discrete Alternatives," Unpublished Doctoral Dissertation, Department of Industrial Engineering, State University of New York at Buffalo.

7. Koksalan, M. M., Karwan, M. H., and Zionts, S., 1983, "An Approach for Solving Discrete Alternatives Multiple Criteria Problems involving Ordinal Criteria," Working Paper No. 571, School of Management, State University of New York at Buffalo.

8. Koksalan, M. M., Karwan, M. H., and Zionts, S., 1984, "An Improved Method for Solving Multiple Criteria Problems Involving Discrete Alternatives," IEEE Transactions on Systems, Man and Cybnernetics, 14, pp. 24-34.

9. Koksalan, M. M., Karwan, M. H., and Zionts, S., 1986, "Approaches for Discrete Alternative Multiple Criteria Problems for Different Types of Criteria," IIE Transactions, 00, pp. 000-000.

10. Korhonen, P., Moskowitz, H., Wallenius, J., and Zionts, S., 1986, "A Man-Machine Interactive Approach to Collection Bargaining," Naval Research Logistics Quarterly, 23, pp. 000-000.

11. Korhonen, P., Wallenius, J., and Zionts, S., 1980, "Some Thoughts on Solving the Multiple Decision Maker/Multiple Criteria Decision Problem and an Approach," Working Paper 414, School of Management, State University of New York at Buffalo.

12. Korhonen, P., Wallenius, J. and Zionts, S., 1984, "Solving the Discrete Multiple Criteria Problem Using Convex Cones, Management Science, 30, pp. 1336-1345.

13. Korhonen, P., and Laakso, J., 1985, "On Developing a Dual Interactive Multiple Criteria Method - An Outline" in Haimes, Y. Y., and Chankong, V., (eds.) Decision Making with Multiple Objectives Proceedings, Cleveland, Ohio, 1984, Vol. 242, Lecture Notes in Economics and Mathematical Systems, Springer-Verlag, Berlin, pp. 272-281.

14. Korhonen, P., and Laakso, J., 1986, "A Visual Interactive Method for Solving the Multiple Criteria Problem," European Journal of Operational Research, 24, pp. 000-000.

15. Marsten, R. E., 1979, "XMP: A Structured Library of Subroutines for Experimental Mathematical Programming," Technical Report No. 351, Management Information Systems, University of Arizona, Tucson.

16. Ramesh, R., 1985, "Multicriteria Integer Programming," Doctoral Dissertation, Department of Industrial Engineering, State University of New York at Buffalo, 2 Vols.

17. Villarreal, B., 1979, Multicriteria Integer Linear Programming, Doctoral Dissertation, Department of Industrial Engineering, State University of New York at Buffalo.

18. Villarreal, B., Karwan, M. H., and Zionts, S., 1980, "An Interactive Branch and Bound Procedure for Multicriteria Integer Linear Programming," in Fandel, G. and T. Gal (eds.), Multiple Criteria Decision Making: Theory and Application Proceedings, 1979, Number 177, Lecture Notes in Economics and Mathematical Systems, Springer-Verlag, Berlin, 1980, pp. 448-467.

19. Wallenius, J., 1975a, Interactive Multiple Criteria Decision Methods: An Investigation and an Approach, Ph.D. Dissertation, The Helsinki School of Economics, Helsinki.

20. Wallenius, J., 1975b "Comparative Evaluation of Some Interactive Approaches to Multicriterion Optimization," Management Science, 21, pp. 1387-1396.

21. Wallenius, H., Wallenius, J., and Vartia, P., 1978, "An Approach to Solving Multiple Criteria Macroeconomic Policy Problems and an Application," Management Science, 24, pp. 1021-1030.

22. Wallenius, J. and Zionts, S., 1976, "Some Tests of an Interactive Programming Method for Multicriterion Optimization and an Attempt at Implementation," in H. Thiriez and S. Zionts (eds.), Multiple Criteria Decision Making, Jouy-en-Josas, France, 1975, Springer-Verlag, Berlin, pp. 319-330.

23. Wierzbicki, A., 1980, "The Use of Reference Objectives in MultiObjective Optimization," in Fandel, G. and Gal, T. (eds.), Multiple Criteria Decision Making: Theory and Application, Springer-Verlag, Berlin.

24. Zionts, S., 1977, "Integer Linear Programming with Multiple Objectives," Annals of Discrete Mathematics, 1, pp. 551-562.

25. Zionts, S., 1981, "A Multiple Criteria Method for Choosing Among Discrete Alternatives," European Journal of Operations Research, 7, pp. 143-147.

26. Zionts, S. and Deshpande, D., 1978, "A Time Sharing Computer Programming Application of a Multiple Criteria Decision Method to Energy Planning -- A Progress Report," In S. Zionts (ed.), Multiple Criteria Problem Solving, Proceedings, Buffalo, NY, 1977, Springer-Verlag, Berlin, pp. 549-560.

27. Zionts, S. and Wallenius, J., 1976, "An Interactive Programming Method for Solving Multiple Criteria Problem," Management Science, 22, pp. 652-663.

28. Zionts, S. and Wallenius, J., 1980, "Identifying Efficient Vectors: Some Theory and Computational Results," Operations Research, 28, pp. 788-793.

29. Zionts, S. and Wallenius, J., 1982, "Identifying Redundant Constraints and Extraneous Variables in Linear Programming," Chapter 3 in Karwan, M., Lotfi, V., Telgen, J., and Zionts, S., Redundancy in Mathematical Programming, Springer-Verlag, Heidelberg.

30. Zionts, S. and Wallenius, J., 1983, "An Interactive Multiple Objective Linear Programming Method for a Class of Underlying Nonlinear Utility Functions," Management Science, 29, pp. 519-529.

FORMAL DEFINITION OF INTEROPERABILITY PROTOCOLS

P. D. Morgan and N. Innes

Scicon Ltd.

Summary

The selection and application of formal verification techniques for use on OSI protocol definitions is discussed, the development of the resultant logic program is outlined, and conclusions are drawn on the effectiveness of the approach.

Introduction

The increasing size and complexity of modern systems is leading to increased interest in the application of formal proof methods to the tasks of specification development and verification. The Project Independent Virtual Data Base (PIVDB) project represents one such system, which is aimed at handling C3 interoperability problems in the European land and land/air environments.

The objectives and approach of the PIVDB project were discussed by Dr Fowler at the 7th MIT/ONR Workshop [1]. In summary, the PIVDB design is based on the ISO Open Systems Interconnection (OSI) model [2]; like the OSI model, it is expressed in semi-formal natural language, and is thus subject to the inherent problems of lack of rigour, and unsuitability for automatic verification. This paper discusses the selection, development and use of a formal method for the task of testing the completeness and consistency of this body of protocols. The PIVDB and its protocols are considered only in so far as they impact on the development and use of the verification method.

The requirements to be satisfied by the verification method were that it should be complete; that it should be machine executable; that it should be consistent with the related work on the extension of the protocols and of the PIVDB language; that the method should be generally accessible; and that it should be amenable to non technical explanation and comprehension. This last criterion has proven incompatible with the requirements for machine execution and transportability.

The selected approach is based on the use of formal theorem proving techniques. The relevant elements of the OSI model were expressed in First Order Predicate Logic (FOPL), and were then developed to provide an executable interpreter to exercise the protocols. The proving of the protocols involves their execution by this interpreter to demonstrate their mutual consistency, and to identify the attainable system states.

The Selection of the Methodology

The initial task in this study involved the selection of a suitable methodology. A range of potentially suitable formal and semi-formal methods were identified from the literature, from past experience, and from the STARTS Guide (Software Tools for Application to large Real Time Systems) [3]; these were assessed for suitability against the previously mentioned criteria.

The semi-formal methods which were considered included both requirements analysis and structured design systems; and techniques based on the use of strongly typed specification languages. The design systems examined had significant advantages in ease of comprehension, but were considered inadequate with respect to rigour, to the level of automation which could be provided, and to the extent to which they matched the task. The requirements of strong typing, generics, and wide availability, ruled out all specification languages except Ada [Ada is a registered trademark of the US DoD AJPO]; the problems which were seen with the use of Ada were those of the relative difficulty in interpreting the operation of the program, the complexity of the language, development and modification timescales, and the scale of the facilities required.

The formal methods considered were based on First Order Predicate Logic in both its general and Horn clause forms. The closed nature of the PIVDB protocols, and the implicit requirement that instantiations of the OSI protocols should be monotonic, render the use of FOPL an obvious alternative. The features of FOPL [4] which make it suitable for use on the protocol proving task are:

a) The power of the Horn clause form of FOPL; this has the advantages that it is sufficient for the representation of any computable function [5], and that it supports an efficient Resolution algorithm.

b) The existence of a widely available means of automating the proof system through the use of Prolog, a programming language based on Horn clause grammar. FOPL and the Edinburgh Prolog syntax described by Clocksin and Mellish [6] have come close to being international standards; thus definitions couched in these formalisms would be widely accessible and long lasting, if not readily intelligible. In addition, effective Prolog systems are available for a wide range of hardware, from micro to mainframe computers, a factor which significantly reduced the development costs.

c) The flexibility of Prolog as a development tool; the declarative organisation of a Prolog program, in terms of separate clauses, makes development, modification, and fault finding much quicker than with existing procedural languages, rendering it an invaluable tool for the exploration of problems of this type.

It was concluded that there is no perfect solution to the problems of protocol verification; but that, in combination, FOPL and Prolog meet more of the requirements than other alternatives. In particular, it was considered that the unusual combination of rigour and flexibility obtainable with this combination renders it well suited to this class of problem. It is acknowledged that the problems of intelligibility and run-time efficiency exist, but it is considered that they are outweighed by the advantages.

Prolog

Horn clause logic differs from normal FOPL, in that it contains neither universal nor existential quantifiers ($\forall$ and $\exists$), the latter being replaced by Skolem variables/constants. Well formed formulae consist of a number of clauses associated by 'and' ($\land$) connectives, where each clause may contain any number of negated terms (literals), and no more than one unnegated literal, all associated by 'or' ($\lor$) connectives [7].

Prolog is based on Horn clause logic and the efficient Resolution algorithm which the Horn clause form makes possible. The translation of well formed formulas from the FOPL syntax into Horn clauses is executed using the standard logical equivalences, such as de Morgan's Laws; in consequence, a logic program may be used to translate well formed formulas into Horn clauses, and thence into the Prolog format [6]. It should be noted that Prolog clauses do not take the conventional Horn clause format, as the equivalence of

$\sim P \lor \sim Q \lor R$ and $P \land Q \Rightarrow R$
--- P and Q imply R

is used to obtain the Prolog clause

R :- P, Q. --- R if P and Q.

Normal Prolog execution makes use of a depth-first, backward chaining, search strategy, in which the literals in a clause are handled from left to right. This strategy ensures that, if a solution is available, then, subject to run-time and capacity constraints, it will eventually be found. Failure on a particular path causes the system to back-track to the nearest unexplored branch, and continue the search.

In addition to the logic formalisms, Prolog also contains a number of non-logical features, which are used to improve execution efficiency. The most important are the 'cut' mechanism (!), which permits user control of back-tracking; and operators which aid in the construction and manipulation of functors and clauses. These mechanisms are essential for efficient operation, but add to the problems of program interpretation and validation, as they have no direct logical equivalents.

Development of the Verification System

Overview

It was recognised at the outset that the system used for the verification of the protocols must itself be verifiable. The potential infinite regression of proof systems was avoided by using the initial OSI model definitions [2] as the axioms of the proof system. This approach reduced the task of determining the validity of the PIVDB protocols to that of demonstrating that the clauses representing the rules of the OSI proof system were well formed formulae; and that the PIVDB protocols were derivable theorems within this OSI system.

The implementation approach was aimed at aiding the normal user by limiting the need for experience in the use of Prolog. This led to the development of a system in which Protocols were held as lists and simple clauses, and the transient Data Units and (N)-state definitions were held as list structures. This is analogous to the common Expert System structure of inference engine (OSI Interpreter), rule base (protocols) and data base (Data Units and (N)-state definitions). Within this approach, the proving of the protocols corresponds to the demonstration that the execution of the protocols by the OSI interpreter results in the required (N)-state transformations.

Basic OSI Definitions

The development of the OSI Interpreter commenced with the representation of the elements of the OSI model in FOPL, and their translation from this into Prolog. The specification of the Open System formed the start point for all the remaining definitions, and is represented by the Function Symbol 'system(S)', where the variable symbol S takes the identity of the system under consideration.

In certain cases the representation of the OSI rules proves self-evident; for example, where the ISO definition of an open system implies the existence of (N)-layers containing subsystems of the same rank [ISO 5.2.1.2]; also, the existence of any (N)-layer above layer 1 indicates the existence of subordinate (N)-layers:

$$\forall S:[\mathrm{system}(S) \Rightarrow \exists N:[\mathrm{equal}(N,1) \lor (\mathrm{lesseq}(N,7) \land \exists M:[\mathrm{sum}(M,1,N) \land \mathrm{layer}(S,M))] \Rightarrow \mathrm{layer}(S,N)]. \tag{1a}$$

this translates to the two Prolog clauses:

```
layer(S,N(S)) :- system(S), N(S) = 1.
layer(S,N(S)) :- system(S),          (1b)
     N(S) =< 7,M is N(S)-1,layer(S,M).
```

The replacement of N by the Skolem variable N(S) implies that the layers present are dependent on the system identity, S; thus the 'Relay Open Systems', which in protocol terms represent separate systems, contain no layers above layer 3 [ISO 7498 page 18 end of para 6.1 and page 19]. The 'tail recursive' form of the expression is adopted due to its economic use of the stack during evaluation [4].

The most direct FOPL representation for the OSI Data Units was not so immediately obvious; the selected representation was based on the use of Prolog data handling techniques. Data Units provide the means of transfering control information between adjacent and peer layers; this information is organised into data packets for transfer, where individual packets may be segmented or concatenated to suit the transmission requirements of the connected systems (neither segmentation nor concatenation were addressed in the protocol validation).

The (N)-Protocol-Control-Information [ISO 5.6.1.1] is a typical Data Unit; (N)-PCIs are exchanged between systems to organise the (N)-services required for the initiation, maintenance and termination of (N)-connections. In the logic program they have been represented as a layer designator, a type identifier, and a set of protocol control parameters:

```
∀S:[∀N:[layer(S,N) ⇒ ∀Type:[type(Type)
⇒ ∃P1:[∃P2:[* * [∃PM:[parameter(P1)   (2a)
∧ parameter(P2) ∧ * * ∧ parameter(PM)
⇒ pci(N,Type,P1,P2,* *,PM)].

pci(N,Type,P1(Type,N,S),P2(Type,N,S),
      P3(Type,N,S)):-
      layer(S,N), type(Type),
      parameter(P1(Type,N,S)),          (2b)
      parameter(P2(Type,N,S)),
      * * * * * * * * * * * *
      parameter(PM(Type,N,S)).
```

The handling of these data structures is simplified by the use of the standard Prolog predicates 'functor' and 'arg' [6]. The predicate 'functor(T,F,NA)' defines a data structure T, which is based on the functor F, and which has the number of arguments (arity) NA. The predicate 'arg(NA,T,A)' provides the means of accessing the NAth argument (A) of data structure T. Using these predicates, clause (2b), was rewritten:

```
pci_type(Name) :- functor(Name,pci,M+2).
source_layer(Name,N) :- pci_type(Name),
            arg(1,Name,N).
type(Name,Type) :- pci_type(Name),
            arg(2,Name,Type).           (2c)
parameter_1(Name,P1) :- pci_type(Name),
            arg(3,Name,P1).
* * * * * * * * * * * * * * * * * * * * *
parameter_M(Name,PM) :- pci_type(Name),
            arg(M+2,Name,PM).
```

This representation has the same logical interpretation as that used in (2b), but has the advantages that it aids specification of the PCI, and simplifies access to the various parameters contained in it.

One significant deficiency was found during the translation of the OSI protocol definitions into the logic formalism; this was the absence of an explicit (N)-state definition. This deficiency was remedied by the addition to the model of a definition of the (N)-state in terms of the system, the layer, and the current parameter settings (ranging between 1 and M), resulting from the previous protocol operations:

```
∀S:[∀N:[system(S) ∧ layer(S,N)
⇒ ∃P1:[∃P2:[* * [∃PM:[parameter(P1)    (3)
∧ parameter(P2) ∧ * * ∧ parameter(PM)
⇒ state(S,N,P1,P2,* *,PM)]
```

This was translated into an analogous form to (2c). The various (N)-states of the systems under investigation were held in semi -permanent form as data-base entries; thus they did not require explicit transfer between predicates, but could be accessed as required.

Protocol Representation

The verification of the protocol definitions does not require detailed representation of the manipulation of data transfered between the OSI Data Structures; consequently, the operation of individual protocols has been represented in terms of the transfer of data tokens between the various Data Units and (N)-states. Individual protocols are implemented in two elements; clauses containing the protocol selection criteria, and data structures defining the data transfers controlled by the protocol. The effect of branching within a protocol is represented by the addition of separate clauses, each of which operates as a separate sub-protocol.

The entry of a new protocol requires the definition of the selection criteria in the form of Prolog clauses, and the production of the data structures and lists which define the PCIs and ICIs generated by the protocol. The initial version of the system provided little assistance to the user in executing these tasks, making an understanding of Prolog essential. Subsequent work has been aimed at producing a dialogue interface, which will remove this requirement, and will also reduce the incidence of errors in data entry.

Protocol Operation

The selection and operation of an (N)-protocol must be uniquely determinable for any combination of state (St) and control information [ISO 5.7.2]. The interaction between an (N)-layer and an (N+1) or (N-1)-layer is effected by the exchange of Interface Data Units (IDUs) containing (N)-interface-control-information (ICI), (N)-protocol-control-information (PCI),and (N)-service- data-units (Data) [ISO 5.6.1]. The manner in which protocols provide (N)-services has been encapsulated in a single well formed formula containing predicates related to the two elements of the protocol discussed above. Branching has been accommodated in the predicate 'select_protocol', leaving the predicate 'execute_protocol' responsible for the processing of data and control in accordance with the selected protocol (Pr):

```
VS:[VN:[VICI:[VData1:[VSt1:[∃Pr:[∃PCI:[
∃Data2:[∃St2:[∃IDU:[select_protocol(S,N,
      Pr,IDU,ICI,PCI,Data1,Data2,St1)
⇒ ∃N2:[∃ICI2:[execute_protocol(S,N,   (4a)
      Pr,IDU,ICI,PCI,Data2,St1,St2)
⇒ provide_service(S,N,ICI,Data1,IDU)].

provide_service(S,N,ICI,Data1,IDU) :-
      select_protocol(S,N,Pr,Pr_list,
            ICI,PCI,Data1,Data2,St1),(4b)
execute_protocol(S,Pr,Pr_list,IDU,
            ICI,PCI,Data2,St1).
```

The above well formed formula is incomplete in that it requires extension to include the recursion termination conditions.

Control of Execution

As previously mentioned, the normal mode of Prolog operation employs a backward chaining, depth first search; the inherent concurrency of protocol operation renders this mode of operation unsuitable for the verification task; in particular, it prevents examination of problems such as deadlock. This problem was overcome by the use of a control clause which modified the flow of control, causing the system to emulate a forward chaining, breadth first search; this control clause provided the goal for the Prolog system to satisfy; the IDUs, which provided the local verification goals, were held in a first in last out list (IDU_list); this caused the protocol operations to be handled in a pseudo-concurrent manner:

```
control_data_flow(IDU_list1,IDU_list2) :-
      IDU_list1 = [IDU1|IDU_list3],
      IDU1 = [S,ICI_list,Data],
      ICI =.. ICI_list,                (5)
      destination(ICI,N),
      write('IDU transfered to layer '),
      write(N), write(' of system '),
      write(S), write(' contains :'), nl,
      write('ICI : '), write(ICI), nl,
      write('Data Unit : '),write(Data),
      provide_service(S,N,ICI,Data,IDU2),
      include(IDU_list3,IDU2,IDU_list4),
      !,control_data_flow(IDU_list4,
            IDU_list2).
```

This clause contains two Prolog list formats, [X, Y, Z], a list with members X, Y, and Z, and [X|Y], a list with head X and tail Y; in both cases the members of a list may themselves be lists; the functor '=..' (univ) provides the means of converting the list ICI_list into the functor ICI. The predicate 'destination' returns the destination layer N from inspection of ICI, and the predicate 'insert(IDU_list3,IDU2,IDU_list4)' adds IDU2 to the tail of IDU_list3 to form IDU_list4. For clarity, the clause has been simplified by omission of the termination conditions.

System Operation

The verification operation is initiated by the user defining the connectivity and initial (N)-states of the interoperating systems, and the initial Interface Data Units (IDU_list1 in (5)); these may be input manually, or may be restored from some previous run. The output of the verification system provides a trace of the (N)-states of the interconnected systems, the protocols being executed, and the IDUs being transfered; in addition, non-fatal fault conditions (missing or preinstantiated parameters) are identified by parameter without halting execution.

Results

The translation of the PIVDB protocols into the form required for execution, and the initial trials of the verification system, revealed a number of relatively trivial inconsistencies and omissions, but no spectacular failures. The primary problems encountered during the exercise were:

a) Lack of consistency in the storage and use of the various parameters; for example, where a parameter is provided as an input to a layer, but is neither stored nor transfered to another layer.

b) Lack of consistency in the naming of the various parameters; this led to uncertainty and possible error in associating parameters.

c) The inherent ambiguity of natural language when applied to system specification; this provided another possible source of errors in interpreting the specifications.

d) The unfriendliness of the user interface, which slowed down the process of protocol entry.

It was thought that there might be a problem related to the size of the state-space to be searched; this was not found to be significant, as a very limited number of protocols are accessible at any stage in the verification process. This inherent limitation of the search space, in combination with the control of recursion between (N)-layers (corresponding to the isolation of successive (N)-layers inherent in the OSI model), has resulted in relatively efficient execution.

Conclusions

The combination of FOPL and Prolog has proven to be effective in handling bounded problems of this type. It is considered that the combination of rigour, traceability, and speed of development would not have been achieved with a procedural language. An additional benefit has been the source-code portability of Edinburgh syntax Prolog; this permitted development and execution to be carried out at different sites, on different machines.

The faults discovered in the protocols confirmed the inadequacy of natural language and manual methods of consistency checking to the specification and development of large systems. It is concluded that the logic programming approach provides an effective means of producing such specifications in a verifiable, executable form, and that the use of natural language should be restricted to explanation and description of the specification. It is also considered that such an approach would prove valuable to the proving of the final system.

Acknowledgements

This paper originates from work carried out under contract to the Procurement Executive, UK Ministry of Defence, and with the assistance of Dr Fowler and the other members of the Scicon PIVDB team. The authors gratefully acknowledge permission from the Ministry to reproduce the information contained in this paper.

References

1] Fowler, M; 'Interoperability - the Virtual Data Base Approach'; Proceedings of the 7th MIT/ONR Workshop on C3 Systems; LIDS-R-1437, LIDS, MIT; Dec 1984.

2] 'Information processing systems - Open Systems Interconnection - Basic Reference Model'; International Organization for Standardization; Ref. No. ISO 7498-1984(E).

3] The STARTS Guide, Part 1, Section 2; 'Choosing Software Tools and Methods'; UK Department of Trade and Industry; 1983.

4] Hogger, C. J.; 'Introduction to Logic Programming'; Academic Press Inc; 1984.

5] Andreka, H. and Nemeti, I.; 'The generalised completeness of Horn predicate logic as a programming language'; Research Report 21; Dept. of Artificial Intelligence; Univ. of Edinburgh; 1976.

6] Clocksin, W. F. and Mellish, C. S., 'Programming in Prolog', Second Edition, Springer-Verlag, New York, 1984.

7] Nilsson, N. J., 'Principles of Artificial Intelligence', Springer-Verlag, New York, 1982.

AN EXPERIMENTAL COMMAND SYSTEM FOR FORCE LEVEL ANTI-SUBMARINE WARFARE

Dr C J Gadsden
Admiralty Research Establishment
Portsdown, Portsmouth
Hampshire PO6 4AA, England

Introduction

The Admiralty Research Establishment's Command Systems Laboratory (CSL) at Portsdown has recently been investigating the provision of machine support to Naval Command during anti-submarine warfare (ASW). The research programme has resulted in the creation of a command system work station implemented on the real time hardware and software simulation facilities within the Laboratory. This paper is concerned with how the design of the work station was developed and also with some of the novel decision support features it contains.

In this research activity, emphasis has been placed on supporting the tactical planning and monitoring functions performed by senior (Flag level) Command. At this point in the command chain, tactical decisions must be made concerning the deployment of an integrated force containing air, surface and sub-surface assets. A paper delivered at the UK's Bournemouth C3 conference in 1985 [1] described some of the results of a command systems analysis study performed as part of this research programme. A key output of the analysis was a model for particular aspects of command, control and co-ordination. This model is called the C Process. A key component of the model is a representation of the mechanism both for seeking the solutions to complex planning problems and for monitoring the effectiveness of a solution following its implementation. The C Process has been employed as the underlying theoretical framework in the development of an experimental command system simulation within the Laboratory. This paper initially describes how the transition was made between the abstract model of C2 and a working command system simulation. The balance of the paper then reviews some key aspects of the system design.

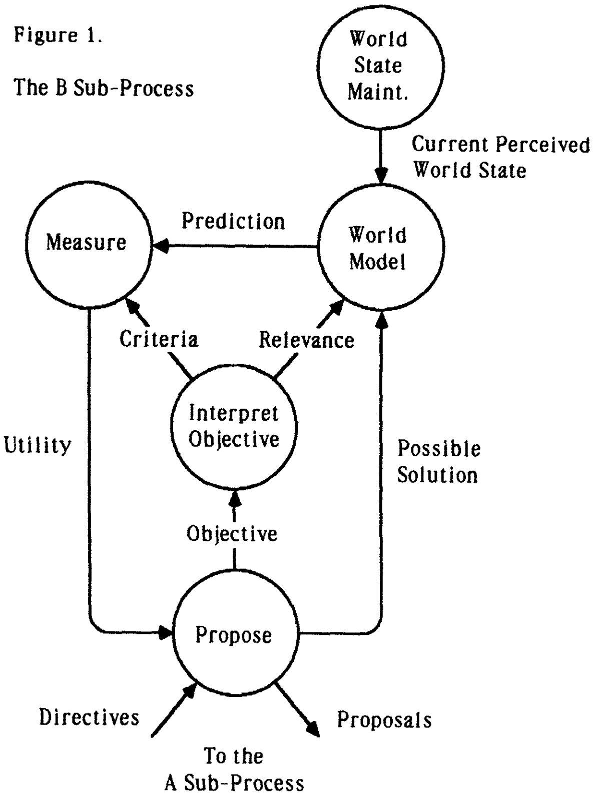
Systems Analysis Considerations

The analysis work reported in [1] employed a semi-formal design methodology based on MASCOT, a method extensively used in the UK for the description and implementation of real time systems, in particular, computer systems. MASCOT syntax concentrates on the functions performed in a system, the flow of stimuli and data as well as the need for actual data storage within the system. Since a strict methodology (containing semantic and syntactic elements) has been employed there is a coherent link between the abstract C Process model and the structure of the implementation. This provides a substantial advantage during system design, implementation and subsequent modification.

The problem solving component of the C Process is the Proposal Assessment Loop. The basic form of the loop is shown qualitatively in figure 1. PROPOSE creates a tentative solution in the light of previous experience, WORLD MODEL predicts the consequences of the proposed solution whilst MEASURE estimates the utility of these consequences in the context of the problem. INTERPRET OBJECTIVES is responsible for specifying not only the effectiveness criteria by which the utility is gauged, but also for tailoring the WORLD MODEL to cover only those aspects of the World of relevance to the problem under consideration. These 4 functions are collectively called the B Sub-Process. The significance of this division is made clear below.

Figure 1.

The B Sub-Process



The remaining functional areas of the C Process, shown in figure 2, are concerned with handling the flow of messages into and out of the command system represented by the C process, as well as tasking the Proposal Assessment Loop. This part of the C Process is collectively called the A Sub-Process. Incoming messages are received by the ANALYSE function which maintains an up-to-date account of the current and future tasks for the C Process. Particular aspects of tasking that concern ANALYSE are the degree of delegated authority and the form of any imposed constraints. Such information is then made available to the DIRECT function. This key function is responsible for understanding how a problem facing the command system can be addressed as a set of parallel or sequencial sub-problems. A key feature of the C

Process concept is that each A Sub-Process can be attached to a number of parallel B Sub-Processes. In addition, a command system can be represented by either just one very complex C Process or, more usefully, by a hierarchic tree of C Processes, the structure of which reflecting that of the command chain it represents. The C Process concept is therefore capable of capturing in a rigorous form the complexity of many aspects of military command systems, in particular those concerned with planning and monitoring.

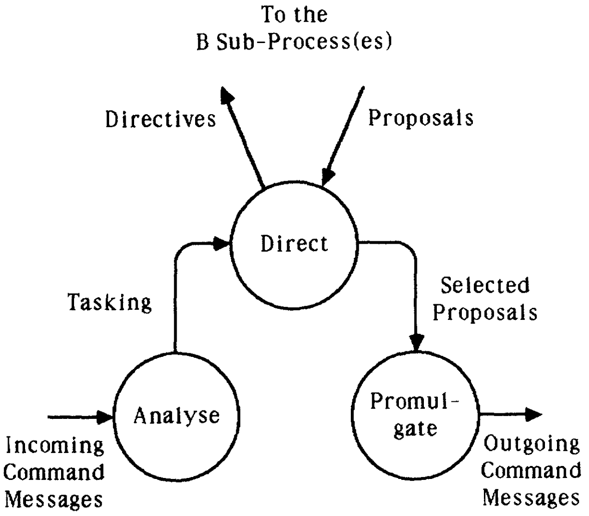


Figure 2. The 'A' Sub-Process

Also shown in figure 1 is another key functional area, the WORLD STATE MAINTAINER (WSM). This function is responsible for providing the Command System's perception of its tactical and geographic environment. Although this is not discussed further here, the internal structure for the World State Maintainer could be similar to that of the C Process. both mechanisms employ a model of the World and its behaviour to predict a current or future World State. When the WSM provides its perception of the World, called the Current Perceived World State (CPWS), the Assessment Loop can exercise its other role of World Monitoring. Instead of proposed solutions being submitted to the World Model, information is drawn from the CPWS to form the input to a prediction process in the same manner as during proposal assessment.

The Command System Design

If a command system is represented by a C Process structure, the efficiency of the working system can be enhanced by providing machine support for selected functions, leaving other functions to be performed in a human-centred way. It should be noted that functional responsibility cannot, and perhaps should not be considered as simply being delegated to machinery. It is likely that the human components of a command system represented by a C Process mentally perform most, if not all the functions present in the structure. Machinery simply provides assistance in selected areas. This assistance can, of course, be both massive and critical.

During the design process for the ASW command system, a formal methodology was developed for deciding whether a function should be human centred or heavily supported by machinery. The methodology was based on the perceived ability of humans and classical computing engines to perform a primitive set of operations which, together, permit a command system to function. Any consideration of knowledge-based computer systems was precluded by the terms of reference of the research. The human and machine attributes were as follows:

Human Attributes:

Comprehension and interpretation of free formatted and unforeseen information

Capability fòr heuristic operation based on ill defined information

Creative problem solving involving flexibility and innovation

Assignment of subjective values to alternatives

Machine Attributes:

Accurate retention and retrieval of literal and numerical information

Fast and accurate application of algorithmic rules to logical and numerically quantifiable terms

Consistent execution of repetitive operations

Except for one function, ANALYSE, it was determined that each C Process function required attributes drawn solely from just one of the above two sets. ANALYSE was determined to be a hybrid function and hence was functionally decomposed until 'pure' functions were determined.

Thus, in order to permit the timely, accurate and quantitative prediction of tactical effectiveness both prior to and during the implementation of a plan, it is necessary that the C process functions WORLD MODEL and MEASURE are machine based. The PROPOSE and INTERPRET OBJECTIVE functions must, however, remain human responsibilities since their heuristic and complex nature precludes any current machine-based solution performing even a substantial fraction of the tasks facing them. The machine aspects of the Proposal Assessment Loop are henceforth collectively called the Outcome Predictor (OP) Facility. This OP Facility is the first of four basic components, or building blocks, of command system machine capabilities that can be assembled together to create a command work station, or C-Station.

An examination of the tasks facing the functions in the A Sub-Process lead to the conclusion that the message handling and distribution functions, ANALYSE and PROMULGATE can be almost entirely machine-based whilst DIRECT must remain a human responsibility. The machine aspects of the A Sub-Process are collectively called the Command Message Processor. This is the second C-Station component. As mentioned above, one aspect of ANALYSE remained human oriented. This was necessary to cater for the handling of informal messages, such as unformatted signals, and human speech which are not easily machine interpretable.

Formally defining the data needs, including its flow, within the C Process, along with the man/machine division of functions, clearly specifies the responsibilities, location and, to some extent, form of the man-machine interface within the command system. If the bi-directional man-machine dialogue is to be designed and controlled in a coherent manner it is desirable that it be controlled and monitored by a central facility, the 'Man-Machine Interface' (MMI) which can be considered to be the third C-Station component. The last component is determined by consolidating all the stored data with the command system within a unified data base. This store of data, called the Common Data Facility, is the fourth and last necessary machine component of a workable C-Station. A minimal command work station configuration is shown in figure 3. Human components of the command system interact with its machine based aspects via MMI. Similarly, the C Station can communicate with other elements of the command chain, or its resources, via the Command Message Processor.

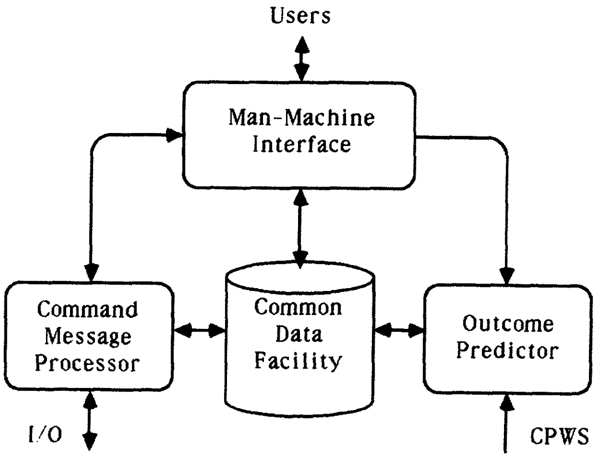


Figure 3 C-Station Components

A command system logical design, based on the above modular concept, has been pursued in a rigorous top down fashion from the abstract level of the C Process to a level of detail where design of the physical implementation could proceed. The main concern of this paper is the set of Outcome Predictor features which are described at the lowest levels of the OP Facility design. Space does not permit a review of the top down design structure but a key design concept is the provision of stratified modelling. Individual models of platforms and equipment can be exercised within the OP but, in addition, model characteristics can be aggregated into a composite structure. An example of this is where an ASW force Commander represents a task group of many platforms by a single entity for the purposes of his tactical planning. Composite object performance would be limited by the performance envelopes of its component parts.

Key Features of the Outcome Predictor

Many of the design features stress and quantify the probabilistic nature of the many types of information facing Command during anti-submarine warfare.

The set of OP features that exist in the system design were developed from the command system functional analysis and a review of the current and anticipated ASW problem. The features exist as an integrated network of software modules, sharing a common database with the rest of the command system. This configuration arose naturally from the use of the C Process concept and is in contrast to the method of construction of many 'stand-alone' decision support aids previously supplied to the UK's Royal Navy and which suffer from little or no integration with other on-board systems. The constraints of this paper preclude an adequate review of all aspects of the Outcome Predictor, instead, the remainder of this paper will concentrate on those features directly concerned with planning and monitoring the deployment of ASW sensors.

1. Platform/Equipment Command Language - A machine based planning system must permit the unambiguous expression of a tactical plan in a form that the machine can interpret. A command language has been developed that captures the basic requirements of platform and equipment control. This includes, for platforms, the need to perform a rendezvous with other platforms and an event driven logical structure within which conventional commands are assembled.

2. Platform Kinematics - Mathematical models of platform performance envelopes and fuel consumption are provided. The models are generic so that any platform, or group of platforms can be accommodated by changing parameter values. This feature permits the compilation and testing of machine interpretable movement plans.

3. Electromagnetic/Acoustic Sensor Models - Generic models for both active and passive sensors are provided. Models provide a time dependent spatial distribution, $G_i(x,y,t,dt)$ of detection probability per unit time interval, dt for the i'th sensor. If a set of sensors are of interest to Command, the net detection field, G can be estimated and presented graphically. The net field calculation currently assumes statistical independence of sensor data and is estimated from:

$$G(x,y,t,dt) = 1 - \prod_i (1 - G_i(x,y,t,dt)) \qquad (1)$$

In general, G will be dependent on sensor platform behaviour, oceanographic and meteorological conditions.

4. Search Residue - For this, and other features, the concept of Submarine Occupancy has been developed. Occupancy is a scalar field, $S(x,y,t)$ providing a net estimate, using many data sources, of the expected density of submarines at any point in an area of interest. It should be noted that S is not a probability density and is not bounded above by unity. If the area of interest is surveyed by a composite sensor field G at time t, then after interval dt the best estimate of a new Occupancy field, now relating to undetected submarines, is $S(x,y,t+dt)$, where:

$$S(x,y,t+dt) = S(x,y,t)(1 - G(x,y,t,dt)) \qquad (2)$$

Equation 2 assumes that submarines move negligibly over period, dt. If a model exists for submarine motion at time, t, over dt, equation 2 can be re-expressed as:

$$S(x,y,t+dt) = M(t,dt).S(x,y,t)(1 - G(x,y,t,dt)) \qquad (3)$$

M is an algebraic operator representing the submarine motion model. The Occupancy distribution obtained from equation 3 is called the Search Residue field. This field is an estimate of the number of submarines likely to exist undetected at point x,y following surveillance for time, dt. By evaluating equation 3 iteratively, S may be evaluated for any time, t. Two motion models have been extensively examined in this calculation. In the Brownian Motion model, submarines are assumed to travel at constant speed, v, but to randomly re-orient their direction after period, dt. In this case, M takes the form of a convolution with a spreading function, F(v,dt). F can be anisotropic and represents probable submarine migration over dt. Equation 3 becomes:

$$S(x,y,t+dt) = F(v,dt).conv.S(x,y,t)(1 - G(x,y,t,dt)) \quad (4)$$

With this model and an isotropic F, a precise submarine datum would evolve into a Normal distribution of Occupancy about the datum position in ocean regions where G=0. By contrast, with the Linear Motion model, submarines are assumed to move at constant speed without course change. To model this, submarine migration and Occupancy is resolved into n equi-spaced components. Equation (4) becomes:

$$S(x,y,t+dt) = \sum_{n} F_n(v,dt).conv.S_n(x,y,t)(1 - G(x,y,t,dt)) \quad (5)$$

In this model, if G were zero, a precise datum would evolve into an expanding annulus, equivalent to a furthest-on circle. Figure 4 indicates how the Search Residue field develops in the presence of non-zero G and a uniform initial distribution of S. In figure 4, the contour shape for time=0 conforms, very schematically, to the probability of detection contours per some unit time for a specified target submarine. At the centre of the pattern is a naval towed array frigate symbol indicating the location of the sensor. The presence of the detector field has left its imprint on the previously uniform Search Residue field (equation 2). As time progesses, knowledge about submarine presence begins to build up due to repeated sensor glimpses into the area of interest. Knowledge also decays due to the sensor platform moving away from areas that it had previously surveyed, permitting the ingress of previously undetected submarines. In general, for a force-wide multi-sensor surveillance operation, S will be of a highly complex form and very difficult to predict without machine assistance. When presented graphically, it is expected that Search Residue will be a powerful tool for evaluating the quality of an ASW search operation. The Occupancy field created portrays directly to Command the level of knowledge that could be achieved concerning the submarine threat, following the implementation of a surveillance plan. The precise nature of such a prediction depends on the form of the initial Occupancy field. This will be briefly discussed in a following section that covers the World State Maintainer.

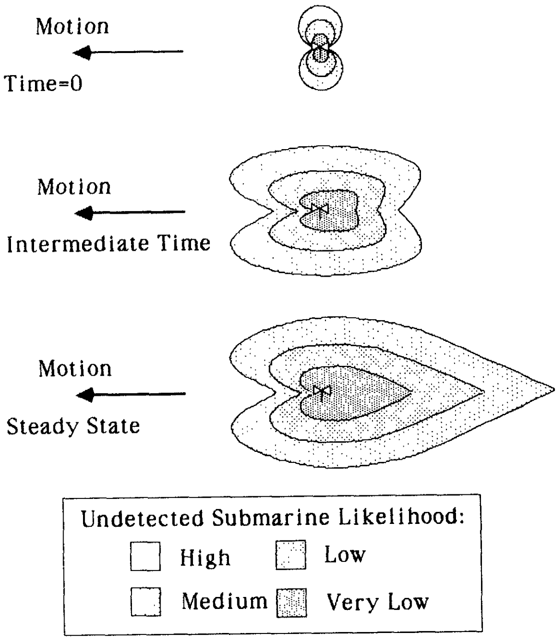


Figure 4 The Search Residue Concept

5. Optimum Route Prediction - This, and the following two features use the techniques of dynamic programming to provide Command with decision support. Many of the concepts in these 3 sections were originally developed for the US Department of the Navy for planning aircraft operations in the vicinity of a Strike Fleet [2] though they have been found to be more generally applicable. A review of dynamic programming is beyond the scope of this paper and the reader is directed to the above reference and standard texts for more information. Dynamic programming provides a method for developing satisficing solutions to problems amenable to mathematical expression. Cost/benefit criteria must be specified for gauging optimality and these are expressed algorithmically in an 'objective function'. The Optimum route feature is concerned with predicting the routes that hypothetical and knowledgeable enemy submarines might follow from a specified start location to reach any/all points in an area of interest. A current implementation searches for routes that minimise cumulative detection probability in the presence of sensor fields whilst also minimising distance travelled (a trade off is usually necessary). The significance of this feature is that it provides a key input to the following two areas of the OP. In reality, an enemy submarine would not have access to any or all precise information on the location and type of sensors deployed against it. By assuming perfect knowledge on the part of the hypothetical submarine, any evaluation based on the derived optimum routes becomes a 'worst case' calculation. Arguably, this is the best type of evaluation in an area of warfare where precise enemy locations are not generally available in real time.

6. Surveillance Penetration - Given the set of 'Optimum Routes' from above, this feature estimates the 'worst case' cumulative probability of a submarine penetrating to a sequence of points along each route. Using conventional probability theory, the calculation takes into account the surveillance field, G(x,y,t) and the rate of progress of the intruder. The final result is portrayed graphically in our implementation as a contour diagram as indicated in figure 5. This concept is particularly useful for assessing the viability of surveillance screens or ASW barriers. In concept, the nature of this output is similar to that provided by the Search Residue calculation. Both provide a graphical representation of the likelihood of the likelihood of undetected submarines. The difference between the two calculations is found in the level of sophistication of the submarine behaviour model.

7. Passage Likelihood - A modified form of the Optimum Route Prediction algorithm is able to estimate the likelihood of an enemy submarine passing through each point in an area of interest. This feature requires that the objectives of the submarine are specified (eg transit, penetration to a point, shadowing). The specification must obviously take a mathematical form. The graphical result of this calculation indicates to Command very clearly any weakness in his ASW surveillance or screening operation. These could be interpreted as likely threat directions, given a knowledgeable enemy.

8. Nuclear Vulnerability - A tactical plan that has been optimised for ASW surveillance, using the mathematical tools described so far, may be far from optimal from the point of view of some other aspect of naval warfare. Naval Command must balance the benefit to be achieved by the plan against the potential cost that may be incurred if the plan is implemented.

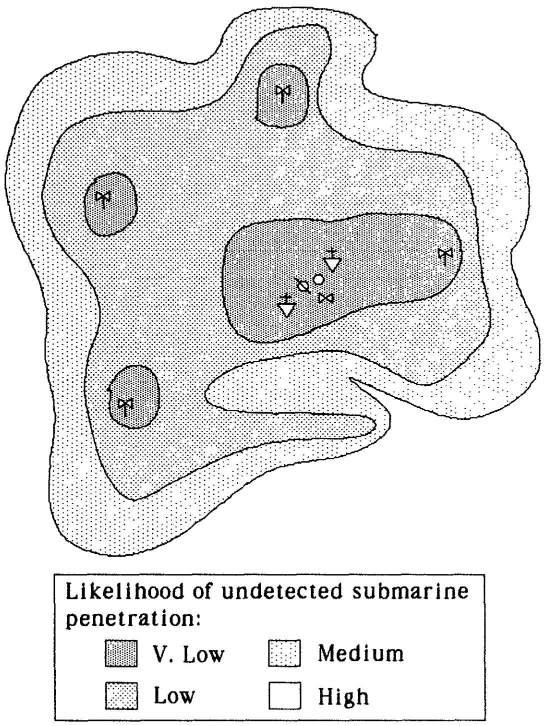


Figure 5 Surveillance Penetration Contours

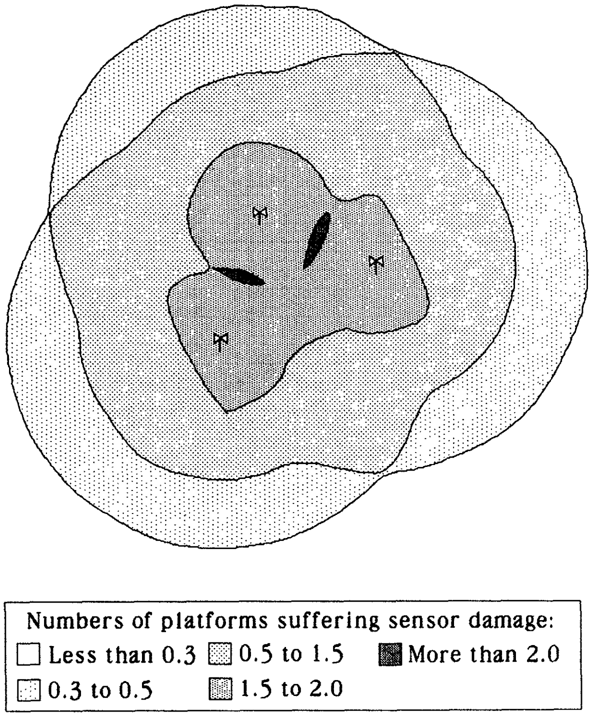


Figure 6 Nuclear Vulnerability Contours

A mathematical tool has been implemented in the command system machinery that evaluates the vulnerability of a platform disposition to damage from nuclear missile attack. Even moderate shock wave over-pressures can cause significant damage to relatively delicate parts of the superstructure such as sensor radomes. The detonation of a high yield nuclear weapon at an optimum height above the sea surface can inflict such damage at considerable distances. This feature, called Nuclear Vulnerability, predicts how many platforms would be affected in a specified way, given a particular type of warhead detonation at any location in the vicinity of the naval force. Results are portrayed graphically are a set of probability contours surrounding the platform disposition. Figure 6 gives an indication of how these contours might appear for a simple three ship disposition. The diagram indicates two regions where a detonation would be expected to affect all three platforms and an extensive region where more than just a single vessel would suffer damage. From the point of view of Nuclear Vulnerability this is a poor disposition but, for example, for ASW screening operations, such a disposition may be necessary to provide the required level of sensor coverage. With such mathematical tools as are provided here, Naval Command is in a position to start making an informed and balanced judgement on what is actually required by the prevailing tactical situation.

9. The Measure Function - The discussion on the Nuclear Vulnerability feature highlighted the need for Command to make cost/benefit trade-off decisions during tactical planning. Currently, such decisions must be made almost intuitively, even when decision aids of the kind already mentioned are provided. As a step toward making such decisions more quantitative, machine support for the C Process MEASURE function have been provided. For each decision aid calculation, a specific set of effectiveness measures could be determined. This approach is not desirable since a User must then remember the correct interpretation for each of them. An alternative approach, adopted here, is to apply common measures to all outputs. The MEASURE features so far implemented are concerned with analysing both the shape and extent of the likelihood contours provided by the decision support aids described above. This involves the calculation of the following quantities:

mean value (normalised surface integral) of the scalar fields representing decision aid output within a specified area of interest.

Fraction of the area of interest enclosed by mean value contour

the degree to which the mean value contour deviates from circular symmetry

The angular extent of any penetrating re-entrants in the mean value contour

From these measures it can be seen that an ideal likelihood contour generated as output from the decisions aids described here would be as large, dense and circular as possible, with no weaknesses indicated by the presence of a re-entrant. It is, of course, recognised that a particular tactical circumstance may present different requirements, requiring careful interpretation of results. Results produced from such calculations on each set of contours are expressed as a set of numbers which are tabulated for comparison.

The World State Maintainer

In the context of this paper, the WSM must provide real time estimates of the Occupancy field, S, prior to its use by the OP prediction features. In the absence of current information on enemy submarine presence, S may be pre-set to a constant value for each point in the area of interest. A degree of realism may be injected by ensuring that the surface integral of S equals the expected number of submarines in the area of interest. Similarly, S could be made anisotropic to reflect knowledge on known submarine operating areas. The significance of the WSM becomes more apparent when intelligence and other forms of sensor information are incorporated. One method of achieving this, but which has only briefly been investigated in this study, uses the techniques of dynamic programming and Bayesian statistics to predict likely submarine behaviour following a loss of direct sensor contact. This approach may be the subject of future work. Currently, attention is focused on the use of more conventional statistics for the estimation of Occupancy distributions. Any sensor contact may be expressed as a 'most likely' position together with an uncertainty distribution about that point, Pi(x,y,t), for the i'th contact. Similarly, estimates may be made of the most likely course and speed for each contact together with their uncertainty distributions. When a contact is lost, a critical parameter is the staleness, Ts of the last sensed set of data. The S field has contributions from both maintained and lost contacts, S = S(l) + S(m) (as well as any background values as described above). The Occupancy field, S(m) can be directly estimated from:

$$S(m)(x,y,t) = \sum_{i} Pi(x,y,t) \qquad (6)$$

(S is not bounded by 1.0)

To estimate S(l) a similar approach is adopted except that Pi(x,y,t) is convolved with an uncertainty distribution derived from the staleness and the uncertainties in course and speed. The Occupancy fields generated by these types of calculation are 'post-experience' estimations. Employing OP features like Search Residue permits a 'pre-experience' prediction of S(x,y,t) to be made. Such calculations would be performed cyclically by the command system to support the Command decision making process.

Concluding Remarks

The objectives of the programme are to determine whether the tasks facing ASW Command can be made easier and whether the overall command system can be made more effective. Answers to these questions are being sought by supplying to experimental subjects such features as are described here and performing controlled experiments within the simulation environment of the Command Systems Laboratory. The features under review are designed not only to speed up the processes of planning and monitoring but to provide Command with a quantitative evaluation of ASW problems and their possible solutions. Exceptionally, these decision support features have been contained within an implementation structure for a complete command system, the form of which directly reflects the results of a formal functional analysis study of the command and control problem. This is in strong contrast to the commonly found 'bottom-up' approach for creating stand-alone decision support aids. This creates systems that are often difficult to maintain and develop and usually cannot make full use of data contained in the physically separate conventional command system machinery they are designed to support.

To support this research activity, the CSL is also developing an extensive scenario simulation capability to provide the experimental command system with a realistic and credible operational environment. Early 1985 saw the beginning of a phased implementation programme for the experimental command system and the scenario simulator. 1986 should see a period of intensive experimentation aimed at resolving the issues discussed here.

Acknowledgments

Much of the work reported here was carried by Mr Dave Galley and Mr Bob Thearle from the UK company Systems Designers PLC under contract to the Ministry of Defence. Their contributions are gratefully acknowledged.

References

1. Galley, D. G., "The C Process: A Model of Command", Advances in C3 systems: Theory and Applications. IEE conference, 1985, Bournemouth, UK.

2. Miller, A. C., et al, "Decision Aids for Navy Tactical Anti-Air Warfare" Prepared for ONR, Dept of the Navy, ADA Project No. 2026, Applied Decision Analysis Inc., U.S.A.

A COMPARISON OF MANUAL AND AUTOMATIC DATA ASSOCIATION FACILITIES IN COMMAND AND CONTROL.

George Brander

Admiralty Research Establishment
Portsdown, Portsmouth,
Hampshire PO6 4AA, England.

INTRODUCTION

This work represents an early part of the programme of command and control systems research at the M.O.D. Admiralty Research Establishment, Portsdown, England, undertaken by the Command Systems Laboratory, a new research facility which has been developed both to explore and test the functionality of novel command and control system prototypes.

The general area addressed by the study was the fusion of data from multiple sensors into a processed picture, useable by command, for tactical purposes.
In particular, the work concentrated on the problem of deploying towed array sonar on surface ships where orders of magnitude increases in contact density are likely compared with the typical submarine case.

Following an examination of the problems and potential deficiencies inherent in current methods of sonar data processing, two alternative approaches were suggested. The aim of this study was to build, test and evaluate these two candidate solutions; learning lessons on the construction of each, gaining insight into the merits of each approach and, eventually, obtaining precise comparative performance data.

At a more fundamental level, this paper outlines a technique for the comparative evaluation of an automatic, algorithmic solution and a human centred or manual solution to a complex decision making task.

A REAL WORLD PROBLEM AREA

The towed array sonar, as currently deployed by Royal Navy frigates, was selected as the system of interest for two major reasons. Firstly, the towed array sonar generates an intensive flow of surface and subsurface contact information that needs to be apprehended by Command. This is due to the relatively large range of this sensor, particularly for noisy vessels in good environmental conditions. Secondly, being a passive device, the towed array does not easily yield range information about its detections and, in addition, it generates observations which are ambiguous in bearing. This ambiguity constitutes a substantial difficulty for the present generation of Target Motion Analysis techniques which demand manoeuvre by the towing ship (i.e. re-orientation of the array) in order to resolve it.

Consider a typical scenario [Figure 1]. A towed array frigate might be tasked with defending the southern approach to a protected lane of allied merchant shipping. All of these platforms are likely to be noisy and the sonar will yield detections on many bearing lines. However, the real world may also contain unknown hostile elements. In this example, an enemy submarine is attempting to intercept our merchant ships and is closing quietly from the south.

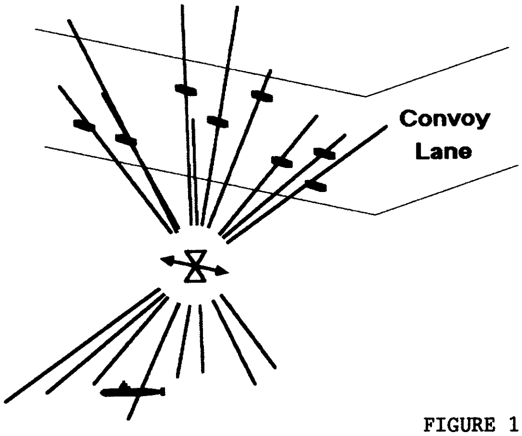


FIGURE 1

The problem for the frigate is to identify friends from foes given the complexities of the sonar data alone: no easy task. The answer may be to utilise the more precise information from other sensors, such as radar or ESM, and to correlate this with the sonar picture.

Radar sensors on "own ship", or link transmitted from another cooperating platform, provide precise data on the range and unambiguous bearing of surface and air contacts. Thus surface vessels within the protected lane should be identifiable by range and bearing from our frigate. Where a known friendly surface vessel coincides with an underwater acoustic emission, detected on the towed array sonar, data can be correlated and classified as non-critical, thereby reducing the sonar processing task to those unresolved or uncorrelated sonar detections alone which may present a potential threat.

PROCEDURE

Firstly the way in which towed array sonar data is processed at present was examined. Currently the task contains a feedback loop which is essential if Command is to obtain useful, comprehensible data from the sensor [Figure 2]. Because processing power (in terms of the capabilities of unassisted human operators) is limited, some filtering takes place on the raw sonar data. This is achieved at present by classifying contacts according to their typical frequency signatures and thereby assessing their likelihood of posing a threat.

It was postulated that this approach was a potential source of error; specifically that a sonar operator may "lock on" to an incorrect classification hypothesis and consequently confuse the picture presented to higher levels of command in the operations room.

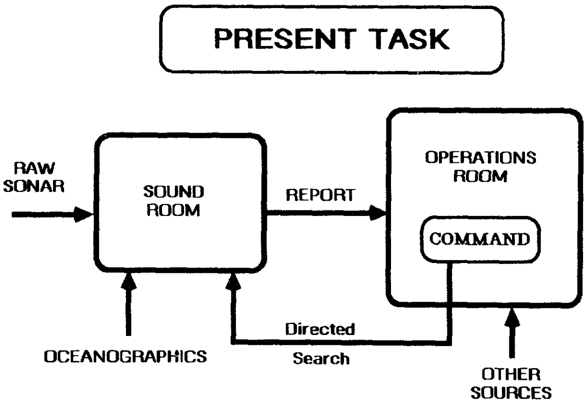
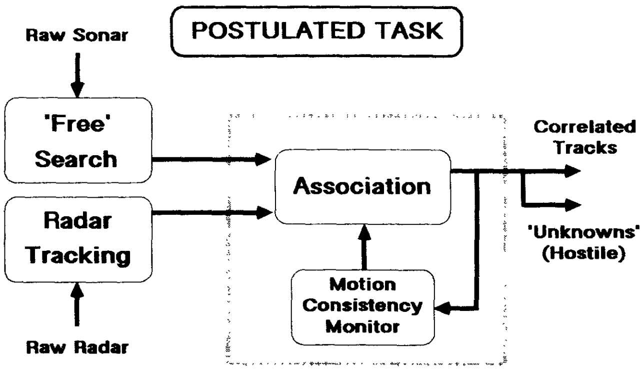


FIGURE 2

A new approach was advocated which argued that by delaying classification and allowing data fusion to occur on relatively raw data and at a number of intermediate levels, the feedback loop described earlier could be avoided. This would free the sensor search task from any constraints, decrease the opportunities for incorrect classification and should increase the throughput of the processing team [Figure 3]. The need for a facility which monitors the internal consistency of the correlated data entities which the operator can create, was also anticipated.



Radar—Sonar Data Fusion System

FIGURE 3

An unambiguously observable characteristic of sensed objects, namely kinematic behaviour, was chosen as the basis on which data fusion would first occur.

Our objectives, therefore, have been to create an environment in which the radar and sonar sensor data can be presented, selectively manipulated, and finally associated to yield a clearer picture to senior command. As well as this manual mode of operation, an automatic mode was also constructed in this environment so that the operator could, if the situation permits or demands it, merely monitor the associations which the system itself is making.

THE MANUAL MODE

In the manual mode [Figure 4] the computer fills up a "history store" from the output of the scenario generator (the simulated real world) and permits the operator to inspect and manipulate the contents of this "history store" through a group of displays. It also provides an "associated data store" for the conclusions of the operator's associations.

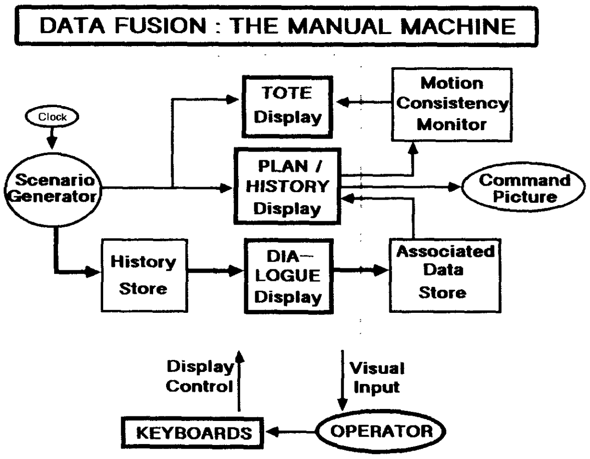


FIGURE 4

The hardware of the system [Figure 5] consists of a workstation which, from the operator's viewpoint, comprises a graphic display screen on which he can select to view either a geographic plan or a history picture and an alphanumeric tote display on which he can examine detailed information about specific chosen contacts. There is also a display and keyboard through which he can conduct a dialogue with the machine and manipulate the data and, finally, a pre-programmed keyboard which allows him to readily manipulate the graphic display and its contents.

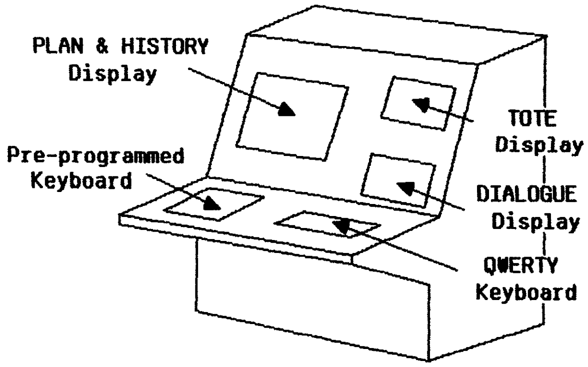


FIGURE 5

The pre-programmed keyboard allows him to select the subset of current data he wishes to inspect in plan (or geographic) form and the scale and offset of the presented display. The operator can also choose to examine the data such as bearings, frequencies or ranges, which have accumulated historically and can display this at different scales against a time axis. To do this, he must first use the dialogue channel to select which two contacts, radar or sonar or both, he wishes to display.

The tote display reports the current data available on these two contacts as well as containing a message area (at the bottom of the screen) where the operator can be alerted by the system to new events such as detections or fading contacts.

The current plan picture shows the sonar lobes which currently contain acoustic contacts, the current radar contacts, a notional radar horizon to give the picture scale, a reduced overview to show the scale of the current picture and its location in relation to the overall playing area, a list of the contacts contained in each activated sonar lobe (lobe contents) and a status area indicating what subsets of the data, raw or associated, are currently on the display. When the operator associates raw data, his main task, he is said to form a "group", indicated by the prefix "G", and this is also displayed on the plan picture.

The association task is carried out by the operator examining the history data on the bearing/time plot [shown schematically in Figure 6]. When the bearings of a sonar and a radar contact overlap throughout their entire history, he can conclude that they arise from the same object and "Join" them through the dialogue channel. He can subsequently "Split" them, if he changes his mind, as he might do if the contact histories were to begin to diverge.

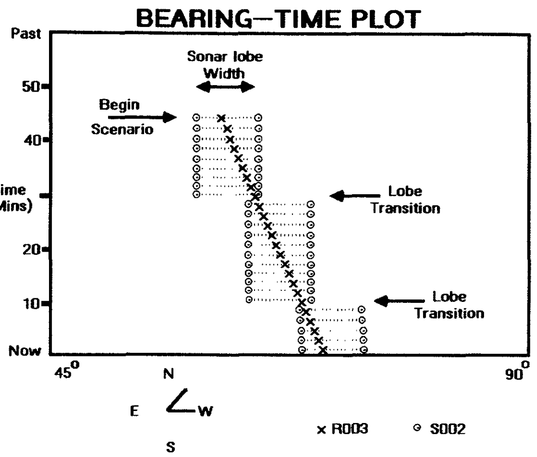


FIGURE 6

THE AUTOMATIC MODE

The structure of the automatic mode is conceptually simpler. It is based upon a mathematical correlation technique or track association algorithm which was developed by the U.K. company SCICON Ltd. under contract to the M.O.D. (References [1] and [2]) This algorithm progressively agglomerates each data stream separately into "segments", then into "tracks" and finally it associates the separate tracks from each stream. All the historically accumulated data is considered afresh on a periodic basis. The objective function which lies at the heart of the algorithm depends upon the sum of residuals of the fit of "segments" to data and upon a group of user chosen penalties which seek to minimize the complexity of the final picture [See Figure 7].

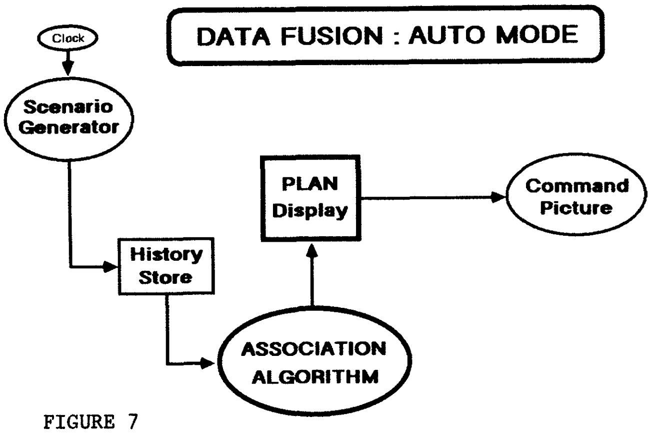


FIGURE 7

THE EXPERIMENTAL RIG

Experimentally our objectives were to examine the relative merits of manual and automatic modes of data fusion as well as to consider the absolute ability of both man and machine to infer the best view, or appreciation, of the data presented. In order to compare performance a simulated environment was created to exercise the two approaches and an experimental test rig was constructed in the laboratory. [Figure 8] Note the conceptual 1-pole, 2-way switch under the control of the experimenter.

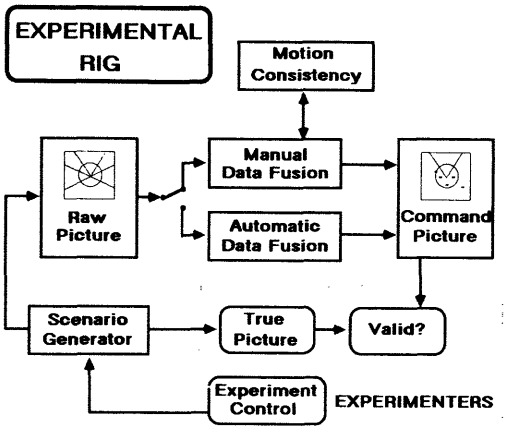


FIGURE 8

A scenario generator enabled random but repeatable scenarios to be constructed, as well as specific, deterministic ones of known complexity. The objects which constituted the scenario provided data flows to the two systems typical of those experienced by radar and sonar sensors currently in use by the Royal Navy.

During an experimental run, the output of each system (that is, the associated or command picture) was recorded for subsequent off-line analysis and, in addition, all human interactions with the manual version were recorded for the later investigation of pertinent man-machine interface issues.

PILOT STUDY

A pilot study has been completed using a simple scenario with five surface ships and one potential subsurface contact. Although this would appear straightforward, since all objects were moving at constant speed and heading, it should be noted that the scenario was constructed so that overlap would occur on three sonar contacts, a situation that poses a difficult decision problem for both the operator and the algorithm.

Six subjects drawn from the population of scientists at A.R.E. were given detailed instruction in the task and allowed to practise the skills involved before undertaking the hundred minute experimental session. Following this they were debriefed and given the opportunity to comment on the task itself and the facilities available to assist them. The automatic, algorithmic mode was presented with the identical scenario and its solutions, at every five minute review period, recorded for subsequent analysis. Finally the performance of each system was measured and compared with the ideal, that is to say perfect, solution.

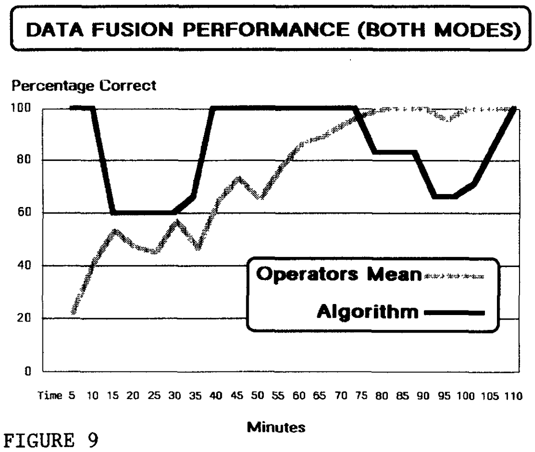


FIGURE 9

RESULTS

The results so far have proved quite interesting: both systems survived the entire experimental session and completed the fusion task [Figure 9]. Operators appear rather conservative in their decision making behaviour, being reluctant to commit themselves to association decisions until a good pattern of information is available in their history store. However they subsequently perform very well indeed, anticipating changes (such as a contact moving across the towed array lobe boundaries) and they react quickly to maintain their refined picture as new data arrive. The algorithm starts well but is prone to periods of poor performance. It is not able to react to changes based upon any prediction of events about to occur. Examining the performance in more detail [See Figure 10] we can see that the algorithm is guilty of errors of omission and commission. Sometimes it does not associate data which should be associated and it sometimes makes the wrong association. The operators, on the other hand, rarely make mistakes and, having achieved a correct solution, can maintain it quite easily even when new events occur.

PROBLEMS

There were a number of problems which were detected and corrected during the development stage. These were primarily concerned with the facilities presented at the man-machine interface which were not well attuned to the task requirements as they were then understood. However, more fundamental problems were only detected during the pilot study and these have not yet been fully overcome. It had been assumed that the two systems (the manual and the automatic) would access identical data from the scenario but it was discovered that they were, in fact, viewing slightly different sets of sensor information. The track association algorithm had an easier task in that it did not receive the multitude of fleeting contacts, caused by a sonar contact intermittently triggering the edge of a sonar lobe, which posed a significant workload on the human operator. Although this casts some doubt upon the validity of the pilot study comparisons, I believe that the general trend of the results merits attention.

LESSONS LEARNED

What have we learned from this pilot study? We have just begun to explore a hitherto nonexistant and unknown task and, in doing this, have learned something of how operators wish to carry it out. Apart from their criticisms of the ergonomics of the console and the rather slow system response times, the major criticisms concerned the facilities provided for comparison of track histories. Operators spent almost forty percent of their time viewing the history picture, but most of their decisions resulted directly from an inspection of this display. Only two tracks could be viewed at once and it would have been preferable if several candidate track histories could have been compared simultaneously. Although it seemed that operators spent more time with the current plan picture selected, this was used as the monitoring display during long periods of low scenario activity when they were waiting for something to happen.

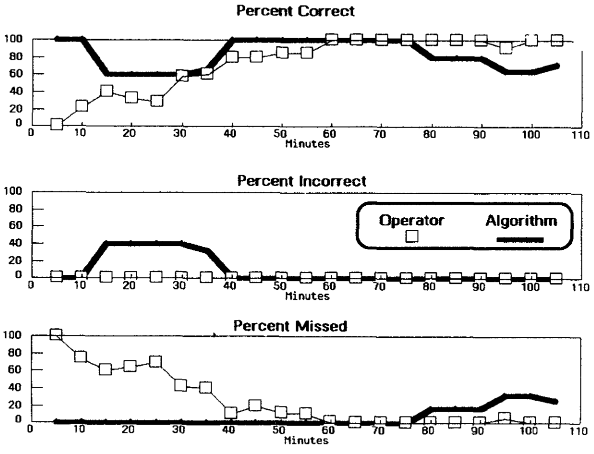


FIGURE 10

WAY AHEAD

In the future there are three broad categories in which this work can be capitalised upon. Firstly the system could be made more realistic and sophisticated by introducing more platforms and sensor types, such as a second co-operating towed array frigate and helicopters with dunking sonars. Secondly there is a substantial body of experimental work to be carried out with the system to discover whether it can meet the loads which might be placed upon it. Finally, the system offers a potential kernel for the study of how, operationally, the towed array could be best incorporated into the Royal Navy. In addition, the experimental system offers a test rig on which other candidate data fusion techniques, such as the expert systems approach, could be tested and evaluated in comparison with the two approaches described above.

CONCLUSION

In conclusion, there appear to be fundamental differences between these manual and algorithmic approaches to the particular decision making task inherent in the data fusion problem. It is anticipated that a symbiotic mode of operation is likely to be most beneficial, whereby the advantages of each approach could act in mutual support. However, it is only by developing our understanding of the capabilities of the human operator and by marrying these to the best capabilities of the available technology, that we will be able to progress towards a more efficient and effective man-machine system.

REFERENCES

[1] Bamford,A.C., Beale,E.M.L., East,D. (SCICON) A Track Association Algorithm. Presented at 6th MIT/ONR Workshop, July 1983.

[2] Bamford,A.C., Beale,E.M.L., Patel,S. (SCICON) The use of Reciprocal Polar Co-ordinates in Passive Tracking. Presented at 6th MIT/ONR Workshop, July 1983.

INTELLIGENT DATA FUSION AND SITUATION ASSESSMENT

W L Lakin and J A H Miles

Admiralty Research Establishment (Portsdown)
Portsmouth PO6 4AA England

This paper discusses two aspects of the application of artificial intelligence (AI) techniques, particularly knowledge based systems, to the problems of data fusion and situation assessment. These are: firstly, the nature of the problem, including the strategy we have adopted in trying to solve it, and secondly the architecture, software and hardware, needed for real-time operation. Both aspects have been the subjects of a major reseach programme at the Admiralty Research Establishment (ARE) over the last four years. User and engineering aspects, which will be addressed in our future research programme, are also important and specific problems in these areas are mentioned. The paper also describes a laboratory demonstrator which forms a major product of our research programme.

Introduction

Background

The work described in this paper forms part of a research programme in progress at the Admiralty Research Establishment into the use of artificial intelligence (AI) techniques to provide automated support to the tasks of situation assessment and resource deployment [1] in naval command and control. The specific problem addressed here is that of generating an appreciation of the tactical situation facing either a single warship or a naval task group. This can be considered in two main stages: the first is the compilation, from the information available, of an objective and coherent 'world picture' in the area of interest; the second is the derivation of an intelligent assessment of what that picture means in tactical terms. We refer to these respectively as 'data fusion' and 'situation assessment'.

In order to build a picture, it is necessary first of all to detect, locate, track and, if possible, classify all objects which might conceivably contribute to the tactical situation. This implies virtually every object within sensor range or within the volume of interest to a single warship or to a group of cooperating maritime units, which may be dispersed over a wide area of ocean. In constructing the picture, it is important to consider, not only all the real-time sensor data, but also what might be termed secondary or non-real-time data so as to provide further evidence for classifying the objects, predicting their intentions and gaining a general appreciation of the tactical situation. Consequently, the information sources include not just radio, acoustic and optical devices, but also human observers providing intelligence data and a background of encyclopaedic information and operational plans, all of which set the context for the more dynamic real-time sensor information. The task of combining such disparate data types, having proved well beyond the capabilities of conventional computing methods, has remained the province of the already overloaded human operator, and yet it has to be undertaken in timescales which allow effective response to be taken against today's high-speed missile threat.

The requirement

Stated simply, the requirement is to maintain an assessment of the tactical situation in the volume of interest to a group of warships and supporting units. In a military context, all available sources of data should be used to produce the best assessment with the least vulnerability to interference, by the enemy, to any particular source.

There are several difficulties with current systems:

- The volume of data available to the command is already overwhelming making it difficult to pick out the tactically significant features.
- Reaction times to many threats are too slow because of the number of manual inputs required from initial detection to deployment of countermeasures.
- There is little or no support for the decision making of senior staff.
- In some cases the partitioning of tasks between men and machines is inappropriate; men being used for simple processing tasks rather than those requiring their cognitive skills.

Future trends will make the problems worse rather than better, for example:

- More powerful advanced sensors, incorporating the latest signal and data processing and producing an even greater volume of data, will be available to the command.
- New types of sensor and improved datalinks will increase the number of data channels.
- Current systems tend to depend on radar as the primary source of the tactical picture, but provide little support for passive operations.
- More complex scenarios are likely to result from greater sophistication in sensor and weapon systems.
- Significant reductions in operations room manning levels will be required in future as a result of escalating manpower costs.

The data fusion problem

Why data fusion?

The requirement for data fusion derives from a number of factors. For example:

- No one sensor provides all an object's attributes, either to the desired accuracy or in the right timescale.

- We need to infer new parameters not directly measured by sensors and in some cases estimate more accurate parameters using measurements from several sensors.
- Not all sensors are always available because of failure, jamming or an emission control policy.
- We need to correlate information over long time periods in order to incorporate longer term Intelligence data and to be able to recognise patterns of behaviour.

Data channels

Inputs available to the tactical data fusion process include:

- Own-ship sensors including: radars, IFF, ESM, active and passive sonars.
- Datalinks for data communications between ships, aircraft and shore bases.
- Signals, voice reports and all forms of Intelligence.
- Tactical plans for ship and aircraft deployments - these provide useful context information for interpreting sensor data.
- Local events including own aircraft launch and recovery, missile firings etc.

In addition, there exists a considerable amount of encyclopaedic information, concerning geography, military equipment details and tactics, which needs to be taken into account.

Correlation and combination

Data fusion can be viewed as a two stage problem. First the evidence must be correlated to find which pieces belong to the same real world objects; secondly, the evidence must be combined to estimate and infer the required object parameters.

The problem is not so much in the combination stage, where statistical estimation and expert systems have demonstrated successful solutions. It is correlation where the most difficulty lies. Because the evidence may be inaccurate, uncertain, false and late arriving, there is ambiguity in the way it fits together. This gives rise to many possible world views, and much of the research effort has been spent finding a general strategy to handle this combinatorially explosive problem.

Strategy for Correlation

Figure 1 illustrates some of the input data as a plan view of the world:

- Radar tracks are shown as lines from first detection to present position.
- ESM contacts are shown as bearing lines along which electronic emissions have been detected, giving approximate direction but little idea of range.
- The sectors represent intended operating areas for friendly units. (A similar representation may be used to indicate approximate positions of units reported as intelligence.)

Figure 1 illustrates another important feature: radar tracks exhibit breaks when the target goes out of range or into clutter, or the set is jammed. These breaks need to be

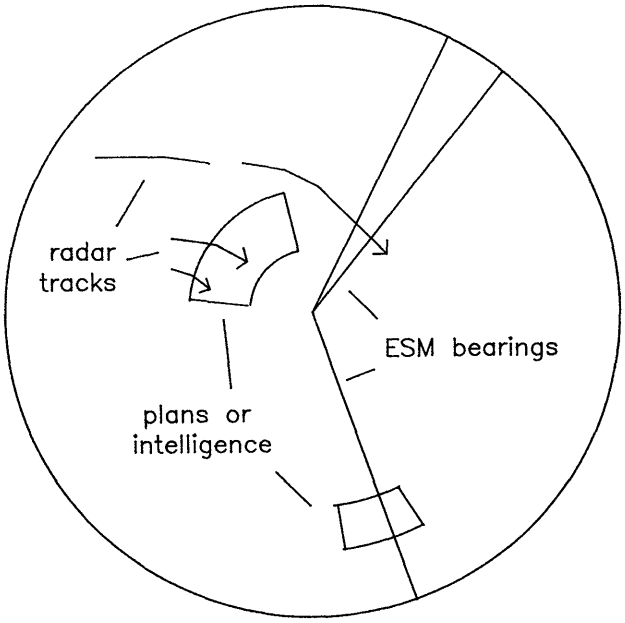


Figure 1: example input data

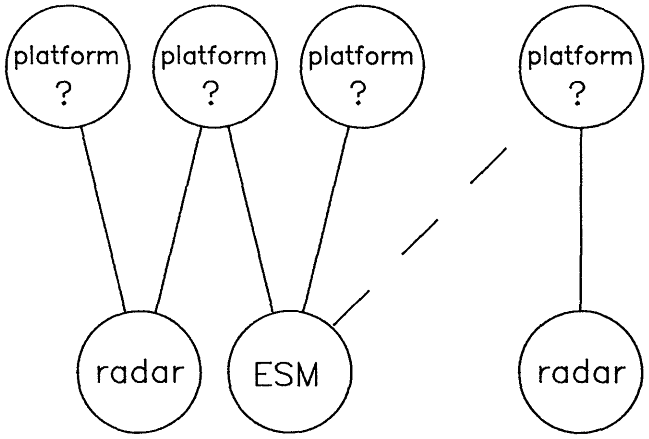


Figure 2: correlation ambiguities

repaired which requires correlation in time of data from the same sensor as well as correlation between sensors.

This senario is, of course, much simplified. In reality there could be many hundreds of track segments to deal with over say a 1-hour period, and a very confusing picture can result. Hence the importance of tying together all data belonging to each real vehicle so that the picture resembles the real world as closely as possible.

What we are seeking to achieve is the aggregation of information from tracks on the same vehicle from different sensors (using "track" to mean a set of data over time from any sensor about a single vehicle, and using "sensor" to embrace both real-time and non-real-time data channels). However, there is a problem of correlation ambiguities, illustrated in Figure 2. Even if the radar and ESM contacts on the left of the figure are deemed by the correlation rules to be capable of correlation, there are still two possibilities:

- Either they are the same object - denoted by the middle platform hypothesis,
- Or they are two different objects - denoted by the other two platform hypotheses.

A single object detected by n sensors will generate $2^n - 1$ hypotheses. As each sensor detects not one but many objects, there is also further ambiguity as to which contacts go together, as illustrated by the radar contact on the right. The inherent inaccuracy of most of the sensor data implies loose correlation rules leading to large numbers of such ambiguities. The need to process several new contacts per second leads to unmanageable combinatorial difficulties with such an approach.

But there is a further problem, illustrated in Figure 3. Having generated all possible hypotheses to explain the data, it is necessary to decide which ones to output as being the most likely. We attempted to do this by generating every possible consistent output set and then scoring these in some way. For example, taking the previous example of two radar tracks either one of which could correlate with an ESM track, there are three possible self-consistent output sets or "views" of the situation. These are:

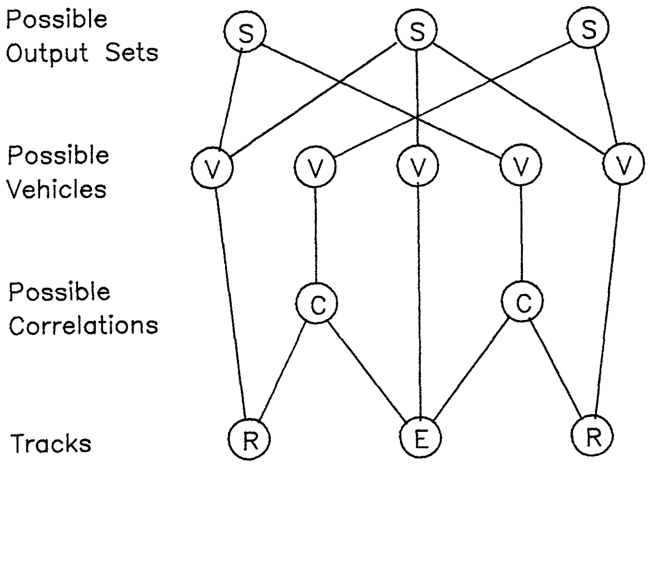


Figure 3: output sets

- Left-hand correlation is valid.
- Right-hand correlation is valid.
- Neither correlation is valid.

By representing all consistent output sets, given a satisfactory scoring system, the "current best view" is simply the set with the highest score. Unfortunately the enormous number of sets ("views") resulting from any realistic scenario renders the approach impractical. For example, even for a single object detected by n sensors, the number of logically consistent views of the situation is given by the so called Bell number B(n), where:

$$B(n+1) = \mathrm{Sum}_{i=0}^{i=n} \left[\binom{n}{i} B(n-i)\right]; \qquad B(0) = 1$$

When n = 10, for instance, B(10) = 116000.

We adopted a less rigorous but more practical approach. This considers only pairwise correlations and involves 3 distinct rule-driven steps:

- First, assume all new contacts are separate and therefore each new contact implies a new vehicle.
- Second, apply rules which create the possible pairwise correlations between each new track and existing tracks in the system. These correlations must be periodically checked to make sure they are still valid; those that fail the check are deleted.
- Third, apply rules to confirm strong correlations and to deny others. Where alternatives are of similar strengths, wait for further evidence. Such a rule might require that, in order to confirm a correlation, its likelihood must exceed some absolute threshold and also significantly exceed other possibilities, ie be relatively unambiguous. In addition, the correlation must be part of an allowed set. For example, if Track A tentatively correlates with Track B, and Track B with Track C, these can only be confirmed if Track C also correlates with Track A. In other words, all tracks supporting the same vehicle must mutually correlate on a pairwise basis.

Tactical data fusion stages

The overall data fusion process is performed in three principal stages:

- Correlation of sensor tracks and reports
- Estimation/inference of platform parameters
- Forming platform groups

The first two stages, discussed already, form the basic tactical picture of individual platforms with position, velocity, identity, allegiance etc. In order to make higher level statements about the tactical situation it is necessary to group the platforms into spatial and functional groups. Inferences can then be made, for example, about own force's defence screen and about the enemy's attack capability and possible strategy.

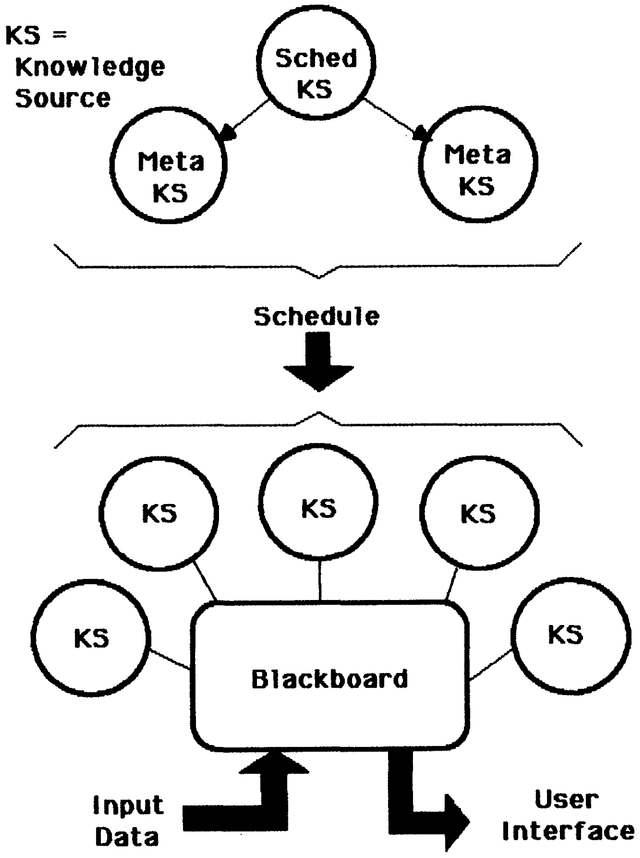


Figure 4: blackboard system

Data fusion demonstrator

Knowledge-based architecture

A 'blackboard' type of expert system architecture [2] has been chosen as the basis for our laboratory demonstrator. This architecture is illustrated in Figure 4 and consists of a global, dynamic hypothesis structure accessed by rules grouped within knowledge sources (KS). Input data are placed on the blackboard as new hypotheses, and the correlation and combination rules are applied to develop further layers of hypotheses to represent the 'tactical world'. Rules are also used to determine which KS to invoke next. These scheduling rules are contained in Meta-KS's.

A general-purpose blackboard framework was designed and constructed for the purposes of this programme and is known as MXA (Multiple Expert Architecture) [3]. MXA supports a language [4] for encoding the rules which is compiled by the MXA compiler into Pascal. Compiled Pascal is then linked with an MXA run-time executive to form run-time system.

Function of demonstrator

A demonstrator for tactical data fusion has been developed in a laboratory environment. It takes six inputs, radar, IFF, ESM, plans, intelligence and events, and ouputs a real-time correlated display.

During the past year, a more advanced version has been under construction. The new demonstrator includes sonar sensors and datalinks, and is therefore a multi-platform, multi-sensor tactical data fusion system. It includes rules to match multiple radars both on own-ship as well as from remote platforms, rules to cross-fix ESM bearings received on multiple platforms, similar rules for passive Sonar bearings and rules for correlating all the different types of data sources.

More comprehensive rules are currently being formulated to derive platform parameters from the correlated evidence, and a start has been made on providing the higher levels of inference necessary for situation assessment.

Components of demonstrator

The demonstration system includes the following components:

- **SDGS** - Sensor Data Generation System. This software provides offline generation of realistic sensor data at the plot level from a scenario description.

- **RATES** - Radar Automatic Track Extraction System [5]. An off-line version of our radar tracking software is used to form radar tracks for input to the data fusion system. A simplified version is used for ESM data.

- **EDP** - Exercise Data Preparation. This software is used to prepare real data, recorded during naval exercises, for input to the data fusion system.

- **Graphics.** In order to demonstrate the data fusion system in real-time, special purpose graphics software has been developed to drive a high-resolution colour graphics terminal using the GKS language.

- **MXA** - the blackboard expert system framework. This runs on a VAX^{tm} computer using the VMS^{tm} operating system.

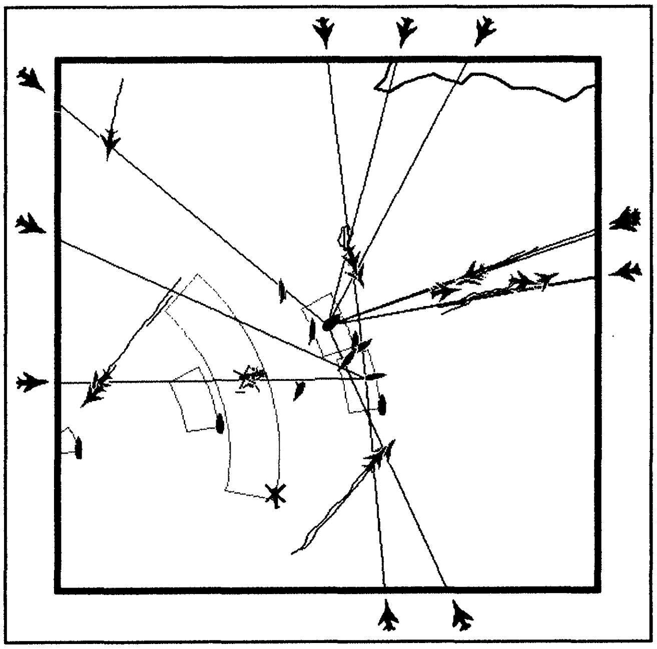


Figure 5: uncorrelated display

- **Data Fusion Model.** This is the data fusion knowledge coded in the MXA language.

Outputs of demonstrator

Figure 5 shows the uncorrelated data display, in other words the input data. It is based on a real naval exercise and uses actual data recorded at sea, supplemented by simulated data where necessary. Bearing lines represent ESM tracks with simple platform type and allegiance (shown by colour on the actual display). Radar tracks have basic vehicle type symbols derived from simple rules which make use of track behaviour. Plans for ships are shown as sectors. Other inputs such as IFF and Intelligence reports are not shown in this example.

Figure 6 shows the correlated display produced by the data fusion rules. Several ESM - radar correlations can be seen (depicted by dotted lines attached to platform symbols, as exemplified by the aircraft symbol in the north-west corner). The two bearing lines to the south in Figure 5 refer to ESM contacts received on two different ships from the same contact. These have been triangulated to form a position hypothesis, and then correlated further with tracks from two radar sets to produce the single aircraft symbol shown in Figure 6.

The three sectors with dashed, as opposed to solid, outlines have been correlated with other data (primarily radar), thus providing contacts both with high positional accuracy and with reliable identity information. The benefits of this are not obvious from the print-out, but derive from the fact that all the objects on the screen are mouse-sensitive and can be selected by the operator to provide a comprehensive text read-out of the object's parameters, along with an indication of how these were derived. As a further aid, the symbols on the actual display are in colour.

Situation assessment

Work has just begun to provide automated assistance to the interpretation, in tactical terms, of the 'world picture' generated in the data fusion process. Situation assessment is not intended as a single form of tactical description but rather a collection of assessments relating to aspects of tactical interest. The following list gives some examples.

- **Threats.** Assessment of potential, possible or actual threats is obviously useful.
- **Engagements.** Assessing the outcome of an engagement in real-time is important but it may be difficult to achieve depending on what evidence is available.
- **Rules of Engagement.** At the build-up stage of a conflict, a commander will have to assess whether he should take positive action, and, for guidance, rules of engagement will be in force. An assessment of the threat against such rules might assist in the judgement of when decisions have to be made.
- **Sensor and weapon coverage.** In judging the effectiveness of the defence screens, an assessment of sensor and weapon coverage would be of value.
- **Plan monitoring.** Given that a number of plans for defensive and offensive actions are in force at any time, this assesment is intended as a check to see how well these plans are proceeding.
- **Surveillance.** When active sensors are used there is a risk that the enemy will be able to detect transmissions and gain information. An assessment of the extent of knowledge on both sides might help to suggest which active sensors should be deployed.

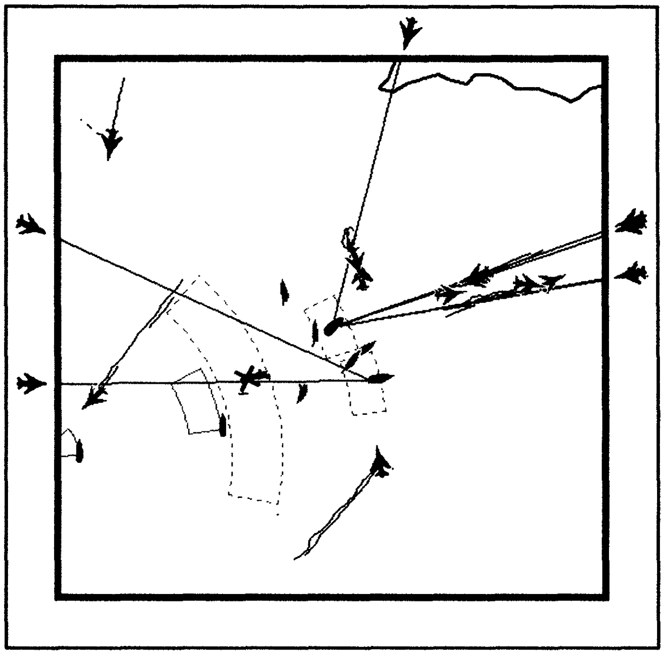


Figure 6: correlated display

Research topics

As our research has progressed, a growing list of subjects for further study has emerged, some of which are outlined below under the principal headings of architecture, user aspects and engineering.

Architecture

Inferencing and uncertainty. The representation of uncertainty and the combination of uncertain evidence is a problem common to many expert systems. We have used a simple Bayesian scheme so far but we are currently studying other methods. It is likely that we will use different ways of representing uncertainty for different dimensions of the problem.

Scheduling. Blackboard expert systems, in principle, use an intelligent, rule-based scheduling scheme to select the knowledge source to be invoked in order to concentrate the inferencing resources on the most appropriate part of the problem. This scheduling knowledge will be application specific but it may be possible to use a general framework for its construction.

Database. In addition to the knowledge held as rules in the expert system, a great deal of background information is needed about tactically significant objects and the world in general. For convenience this information could be held in a general purpose relational database. Unfortunately this approach incurs access overheads which are unacceptable for real-time operation. Some intelligent interface is required to such a database in order to overcome this problem.

Parallel hardware. It has been recognized that in order to achieve real-time performance in a tactical data fusion system for operational use, it will be necessary to provide much faster processing than in our demonstrator. Existing parallel processing systems are being studied to see whether our problem solution could be mapped onto them with an effective increase in performance.

Massively parallel archtectures. The above study of parallel processors, to support our existing type of expert system, is limited essentially to coarse-grain parallelism. As a longer-term activity, fine-grain parallel sysems, such as the TMI Connection Machine^{tm} [6] are also being studied. The difficulty here is to map the problem onto the architecture efficiently to gain the potential benefit of their parallel operations.

User aspects

Explanations. The man-machine interface (MMI) is particularly important in expert systems because it is essential that humans can easily comprehend the reasoning which the system is applying to the problem. Explanations are often quoted as the mechanism for human comprehension of the system, but, in a real-time expert system dealing with a large amount of evidence and firing up to a hundred rules per second, it is not easy to see what form explanations should take. Animated graphics are one form of explanation which can cope with the volume of information to be communicated and these form an essential part of our MMI. There is however more work to be done on the interactive facilities.

Human factors. A more fundamental question is the relationship of humans to a real-time expert system. Such a system, if provided with sufficient expert knowledge, could operate quite autonomously. To what extent the human should be allowed to control its operation is important from the point of view of flexibility and responsibility for actions. These human factors questions require further study.

Engineering

Engineering disciplines. Three types of engineering are required for a complete knowledge-based system (KBS). These are:

- Knowledge engineering.

- Software engineering.

- Hardware engineering.

Software and hardware engineering principles are fairly well established but no one has yet produced a definitive set of principles for knowledge engineering.

There are good reasons for considering knowledge engineering quite separately from software engineering. Firstly it is the objective of most knowledge-based systems to keep the knowledge visible to the expert at the knowledge acquisition stage and visible to the user during operation, whereas the user of a software system usually has no interest in the program at any stage but only in the results it produces. Secondly, software systems can usually be described by simple generic models such as database transactions and syntax driven logic, whereas knowledge can be specific, unstructured and impossible to specify in a more concise form.

The ideal KBS consists of:

- Domain knowledge - expressed in a language easily understood by both domain expert and user,

- Domain independent software - to support and apply the knowledge,

- Hardware and firmware - to run the software efficiently.

In many cases, however, some domain dependent software will also be required, for example to provide the man-machine interface and to implement algorithmic parts of the problem.

Specification, evaluation and acceptance. At this stage, development of a real-time expert system for tactical data fusion poses significant difficulties, and we have not yet determined how it would be specified for procurement by the RN. Evaluation even of a laboratory demonstrator is difficult because of the complex environment in which it must work. Specification, evaluation and final acceptance into service are all outstanding problems.

Conclusions

This paper has described research aimed at applying KBS techniques to the problem of tactical data fusion and situation assessment with specific reference to our completed data fusion demonstrator. Our objective is to gain some insight into whether or not expert systems are capable of supporting one of the main elements of command and control, and we have outlined a number of key research issues which need to be addressed. The problems of incorporating sufficient knowledge to cover all situations and of proving the performance of the system are substantial but, although the system can never be perfect, it may, as with other uses of computers, be provably better than existing manual methods.

References

1. Gadsden, J.A. and Lakin, W.L., "FlyPAST: An Intelligent System for Naval Resource Allocation", ibid., 1986.

2. Nii, H.P., Feigenbaum, E.A., Anton, J.J., and Rockmore, A.J., "Signal-to-Symbol Transformation: HASP/SIAP Case Study". AI Magazine, **3**, 23-35, 1982.

3. Rice, J.P., "MXA - A Framework for the Development of Blackboard Systems". Proceedings of the Third Seminar on Applications of MI to Defence Systems, RSRE UK, 1984.

4. Stammers, R.A., "MXA Language Manual". SPL International, UK, 1983.

5. Shepherd, A.M, White, I, Miles, J.A.H, "RATES: Radar Automatic Track Extraction System - A Functional Description", internal memorandum XCC82003, Admiralty Research Establishment, Procurement Executive, Ministry of Defence, Portsmouth, Hampshire, England, 1982.

6. Hillis, W.D., "The Connection Machine", The MIT Press, Cambridge, Massachusetts, 1985.

Acknowledgements

We would like to acknowledge the contributions of the many people who have been involved in the ARE data fusion programme, and particularly those of John Montgomery and Jon Haugh.

FlyPAST: AN INTELLIGENT SYSTEM FOR NAVAL RESOURCE ALLOCATION

J A Gadsden and W L Lakin

Admiralty Research Establishment (Portsdown)
Portsmouth PO6 4AA England

This paper describes FlyPAST, a knowledge-based system built at the Admiralty Research Establishment to provide automated support to a naval command team in the production of fleet flying programmes. The design of flying programmes is a resource allocation problem which allocates fleet resources (aircrew and aircraft) to missions; the design is guided by the application of multiple constraints. This paper describes the nature of the problem, the FlyPAST structure, the design of the current implementation and the work in hand to improve this. It also indicates the areas for future research.

Introduction

The work described in this paper forms part of a research programme in progress at the Admiralty Research Establishment into the use of artificial intelligence (AI) techniques to provide automated support to the tasks of situation evaluation [1] and resource deployment in naval command and control. For convenience, the research into resource deployment has been subdivided into four categories, the basis for subdivision being the effective timescale of the plan generated. The following category names have been arbitrarily assigned:

- Mission planning (timescale: one or more weeks).

  The formulation, by the senior command echelon, of objectives and general deployment patterns for major resources such as a naval task group.

- Resource planning (timescale: one or more days).

  The more detailed allocation and deployment of individual assets on a day-to-day basis in response to the overall requirements for the mission. Examples include assignment of the patrol areas for the ships and aircraft in a defensive screen, and the management of frequencies for the electromagnetic resources of a task force.

- Resource allocation (timescale: minutes to hours).

  The reactive allocation of resources, eg aircraft, weapons, sensors, in response to the immediate tactical situation and particularly to any perceived threat.

- Automatic response (timescale: seconds).

  The actions of automatic response weapons (normally close-in missile or gun systems) in situations where the need for rapid response does not allow human involvement in the direct control loop.

The specific naval planning task described in this paper falls into the second of these categories (resource planning) and relates to the generation Naval Flying Programmes. An experimental system has been constructed to provide intelligent support to the generation of such flying programmes and is known as FlyPAST (Flying Programme Assignment Support Tool).

Problem Domain

Flying Programmes

A flying programme is a schedule showing the planned deployment of the air assets of a naval task force. Within the Royal Navy, the assets are of two principal types: fixed wing, consisting exclusively of Sea Harriers, and rotary wing, which includes a variety of helicopter types. Harriers are deployed primarily on combat air patrol (CAP) to provide defensive measures aginst air attack, whereas helicopters are deployed on a wide range of activities including surface search, anti-submarine warfare (ASW) and general delivery services. In an RN context, a task force may consist of up to 20 or 30 warships and auxiliaries, of which only one (except in very unusual circumstances) is likely to be an aircraft carrier. Whereas the Harriers will be flown exclusively from the carrier, the helicopters will be divided amongst many warships and auxiliaries, spread over a wide geographic area.

The aircraft are carried in order to fulfil missions. Typical examples might be expressed in the following form:

> "Maintain two helicopters on surface search continually from 0800 to 1130"
>
> "Maintain four Harriers on combat air patrol over the hours of dawn and dusk"
>
> "Maintain one helicopter on 5-minute alert during daylight hours"
>
> "Provide a helicopter for mail delivery sometime during the morning".

Typically, some twenty such missions may need to be undertaken in a 24-hour period, each involving several aircraft.

The assignment of aircraft to missions is controlled by a multiplicity of constraints. Constraints can relate to ships, aircraft or aircrews, or to any combination of these; some are obvious, such as the fact that an aircraft may not be in two places at the same time. Examples of constraints involving ships are the fact that deck space is limited and only a certain number (usually one) of aircraft may take off or land at the same time, and the fact that ships should turn into wind to launch aircraft, thereby saving the aircraft's fuel and increasing its endurance at the expense of disrupting the progress of the task force. Constraints relating to aircraft concern their suitability to the mission, time taken to reach their patrol station, endurance versus weapon load, and mandatory restrictions of flying times and maintenance periods. Similar rules exist for the aircrews concerning flying hours, rest periods and qualifications for certain tasks such as night flying. An example of a more complex relationship is the fact that an aircraft parked or undergoing maintenance on the deck of a carrier will shorten the effective runway, causing other aircraft to use more fuel on launching thereby reducing their endurance.

These constraints render some allocations impossible and some merely undesirable; such constraints are termed 'hard' and 'soft' respectively. Hard constraints tend to be the physical limitations on the allocation, such as the facts that aircraft cannot be flying and on alert simultaneously, and that deck space must be free for launch and recovery. Soft constraints can be thought of as preferences covering, for example, the many complex rules which determine how long aircrew can fly and how long they should be kept at various stages of alert. These rules are adhered to whenever possible but in times of emergency they may be relaxed. For instance, during the Falklands War, flying times were extended considerably [2]. The relaxability of constraints is actually a continuous spectrum, rather than a hard-soft dichotomy. For example, the limitations on flying hours for aircraft must be far more strictly maintained than the similar rules for crew.

Manual Plan Generation

The problem of generating a flying programme is one of allocating resources (ships, aircraft, aircrew) to meet the requirements (missions to be flown) in such a manner as to satisfy the multiple constraints. The input to the problem is a scenario consisting of the resources and the requirements, and the output is a flying programme showing the allocation of aircrew and aircraft to missions and the launch and recovery times for each flight, each mission consisting of several flights. The current manual generation of flying programmes is rendered feasible through the use of heuristics which have been developed over the years by the human experts. These heuristics take the form of the 'templates' by which the user specifies the missions. These templates represent 'compiled expertise': the user knows that, in previous scenarios, a certain use of aircrew and aircraft has provided efficient coverage of particular kinds of mission.

A typical flying programme, shown in Figure 1 (although not generated manually), illustrates the use of these templates. The horizontal axis shows time from 1 to 24 hours, with sunrise and sunset icons in the appropriate locations, and the vertical axis shows the available aircraft grouped into squadrons. The plan shows a 24-hour schedule covering four missions. Although the individual missions are not obvious to the layman, the naval officer can easily recognise them from this use of templates. Two squadrons are shown, one of Harriers and one of helicopters, being used to provide combat air patrol and ASW surface search coverage. Each arrow indicates one aircraft in flight together with its call-sign. The top three rows of flight arrows illustrate the use of a template used in this case to maintain a minimum of two Seaking helicopters on station between 0700 and 1700, taking into account their transit times. Harriers 1 to 8 are scheduled according to a template used to maintain appropriate coverage for a combat air patrol, the differences in the templates employed resulting primarily from the differing aircraft characteristics

Currently the task of preparing flying programmes is performed manually, taking a senior officer several hours to produce a 24-hour schedule for a fleet of ships. This, however, is only part of the problem: military scenarios are, by their nature, both dynamic and unpredictable. The scenario on which the plan was based is likely to evolve both in terms of the requirement, owing to the changing nature of the threat, and in terms of the resources, as these develop faults or are lost in combat. Thus a considerable amount of replanning is involved. Not only does this form a significant factor in

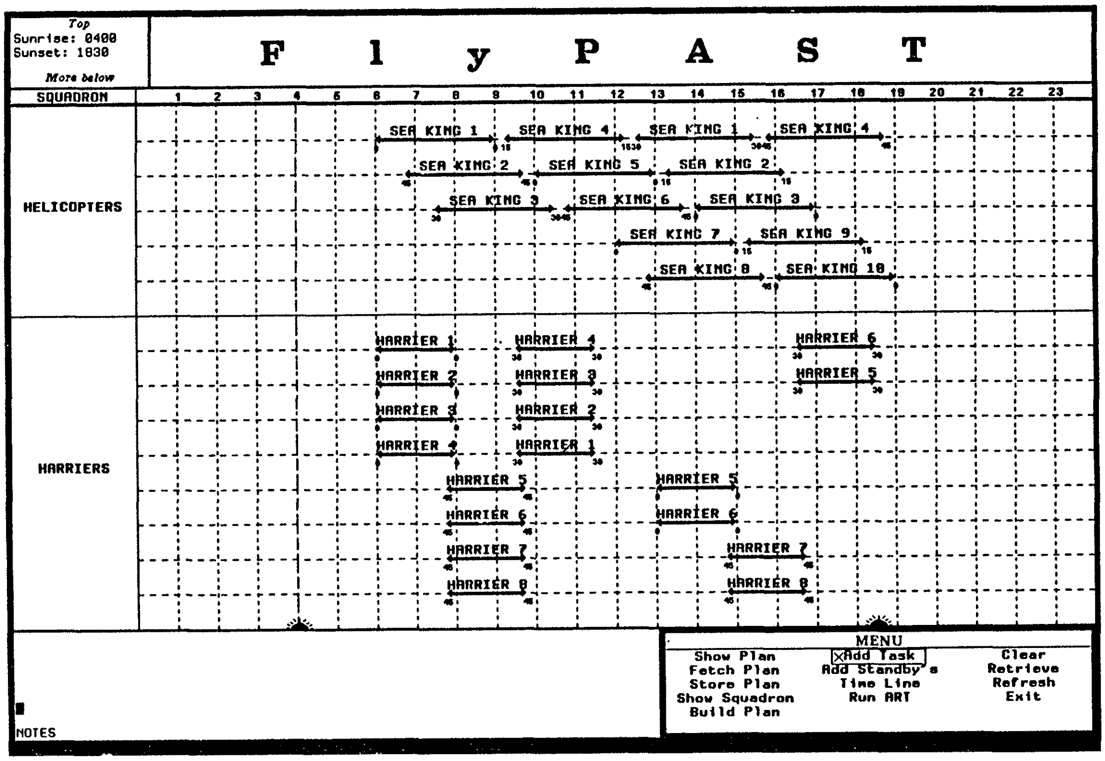


Figure 1 : A Typical Flying Programme

establishing the need for automated support for this task, but it also raises key issues in the design of the system.

The intended role of the FlyPAST system is not to replace the human planner but to provide him with automated support that will take care of the more obvious aspects of planning (and replanning), leaving him free to add his superior knowledge and expertise in an interactive dialogue with the system.

Resource Allocation and Activity Scheduling

It is interesting to consider the differences between the type of problem being considered here (resource allocation) and many activity scheduling problems. Both types of problem are sometimes referred to as 'planning' by different people but the resource allocation problem (while it does produce a schedule) is more like a design process than the scheduling process (which often results in PERT-chart types of output). Many planners (and planning programmes) address activity-scheduling problems and operate on the basis of precedence ordering of the various tasks to be performed [3,4]. The focus there is on time-ordering and precedence-relationship types of constraints. Such problems are typified in the AI world by the 'blocks world' example.

The flying programme, on the other hand, does not require that one flight be scheduled before another, and there are few constraints on the allocation of a particular resource to meet a particular requirement. Rather the allocation process is very much like a design problem, where there are a multitude of alternatives available. Most of the constraints come into operation only after a major part of the entire programme has been put together. Although no aircraft may take off on a mission until it has returned from the previous one, this precedence relationship does not become operational until missions have been scheduled for that aircraft. That is, there are no precedence relationships for that aircraft which form part of the problem input. The constraints are global and apply to the whole programme rather than to individual requirement-resource pairs.

The FlyPAST System

Development Environment

An investigation [5] was undertaken to identify the most suitable knowledge-based programming environment on which to build FlyPAST. At that time (December 1984), ART^{tm} (the Automated Reasoning Tool from Inference Corporation) [6] appeared to be the only commercially available supported product which could provide the facilities needed to tackle problems of planning and resource allocation. Features which were considered particularly useful were ART's 'viewpoint'^{tm} mechanism and the associated truth maintenance system [7] which enable hypothetical reasoning about different contexts. Thus it is possible to hypothesize different assignments of resources to requirements and carry forward the reasoning in parallel (conceptually). ART also provides schemata (a frame-based representation) and forward- and backward-chaining inferencing mechanisms.

The support environment for ART was chosen to be Symbolics 3600^{tm} Lisp-machine. This provides a powerful facility for interactive program development enabling rapid prototyping and providing a number of additional features, one of these being 'Flavors'^{tm}, an object-oriented programming capability. Although ART offers an impressive graphics capability (Artist^{tm}), this was not at the time sufficiently developed to meet the needs of FlyPAST, and so the FlyPAST graphics interface was programmed using Flavors. Thus the sytem has essentially two main components: an interactive graphics interface built using Flavors, communicating with a rule-based planner built using ART.

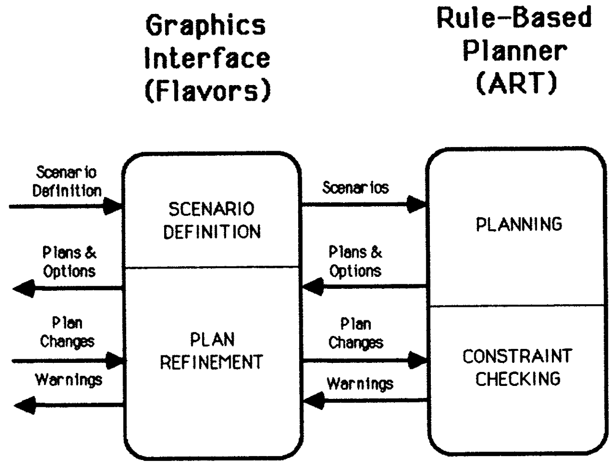


Figure 2 : Flypast Stage-1 Design

Stage-1 System

An initial laboratory prototype or 'demonstrator' has recently been completed, the design of which is shown in Figure 2. It supports the following functions, the first two within the graphics interface and the remaining two within the ART planner:

- Scenario definition
- Interactive plan refinement

- Planning
- Constraint checking

These are described below.

Scenario definition. The scenario definition phase provides a menu-driven facility for constructing input scenarios for FlyPAST, the scenario having two components: resources and requirements. The resources are chosen from a menu of British navy ships. These ships have default options for the number of air resources and aircrew usually carried. The user may elect to accept the default values or edit the ship description in order to change these values. The requirements are specified by the user by choosing from a list of 'templates' which can be used to define a variety of different mission types. These include combat air patrols, anti-submarine warfare, surface search, alert states and helicopter delivery services (HDS). The user can specify values to indicate, for example, the duration of the mission, the number of aircraft to be kept on task at one time, the start time and the preferred aircraft type. The completed scenario definition is translated into the ART schema representation and is passed to the rule-based planning module.

Planning. The total search space for this problem would consist of allocating every possible crew to aircraft they are qualified for and every possible combination of resources (aircrew and aircraft) to each flight of each mission for which it was suited. This is too large a space to be explored totally in a reasonable time. Several techniques are therefore used to prune this search space.

The primary method of pruning the search space is by the use of heuristics, referred to by Mitchell [8] as 'taking advantage of design disciplines to aid in constraint management'. These are the well tried and tested templates formulated and used by the human planners, and the use of such heuristics in FlyPAST provides two major benefits. Firstly, they cut down the size of the search space considerably. Secondly, and possibly more importantly, they result in the plans produced by the automatic planner being recognisable by the human expert. This causes him to have more confidence in the system than he might if it produced strange, though possibly efficient, schedules. It is vitally important in a system such as FlyPAST, which is intended to be highly interactive, that it should work in a way which is familiar to the end user.

Secondly, the missions are planned independently in priority order: combat air patrols, non-HDS helicopter missions, HDS missions. This ordering results in the most constrained missions being planned first. For instance, only Harriers can perform combat air patrols and they cannot carry out any other missions, so they are allocated first. The helicopter delivery services are planned last as they tend to be one-off flights and of low priority. As each mission is planned, new constraints are generated indicating the use that has been made of aircrew, aircraft and deck space. These new constraints are passed on to the next stage of planning for the next-priority mission.

Finally, the rule-based planner uses a best-first search to choose the assignments of resources to requirements. This means that at each decision point the possible allocations of resource to requirement are listed in some order of preference and the three 'best' are chosen for expansion. The best of these is actually expanded further; the other two are kept as back-ups in case of constraint violation. During the planning, backtracking is currently only used locally. The best-first search will select the best of the available resources to allocate to a particular mission and, if this choice results in constraint violations, the planner will backtrack and choose one of the other resources. However, if no choice of resources within that mission can meet the constraints, no attempt is made to backtrack to previous mission plans to make a different choice point there.

The use of the word 'backtracking' may be slightly confusing. Backtracking in the sense as it is used by many researchers in knowledge-based planning involves destroying a chain of reasoning which has been built up down the search tree, retracting the inferences that arose from the choice made at a particular decision point. In this sense backtracking is not meaningful in ART terms if the search tree is represented by viewpoints, as the chain of reasoning is not destroyed (although 'poisoning' may result in a particular leaf node being destroyed - or more accurately never being allowed to exist). However, the search strategy may 'back up' the tree structure to the viewpoint where the choice was originally made and independent lines of reasoning were spawned off. If all lines of reasoning are set off at once then indeed backtracking is not meaningful, but this may be inefficient in some cases. It may be necessary, as in FlyPAST, to expand only one line of reasoning at a time for efficiency's sake and then only when that line fails does the strategy expand an alternative line; nothing is destroyed. de Kleer [9] points out that the advantage of not destroying reasoning chains during backtracking is that alternative, incompatible, choices can be compared. Backtracking in its more usual sense requires a globally consistent set of assertions.

The generation of the search space and its subsequent pruning are easily represented by the 'viewpoint' mechanism provided with ART [7]. The viewpoint facility enables new nodes in the state tree to be hypothesised; plan states which prove undesirable are removed by the ART 'poisoning' facility which also ensures that these states are never recreated. A more detailed description of the use of the viewpoint mechanism within FlyPAST is given by Nielsen and Gadsden [10].

Currently, when a particular mission fails, the operator is informed of this fact and the planner continues to try and plan the next mission. Future versions of FlyPAST will give the operator a list of options as to which constraints could most usefully be relaxed in order to meet the full requirements. This will require the constraints to be represented as objects in their own right, together with the relationships between them.

Interactive plan refinement. The automatically generated plan is displayed to the operator in a format resembling as closely as possible that with which he is already familiar from his manual planning experience. In fact Figure 1 is a printout of the Symbolics display of an automatically generated plan. Most of the displayed items are mouse sensitive; for example, selecting one of the flight arrows provides additional information about each flight by bringing up a menu giving details of individual aircraft weapon and sensor fits. In addition, a vertical 'time-line' may be invoked at any point and details of aircraft in flight at that time requested.

It is an essential part of FlyPAST that the operator should have the final decision on flying programmes. This implies that he should be able to modify the programmes generated by the system; this modification can occur at two different levels. On a local level, facilities are provided by the plan refinement and constraint checking modules to allow individual flights to be removed and added, or for the aircraft used in a flight to be changed. More global changes to the scenario as a whole require the facilities of the scenario redefinition and replanning modules which form part of the current work programme and are described later.

When local changes are made by the operator, the information about those changes is passed to the rule-based planner where the constraint checker comes into action. There may, however, be a problem when local changes are made by the operator. The system will not be able to 'understand' why these changes have been made and therefore it may subsequently be difficult for the system to replan in a sensible manner. Thus it may be necessary for the system to be allowed to revert to a machine-generated plan before any replanning can take place.

Constraint checking. The constraint checker will take local changes made by the operator and check them against the constraints on the system. It can then issue warnings to the operator about constraints which have been violated. In this case no attempt will be made to stop the operator breaking constraints as there may well be valid reasons for doing so, but he will be reminded of the consequences of his actions.

Current and Future Work

FlyPAST Evaluation

An evaluation of the stage-1 demonstrator has been undertaken from two standpoints. The first is that of the user: a brief assessment by the domain experts, i.e. experienced naval flight schedulers, indicated a greater need for user interaction and consultation during the generation of the initial schedule than had previously been requested. Additionally, many more constraints, some quite complex and nebulous, were identified over and above those originally identified. The second was a technical evaluation involving running FlyPAST against scenarios of varying degrees of complexity. Performance aspects considered included its ability to derive a correct plan, the efficiency of the search, in terms of branching and wasted search effort, and which relevant constraints, if any, were not used.

Current and future work, involving both research and development, is intended to improve the functions and performance of the stage-1 FlyPAST system, and to provide additional capabilities in the form of further functional components.

Basic FlyPAST Improvements

Based on the insight gained both into the problem of flying programme generation and into planning methods and technology, extensive redesign of the basic FlyPAST system is now underway. Improvements will include more interactive plan generation allowing user intervention between the planning of individual missions, a much bigger set of constraints, identification of options for constraint relaxation in the event of failure, and a more intelligent planner.

The current planner suffers from two major limitations. Firstly, it does not backtrack beyond the scope of the current mission; for example, if it fails in an attempt to schedule a helicopter mission, it does not go back and replan the Harriers in order, say, to free up some deck space. Secondly, its backtracking is blind (sometimes referred to as 'chronological') in that it will merely backtrack to the previous decision point in the search tree and continue its search, irrespective of whether or not that decision point can have any influence on its reason for failure. The new version will include a form of dependency-directed backtracking which will allow backtracking to the most appropriate decision point on constraint violation. This will involve keeping a record of the reasons for and implications of each decision so that, for example, if a plan fails for lack of deck space then the choice which gave rise to the constraints on deck space can be altered. This is more likely to result in a successful schedule than 'unintelligent' chronological backtracking.

The new approach taken in the planning stage must also provide a more integrated approach to replanning. In order to handle the constraint-relaxation advice from the planner and also intelligent replanning, it will be necessary to modify the knowledge representation to make it more constraint based. The work to date has given a better understanding of the nature of the problem and and of what the available knowledge representations can do. It is now recognised that the design of flying programmes is a constraint-based problem and the new knowledge representation will reflect that more closely. This will simplify the implementation of the remaining modules.

Additional Modules

FlyPAST is also being extended by the addition of four further functions. These are:

- Scenario redefinition
- Replanning
- Plan execution simulation
- Plan monitoring

The overall functional design is shown in Figure 3.

Scenario redefinition. Larger scale changes made by the operator are more likely to take the form of a redefinition of the scenario. This will take into account the changing requirements and resources which may arise during the execution of the original plan; for example, a new mission which must be fulfilled or an aircraft unexpectedly being out of action for several hours. The operator will be given facilities to specify the changes to the scenario, and the formulation of a modified plan will be carried out by the replanning module.

Replanning. These scenario changes will not be passed to the original planning module as total replanning could well result

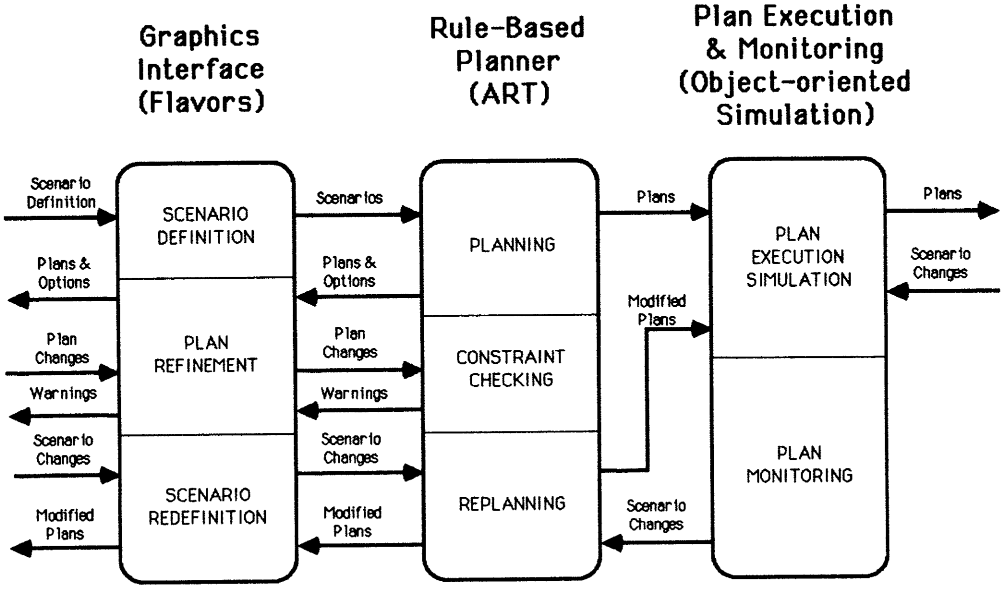


Figure 3 : FlyPAST Overall Design

in a new schedule which bears no resemblance to the original one. This could imply drastic changes to crew and aircraft schedules which would be disruptive. What is usually required in these circumstances is a 'fix' to the original plan which will cover the new scenario but eventually feed back into the original schedule without too much disruption to the basic schedules for men or machines. What is returned to the user therefore is a modified plan and not a totally new one. To achieve this the replanner must be aware of which flights support missions which are still valid and attempt to leave these undisturbed while using the aircraft made available by the changed scenario to fill in where necessary.

Plan execution simulation. The modules described so far form the basic facilities of the FlyPAST system to enable planning and replanning. The remaining two modules will provide facilities for simulated plan execution and monitoring. Once a plan has been generated it is intended that a plan execution module (probably written in an object-oriented simulation language) will provide a graphic simulation of that plan. The operator will be provided with facilities to interact with the simulation in order to make changes to the scenario (e.g. by sinking a ship or ordering a new mission). These changes will be similar in nature to those handled by the scenario modification module.

Plan monitoring. The plan simulation will be monitored by the plan monitoring module. When changes are made to the scenario, these will be passed to the rule-based replanning module which will issue modified plans to the plan execution simulation. This will enable the operator to study 'what-if' situations in relation to his chosen plan and to develop more robust and flexible plans.

Future Research Issues

Replanning. One of the most important aspects of FlyPAST is the support it will give to the operator in replanning. The preparation of an initial plan is both less demanding and less time critical. Replanning is essential and it is important that this should be rapid. It may well be necessary for the operator to provide additional guidance to the system in order to produce adequate response times. This is acceptable in a system whose aim is to provide support to a human operator rather than totally replace him. It is more important that FlyPAST should provide timely and useful support for a real-world problem than to automate the process totally but take too long to do the job.

What is 'optimal'? Another issue which has arisen is discussion of what is meant by 'optimal' in this context; there is no notion of there being an optimal plan. There are many legal solutions which would be acceptable to most naval experts, but it is unlikely that they would produce the same plan given the same requirements and resources. It is therefore difficult to provide the planner search strategy with a measure of effectiveness in order to assess partial plans. It is believed that the measure of effectiveness should be the flexibility of the plan, given the importance of the replanning process. A plan which meets the current situation adequately but gives no room for later modification to meet likely new requirements or changing resources is of little use.

What is 'explanation'? It is often quoted that knowledge-based systems should be capable of giving 'explanations' of their reasoning. It is difficult to envisage what is meant by explanation in the FlyPAST context (or indeed in any planning system). To some extent the explanation is in the display of the plan (which meets all the given constraints) and, with FlyPAST, also in the fact that the heuristics used to derive the plans are the compiled knowledge of the human experts. This latter point results in the plans produced being recognisable to the human and he therefore does not question their rationale any more than he would question that of another human planner. Even if the plan is not exactly the one he would have produced himself, if it meets the constraints (and is flexible) then it is acceptable; an explanation in the conventional sense would probably not be of any use. By allowing the operator to change the plan we allow for individual idiosyncrasies to be incorporated. The problem was outlined above of how the machine can 'understand' the local changes made to the plan by the operator. This highlights the need not only for the man to understand the machine's rationale but for this 'explanation' to go both ways.

Conclusions

FlyPAST as it exists to date has provided a good basis for future work. It is has given us a sound understanding of the nature of the problem and what should be possible to implement. It has enabled us to arrive at the design described in this paper and to identify the strategies to be attempted. This will imply significant design changes to the existing model in order to provide a more suitable knowledge representation for the remaining implementation. It is hoped that another year's effort will see the majority of the ideas outlined here implemented at least in part.

References

1. Lakin, W.L. and Miles, J.A.H., "Intelligent Data Fusion and Situation Assessment", ibid., 1986.

2. "The Falkland Campaign: The Lessons", Cmnd. 8758, HMSO, London, December 1982, (para. 224).

3. Tate, A., "Generating Project Networks", Proceedings of IJCAI-77, Boston, Mass., 1977.

4. Vere, Steven A., "Splicing Plans to Achieve Misordered Goals", Proceedings of IJCAI-85, Los Angeles, Calif., 1985.

5. Brown, David R., "Planning System Survey: A Review of the State of the Practice", Task Report Project No. 7306, SRI International, Menlo Park, Calif., December 1984.

6. "ART Programmers' Reference Manual", Inference Corporation, Los Angeles, Calif., 1985.

7. Clayton, Bruce D., "ART Tutorial", Volume 3, Inference Corporation, Los Angeles, Calif., 1985.

8. Mitchell, Tom M. et al, "A Knowledge-Based Approach to Design", IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume 7, No. 5, September 1985.

9. de Kleer, Johan, "Choices Without Backtracking", Proceedings of AAAI-84, University of Texas at Austin, August 1984.

10. Nielsen, Norman R. and Gadsden J.A., "Knowledge-Based Scheduling in FlyPAST", submitted to 3rd IEEE Conf. AI Applications, Orlando, Florida, February 1987.

Acknowledgements

The ideas presented in this paper are the results of many discussions with domain experts in the Royal Navy, and with many fellow researchers. Particular thanks are due to Bob Irving, David Mott, Connie Brede and her colleagues in SRI.

An Experimental Interview System for Multiple Interacting Decision Makers

Robert L. Stewart and Bruce W.Hamill

The Johns Hopkins University
Applied Physics Laboratory
Laurel, Maryland 20707

As part of our work on the problem of knowledge acquisition and representation for distributed command decision aiding, we have developed a computer-based system to support the simulation of tactical problems and the collection, organization, storage, and analysis of data and information obtained from structured interviews involving experienced decision makers and their plans for Naval tactical missions. The purpose of this paper is to describe this system in the context of our research objective and approach, requirements for data and information collection, analysis, and interpretation, and requirements for distributed tactical decision making. Also described are the current status of the interview and simulation system and planned upgrades.

Research Objective and Approach

One possibility for improving distributed decision making at a given level of command, which we are exploring, is for decision-making peers, such as mission area commanders, to use local instantiations of a common knowledge-based decision-aiding system incorporating doctrine and expertise from all mission areas. With such a system of knowledge bases operating in reduced-communication scenarios, each participant could, in principle, develop an improved estimate of the likely response of other system participants to evolving events. Given this concept as a general goal, our objective is to conduct basic research on the critical factors underlying knowledge-based system development, namely, knowledge acquisition, representation, and utilization, with special emphasis on requirements for distributed support of spatially separated decision makers.

In support of this objective, we have focused on the processes of eliciting knowledge from experienced tactical commanders responding to mission area requirements in realistic tactical problems requiring coordination with other peer-level tactical commanders, and representing the elicited knowledge in appropriate structured formalisms that can be stored and used in a computer-based system.

Our approach to these knowledge elicitation and representation issues combines theoretical perspectives from psychology, artificial intelligence, computer science, and mathematical modeling within the framework of a composite model comprising the psychological, computation, and communicative dimensions of the problem space being investigated (see Hamill and Stewart, 1986). Our multidisciplinary approach to this multidimensional problem space necessitated the development of a system within which to structure and organize our effort. This requirement led to the development of the interview and simulation system, which we use to record, organize, and analyze the detailed individual and group plans developed by our subjects in response to mission requirements, and to facilitate comparison of such plans among subjects in our search for relevant scientific hypotheses and performance variables.

By using specified scenario knowledge in the system, we can conduct structured interviews with individual subjects to assist them in developing mission area plans optimized for their particular mission area requirements.

After those plans have been developed and recorded in the system, agents specified in the plans can be defined, instantiated, and tested againest selected critical events in which such agents have to be effective. (See Gilbert and Stewart, this volume, for details of agent instantiation.)

The next step in our approach is to develop an initial model of each subject's planning process. This procedure entails identifying states of knowledge underlying planning choices and decisions and determining knowledge requirements for the interaction of agents defined by multiple decision makers in anticipation of coordination requirements.

A planning process model is also developed for each team of three subjects serving in the roles of mission area commanders as they engage in a negotiation session to develop a joint plan. This planning process model for multiple interacting decision makers thus has components of the individual's own mission area-optimized plans and of the negotiation session interactions. The interview and simulation system serves to support model development by enabling us to document and compare aspects of the several individual and joint plans for each team of subjects.

General System Requirements

The general research support requirements addressed in our design of the multi-node interview and simulation system include tactical problem representation, knowledge acquisition, knowledge representation, knowledge utilization, and an analytical framework.

Tactical Problem Representation

In representing simulated tactical problems, we have tried to maintain a level of realism commensurate with the expected level of expertise of our tactically experienced subjects, while focusing on specific aspects of the problems which we expect to yield scientifically interesting observations, hypotheses, and variables related to distributed decision making issues. To do this we provide each subject with a statement of the battle group mission, an assessment of the threat faced by the battle group as it bears on his specific mission area assignment, and a specification of the resources available to him for the performance of his mission. Documentation of all of these aspects of the tactical problems resides in the interview and simulation system. To concentrate on basic decision making processes and to avoid classification constraints, we have devised a scenario of Red and Blue forces using systems capabilities and values as appearing in the open unclassified literature.

We have further limited the complexity of the problem domain to two opposing task groups without aircraft carrier support and moderate air surveillance assets.

Knowledge Acquisition

The knowledge acquisition process is controlled by the presentation of structured events to which subjects can develop and apply their plans. Subjects' responses to these structured events are probed through interviews in which they elaborate on details of their prepared plans in terms of how those plans support their response to specific aspects of the events. The structure of the set of events, together with the structure of subjects' plans, enables us to classify subjects' responses to the events within representational structures contained in the support system. This permits detailed record keeping for each subject's plan and facilitates comparison of plan features across subjects and between individual and team plans.

Knowledge Representation

The subjects' structured plans and the knowledge elicted during interview sessions are represented in the computer system in several forms, with the particular form being selected or designed to meet requirements for capturing details appropriately. We expect that these forms of representation can eventually be transferred into appropriate representational formalisms, such as schemas and rules, for use in knowledge-based systems.

Template forms. These generalized structures take the form of templates covering the information and knowledge elements we have observed in our subjects' plans, and these have been elaborated in detail for specific aspects of these tactical problems we pose to subjects.

Position plots. Position plots are graphic representations of the relative spatial locations of naval assets under the control of a subject, containing information about distance, speed, and direction for platforms and their sensor and weapon systems.

Narrative information. In addition to the above structured forms of plan and knowledge documentation, we record narrative information which is relevant to the problem and to subjects' plans, but which does not fit into existing structures. Such information includes command organization (both at and above the subjects' positions in the command chain) and emission control plans (including limitations on sensor usage, communications, etc.).

All of these forms of each subject's plans and knowledge for tactical problems reside in the computer system, and they may be displayed in several "windows" of the display at the same time. This enables the subject to see several aspects of his plan simultaneously and to use information in one window to facilitate updating of information in another window. This arrangement can also be used to provide feedback to subjects about the status of their plans with respect to test events from the scenario for a tactical problem. In addition it greatly facilitates our interactions with subjects in the process of acquisition and representation of plan details and knowledge states.

Knowledge Utilization. To model distributed decision processes so as to support computer-based analysis of interactions among separate decision makers, we developed a structure of high level representation languages, multi-tasking network operating system, and a full-protocol local area network. By representing selected problem-solving procedures of individual subjects within this system at one or more of the three warfare commander nodes, we have achieved a limited simulation ability to support procedural and performance analyses of distributed decision processes.

Identification of distributed decision process has focused on the identification of sets of knowledge states and associated procedural steps, defined as *protocols*, that our subjects have exhibited during structured interviews. Where possible, these protocols are functionally decomposed into component subsets of functions assigned by the subject to subordinate individuals or human-computer process. These functional sets, defined as *agents*, are then represented, as possible, using the interview systems. For example, each of two subjects identified circumstances and procedures under which they would authorize their Staff Watch Officer (SWO) to engage in a discourse with the SWO of the other to devise an emergency reorientation of forces. This set of authorized procedures constitute a cooperative protocol, with the watch officers functioning as agents for the two commanders. By representing such protocols and agent processes at several nodes, we can examine performance issues of knowledge utilization.

Distributed Tactical Decision Making Requirements

As our objective is the study of distributed, as opposed to individual decision making, it was necessary to restrict the design of the interview and analysis system to represent events appropriate to distributed problem solving. In addition, the structures used to record subject decision-making behavior in relation to these events must promote identification and possible quantification of neutral, conflicting, and synergistic decision elements among two or more separate subjects. A third restruction necessary to our approach is that the DTDM events and the structures for representing subject DTDM behavior must also support descriptive analysis of our proposed DTDM models. In particular, the interview and analysis system has been designed to represent events requiring coordinated planning with subject plans represented in structures corresponding to the composite model and the Wilensky planning model.

DTDM Events

Our problem domain is the Navy battle group with projected 1990 capabilities as it opposes similar forces. Each unit of the battle group is presumed capable of support of many missions and all mission areas (with varying tactical effectiveness). In this environment all tactics involve coordination among these commanders who, because of their relative expertise, are delegated authority to plan and conduct operations against air, surface, or subsurface threats. Certain events, however, may entail significant alterations of joint plans or allocations of forces to each warfare area commander. Other events, such as the loss or departure of units, may require considerable thought and discourse among commanders to devise an optimal reconfiguration of battle group forces; a luxury of time and open discourse not permitted in a tactical environment. In the short time expected to exist to reorganize forces with limited opportunity for discourse among the three warfare area experts, many suboptimal solutions may be derived, each with some level of conflict and synergy for each warfare area objectives. for example, the movement of a ship to fill a gap in anti-submarine surveillance has in consequence the potentially conflicting result of reducing anti-aircraft surveillance in the vacated area.

For purposes of our research, therefore, scenario and planning events have been selected with particular regard for their associated opportunities for decision conflicts and synergies.

Structures to Identify DTDM Conflict and Synergy

To assess distributed decision conflict and synergy requires representation of changes of state. Neutrality, conflict, and synergy of plans or actions may then be determined by comparison between separate and coordinated plans for the same time period, or between current and projected capabilities. In other terms, distributed planning conflict and synergy may be measured if representations of independent plans can be compared to each other and to the unified plan derived through discourse among the principals. Similarly, the representations of change of state from a unified plan to a new plan (with new structures) may permit evaluation of reconfiguration processes in terms of associated conflict and synergistic value.

In our approach, independent and unified plans are represented in common structures via separate interactions with our subjects. Issues associated with development of a unified plan are addressed using representations of individual plans and a structurally equivalent representation of the unified plan. Issues associated with reconfiguration are addressed using representations of current and projected plans.

Relevance of Test Events to DTDM Models

Given proper selection of DTDM events and structures to represent change-of-state conditions, our approach further requires selection of events that facilitate functional analyses with respect to proposed models. Thus, events prompting actions at graduated levels of abstraction and processing, or that may be decomposed into psychological, computational, and communicative elements are required. Examples of these 'critical' events' include:

1) loss or introduction of a major mission capability,

2) reversal of major intelligence estimate, and

3) emergency command reorganization.

Interview and Analysis System Design

The design of our DTDM interview and analysis system incorporates LISP and C computer programs, the UNIX operating system (with network sockets), SUN-3 high resolution monitors, and a local area network of computers. To represent and emulate selected decision making processes of three separate commanders the system is designed to permit concurrent processing at several nodes and dynamic interprocess communication at all levels of the OSI protocol.

Figure 1 is an overview of the resulting design for an open system of interview and analysis nodes. Each of three nodes is depicted with a separate user interface to display information and to service requests or orders from the user. Knowledge is partitioned in the system into broad categories of global and local data structures. Separate processes for acquiring and updating global and local data are defined.

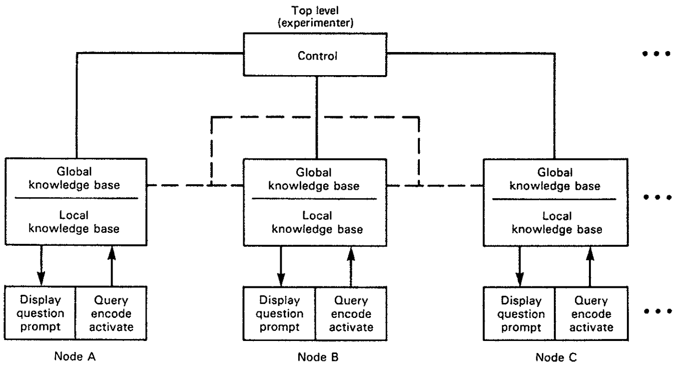


Figure 1: DTDM Interview and Analysis System

Each node is capable of communications with all other nodes with communication effectiveness determined by experimenter control of error functions. Overall control of the system is exercised through remote modification of data values or processing constraints by the experimenter.

Status of the Interview and Analysis System

The interview and analysis system currently supports multiple instantiations of a common structure for event presentation, recording of interactions of subjects, and the representation of generic warfare area commander processes and variables. This structure is similar in form to schema representations in cognitive science, and is constructed using Franz LISP as extended by Allen et. al. (1983a) to represent processes and variables associated with

1) narrative text,

2) property lists,

3) directed graphs,

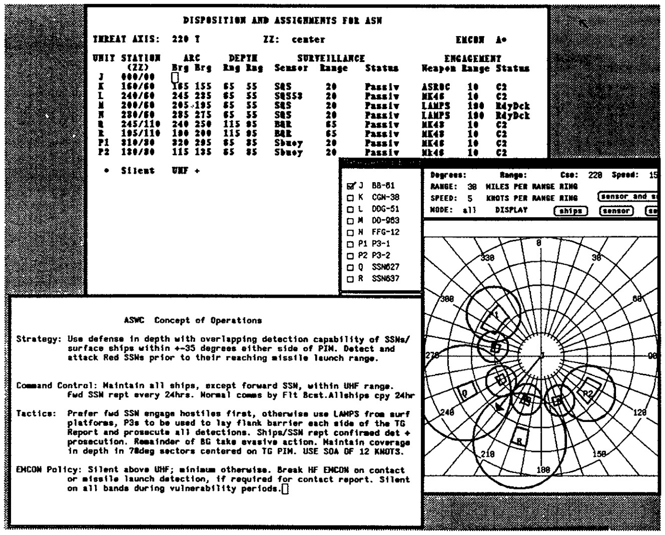
4) schema with inheritance, and

5) production system objects.

To permit dynamic interaction of two or more instantiations, separate Flavors objects for communications between LISP processes and associated C/UNIX code have been developed. This capability is presently restricted to a static network definition, and, thus, dynamic reconfiguration of communication connectivity is not supported. Static network definition does, however, permit analysis of a limited set of interactions among nodes.

To avoid use of LISP syntax at the user interface, screen oriented templates are presented for fill-in by the user. Resulting data entered by the user is then converted by a separate program into LISP expressions appropriate to the data. For example, a tabular array of position, sensor, and weapon assignment entered as a planned disposition of forces by a subject is converted internally to a directed graph of property lists in LISP.

Figure 2 is an illustration printed from a SUN-3 computer monitor depicting three windows of planning information typically provided by our subjects. The graphical window depicts spatial orientation of the assigned ships, aircraft, and submarines in terms of range (note range scale) and bearing (in true degrees) from the arbitrated center of the formation (known as 'ZZ'). The other windows provide textual records of the subjects' concept of operations and numerical stationing, surveillance, and weapon engagement assignments. To aid in the development of plans by our subjects, a graphical tool, analogous to the *maneuvering board* used by Navy tacticians, has been developed. This tool allows the user to lay out candidate dispositions of forces, overlay sensor and weapon envelopes and calculate time and distance intercepts to other forces. Using this project and test tool also promotes rapid evaluation of subject planning.



DISPOSITION AND ASSIGNMENTS FOR ASW

THREAT AXIS: 220 T   ZZ: center   EMCON A*

UNIT	STATION (ZZ)	ARC Brg	ARC Brg	DEPTH Rng	DEPTH Rng	SURVEILLANCE Sensor	SURVEILLANCE Range	SURVEILLANCE Status	ENGAGEMENT Weapon	ENGAGEMENT Range	ENGAGEMENT Status
J	000/00										
K	160/60	165	155	65	55	SQS	20	Passiv	ASROC	10	C2
L	240/60	245	235	65	55	SQS53	20	Passiv	MK46	10	C2
M	200/60	205	195	65	55	SQS	20	Passiv	LAMPS	100	RdyDck
N	280/60	285	275	65	55	SQS	20	Passiv	LAMPS	100	RdyDck
Q	245/110	240	250	115	95	BQR	65	Passiv	MK48	10	C2
R	195/110	190	200	115	95	BQR	65	Passiv	MK48	10	C2
P1	310/80	320	295	65	85	Sbuoy	20	Passiv	MK46	10	C2
P2	130/80	115	135	65	85	Sbuoy	20	Passiv	Mk46	10	C2

* Silent UHF +

☑ J BB-61
□ K CGN-38
□ L DDG-51
□ M DD-963
□ N FFG-12
□ P1 P3-1
□ P2 P3-2
□ Q SSN627
□ R SSN637

Degrees: Range: Cse: 220 Speed: 15
RANGE: 30 MILES PER RANGE RING
SPEED: 5 KNOTS PER RANGE RING
MODE: all DISPLAY

ASWC Concept of Operations

Strategy: Use defense in depth with overlapping detection capability of SSNs/ surface ships within +-35 degrees either side of PIM. Detect and attack Red SSNs prior to their reaching missile launch range.

Command Control: Maintain all ships, except forward SSN, within UHF range. Fwd SSN rept every 24hrs. Normal comms by Flt Bcst.Allships cpy 24hr

Tactics: Prefer fwd SSN engage hostiles first, otherwise use LAMPS from surf platforms, P3s to be used to lay flank barrier each side of the TG Report and prosecute all detections. Ships/SSN rept confirmed det + prosecution. Remainder of BG take evasive action. Maintain coverage in depth in 70deg sectors centered on TG PIM. USE SOA OF 12 KNOTS.

EMCON Policy: Silent above UHF; minimum otherwise. Break HF EMCON on contact or missile launch detection, if required for contact report. Silent on all bands during vulnerability periods.

Figure 2: Illustration of DTDM Planning Representation

Subject Rule Bases

To augment representation of subject decision making processes, the system permits the incorporation of individual rule bases using the production system objects developed by Allen et. al. (1983b). This production system, known as YAPS, defines rule sets as objects. Thus, any set of processes can be isolated in terms of a rule set. Using this technique, we are developing rule sets corresponding to specific DTDM protocols of our subjects. These rule sets may contribute to separate development of expert support systems in future work. Their use presently is restricted to augmentation of representation of subject processes and as instances of possibly generic methods associated with schema representation of a warfare area commander.

Planned Upgrades

The major requirement for upgrade of the interview and analysis system is for improved representation of observed DTDM processes in the dimensions and levels of abstraction of our composite model. Accordingly, we will emphasize empirical extension and validation of structures over efforts to improve simulation or emulation capabilities. Of particular interest is the prospect of elucidating a comprehensive representation of a warfare area commander schema in terms of empirically validated processes.

References

Allen, Elizabeth M., Randall H. Trigg, and Richard J. Wood. 'The Maryland Artificial Intelligence Group Franz Lisp Environment'. College Park, Maryland: Department of Computer Science, University of Maryland. TR-1226, 1983a.

Allen, Elizabeth M. 'YAPS: Yet Another Production System'. College Park, Maryland: Department of Computer Science, University of Maryland. TR-1146, 1983b.

Gilbert, J. M., and Stewart, R. L. 'The definition, implementation, and control of agents in an interview system for distributed decision making'. (in this volume.)

Hamill, B. W., and Stewart, R. L. 'Modeling the acquisition and representation of knowledge for distributed tactical decision making'. *Johns Hopkins APL Technical Digest*, 1986, 7 (1), 31-38.

A CHANGE IN SYSTEM DESIGN EMPHASIS: FROM MACHINE TO MAN

M. L. METERSKY
JOAN M. RYDER

NAVAL AIR DEVELOPMENT CENTER, WARMINSTER, PA 18974-5000
PACER SYSTEMS, INC., HORSHAM, PA 19044

INTRODUCTION

Historically, even though the majority of systems designed and fielded have met their specifications, they often have not exhibited predicted performance. We believe this occurs because the role played by the human in the system, particularly in terms of decision making, has not been adequately considered in the design process. This inability to perform as expected is prevalent in weapon systems and exacerbated with command and control (C^2) systems. It is more pronounced as the level of decision dependency increases. Regardless of decision level dependency, current design philosophy ignores the human's role or relegates it to a minor supporting role. We propose that it is necessary to reorient the system design approach, especially for C^2 systems, to stress the decision making function. As a consequence, system performance potential could be maximized by proper emphasis of the role played by humans.

In the past, and to a large extent even today, the emphasis in system design has been on defining hardware requirements. In many cases, advances in hardware technology, not the ability to meet mission requirements, were the driving factor that determined the need for upgrades to or replacement of a weapon system. Even though software has increased in importance and percentage of cost in system development, it is still philosophically considered as a means to facilitate hardware performance. This thinking has had an adverse effect on system performance by relegating decision requirements, which can be derived directly from mission requirements, to a minor or nonexistent role in system design. It is the premise of this paper that system design should be dictated by decision requirements since decisions humans make determine how well and to what degree a weapon system's inherent capabilities will be utilized.

In the decision-oriented approach proposed, the system is considered not as a package of hardware and software capabilities integrated to fulfill well-defined mission functions, but rather as a decision making system composed of three interacting elements -- hardware, software and personnel. The design approach based on this viewpoint begins with mission requirements. From this, decisions necessary to perform specified mission functions, and hardware and software needed to support them, are then defined. This design approach focuses on maximizing the decision-making ability of an entire system by viewing the three system components as complementary and by integrating their capabilities synergistically.

RECENT TRENDS AND POTENTIAL SOLUTIONS

Recent trends, including more and better sensors, improved communication links and advances in data processing technology have produced more data and associated improvements in processing capability. However, large amounts of data cannot be assimilated rapidly enough for timely decision making. In addition, operational situations have become more complex and faster moving, creating the need for personnel to have expertise in multiple warfare areas and to make decisions in ever shorter time periods.

These developments, coupled with limited human cognitive processing capacity, necessitate the development of systems with a higher degree of integration and automation. The degree of integration and automation required can only be achieved by newly evolving technologies and techniques. The following three elements together may provide a technology solution to the data overload problem:

- artificial intelligence (AI) technology
- human-computer interface (HCI) technology
- decision augmentation orientation

AI technology will play a large role by providing automatic and augmented functional support. AI is a set of techniques that can be employed to develop systems that simulate human cognitive functions, such as problem solving, decision making and information processing. Advances in AI are occurring rapidly and by the 1990's, AI will be able to perform many decision functions currently performed by humans.

HCI technology deals with the methods by which a human and computer interact. It includes methods of information presentation, data entry, and function sequence control. By careful structuring of the interface and exploitation of innovative HCI techniques, such as color graphics, automated voice recognition and synthesis display windowing, etc., information transmission can be improved, and user learning time and processing demands reduced.

The proposed system design approach is facilitated by introducing decision augmentation. Decision augmentation software is designed to satisfy human information requirements given a particular mission and at some level perform the data to information transformation. Information is defined here as data processed to match the cognitive capability of the human and directed toward a specific decision task. Decision augmentation software is also designed to quantify information uncertainty and assist in using uncertainty in decision making. The decision augmentation orientation focuses on the decision requirements necessary to successfully accomplish mission objectives. Augmenting classical techniques with AI and HCI technologies can provide the means by which the proposed approach can be implemented in the operational environment.

A decision augmentation framework is most important when decision dependency in the system is high. The first graph in Figure 1 shows that theoretical system performance using classical approaches drops off as decision dependency increases. The second one shows that use of a decision augmentation approach should bring operational performance closer to its theoretical limit. Figure 2 shows a simplified schematic of a C^2 system with and without a decision augmentation orientation. In either case, data including intelligence, environment, sensor, threat, and own force, comes into the system and is processed. Without a decision augmentation framework, the processed data is presented directly to the decision maker. The two shaded boxes are added within a decision augmentation framework. Combining decision augmentation software and innovative information presentation techniques, tailored to the specific decision-making task, should greatly improve decision-making quality. Decision augmentation software can include any or all of the following:

- "expert" knowledge bases and decision rules (AI technology)
- ability to structure the situation into a set of well-defined alternative courses of action
- capability to predict the consequences of each alternative course of action
- rank ordering of each alternative against one or more utility measures

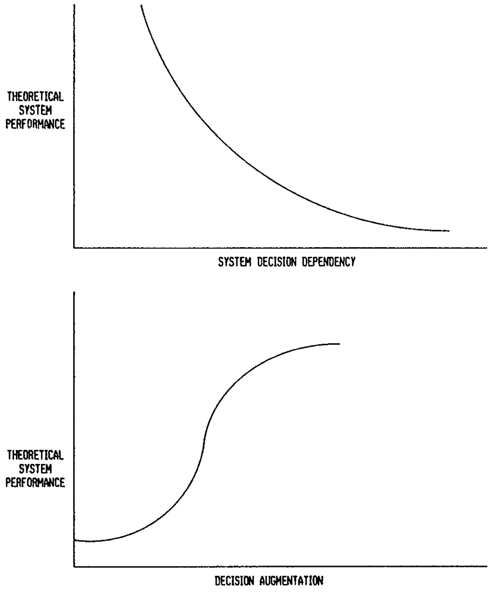


FIGURE 1. SYSTEM PERFORMANCE DEPENDENCIES

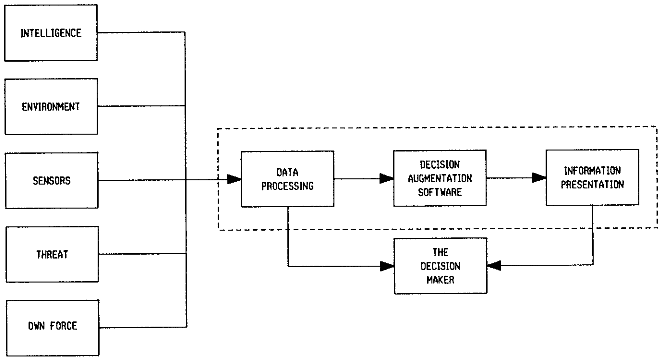


FIGURE 2. DECISION FUNCTION EMPHASIS

Each of these, and other decision augmentation techniques, help to overcome some cognitive processing limitations and biases, and therefore improve the decision-making process. Humans have limited working memory and are not proficient at doing complex numerical calculations unaided. Furthermore, in evaluating alternatives, they often make simplifying assumptions, are biased toward considering solutions from their past experience, and usually do not simultaneously consider more than about three hypotheses. In general, when confronted with excess data, people resort to heuristics which simplify the problem and reduce the amount of data that must be considered. These heuristic biases may, in fact, be erroneous. Decision augmentation, particularly using AI techniques, allows more complex decision rules to be incorporated, more alternatives to be investigated and complex and accurate calculations to be made. This provides the ability to predict action consequences and evaluate their associated utility.

Information presentation, the second additional box, is based on an understanding of human cognition, and includes the following:

- fusion of sensor data, presenting only relevant data formatted to directly support the decisions which must be made
- use of graphics, color and easy access of backup data (HCI technology).

Mechanisms of information presentation directly affect the ease with which the operator can assimilate and use the information presented. For effective task performance, the information displayed must be relevant to the operator's needs. The information should be structured, labeled and coded to highlight information content and relationships. Graphic displays are superior to alphanumeric displays for representing overall relationships among variables, with alphanumeric displays being most appropriate when precise information on specific variables is required. Color is a powerful highlighting technique and can be utilized to draw attention to the most important aspects of a situation. Since the displays should present only relevant information, easy access of backup data should be provided, including specific sensor data to resolve conflicts, or historical data to analyze specific trends.

LEVELS AND TYPES OF DECISION AUGMENTATION

Decision augmentation systems vary in the level of automation involved. They range from completely automatic, in which no operator action is involved, to manual, in which only computational aid is provided for the decision maker. Intermediate levels of decision augmentation include semiautomatic, in which the system provides decision alternatives and recommended courses of action and the operator reviews and accepts or overrides the system, and interactive, in which the operator and software support each other symbiotically. Level of decision augmentation is determined based on, at a minimum, the following considerations:

- operational situation
- data load
- decision importance
- intuition likelihood
- user acceptance (with decision level)

One of the major operational situation factors influencing decision level is time in which a decision must be made. Time to decide is only one variable in the time domain. Three time variables must be minimized to control the operational situation.

$$T_{control} = T_{collect} + T_{decide} + T_{transmit}$$

In order to maintain operational control:

$$T_{control} < T_{crit} - T_{op}$$

where: T_{crit} = critical time (time within which the operation must be executed in order to have the intended effect)

T_{op} = time required to execute the operation

If time were the only factor impacting decision level, their inverse relationship would suggest that the less time available for decision making, the higher the level of autonomy.

Even though time should be considered the most important factor when determining which decision augmentation level to use, all factors listed (and perhaps others) must be considered. Table 1 shows the relationship between the five independent factors and decision augmentation (DA) level in terms of an idealistic environment. It is easy to envision in a realistic environment that conflict among the factors is not only possible but likely, e.g., between operational environment and decision importance. If the decision is to have its intended impact, it may be necessary to have decisions made automatically when the critical time is very short, e.g., activate SDI to maximize boost phase kill, a very important decision.

Table 1

LEVEL OF DA	OPERATIONAL ENVIRONMENT	DATA LOAD	DECISION IMPORTANCE	USER ACCEPT	INTUITION LIKELIHOOD
Automatic	$T_{CRIT} = \varepsilon$	High	Low	*	Low
Semi-Auto	$T_{CRIT} > \varepsilon$	High	Mod	*	Low
Interactive	$T_{CRIT} >> \varepsilon$	Mod	Mod	*	Mod
Manual	$T_{CRIT} >>> \varepsilon$	Low	High	*	High

ε = Short Time
* Variable, a Function of User

The level of decision augmentation desired determines which techniques should be used in implementation. Figure 3 shows where in the requirements analysis process this determination should be made and how it influences information processing and HCI requirements. The type of decision augmentation technique should be chosen to match decision situation requirements. While AI (expert system) techniques will prove extremely valuable, they are not applicable to all situations and should not be considered a panacea. A number of taxonomies for decision situations and decision aiding techniques have been proposed (Keen and Scott-Morton, 1978; Rouse, 1984; Wohl, 1981; Zachary, 1986) which may prove useful in determining the type of decision augmentation technique to use. Zachary suggests a taxonomy for decision augmentation based on the kinds of cognitive support that the various computational techniques provide to human decision makers. For example, deterministic or stochastic process models support the selection of an action from a set of known alternatives by projecting the implications of each alternative based on assumptions about the process. In order to support reasoning processes in ill-structured problems with incomplete or contradictory data, AI techniques can be employed to provide symbolic reasoning capabilities based on a body of knowledge and specific kinds of interferencing procedures. Representational aids such as pictorial or spatial representations help the decision maker develop an understanding of a complex situation. Database management tools allow the user access to subsets of complex data aggregated according to a number of predefined or ad hoc criteria. Other types of tools support other decision making needs.

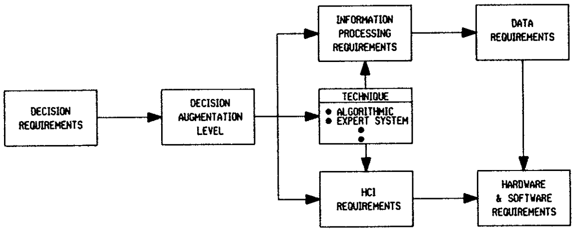

DECISION REQUIREMENTS
DECISION AUGMENTATION LEVEL
INFORMATION PROCESSING REQUIREMENTS
TECHNIQUE
● ALGORITHMIC
● EXPERT SYSTEM
HCI REQUIREMENTS
DATA REQUIREMENTS
HARDWARE & SOFTWARE REQUIREMENTS

FIGURE 3. IMPACT OF DECISION AUGMENTATION LEVEL

Working within the area of tactical command and control, Wohl presents a decision aid taxonomy based on the anatomy of tactical decision processes, called the SHOR model. The SHOR model defines four elements of a decision: Stimulus (data), Hypothesis (perception alternatives), Option (response alternatives), and Response (action). Information processing techniques are identified which are appropriate to each part of the decision process, depending on processing complexity, the time available for the decision and the degree of information aggregation required. Some processing aids suggested include: sensor correlation aids, zoom in/out with variable detail, speeded-up play back of selected battlefield history by target or unit type, knowledge/rule based systems, if/then triggers, English language data base access and pattern recognition aids, among others.

DECISION-ORIENTED SYSTEM DESIGN

The usability of a weapon or C^2 system is ultimately a reflection of its design philosophy. The most commonly applied system design approach begins with a detailed statement of platform mission requirements. Mission requirements are then used to derive functional requirements that will support successful mission accomplishment. The functional requirements are allocated to either hardware or software elements of the system. In practice, this approach emphasizes hardware considerations, with software being designed to facilitate hardware use.

Because informed, timely and "correct" decisions are the key element in system effectiveness, system design should be based on decision requirements. System hardware and software should be specified and designed to support the decision making function. Requirements should be based on operational performance deficiencies rather than on advanced technology, which is the tendency in a hardware oriented design approach.

Figure 4 defines an R&D approach which can be systematically applied to develop a C^2 system (or a weapon system) on a decision-oriented basis. This design approach also begins with the specification of mission requirements. Subsequent steps attempt to develop more specific system and subsystem functions to fulfill the primary mission requirements, as is currently done. Based on the mission analysis, all decisions necessary to perform specified mission functions are defined. The step of defining decision requirements is done early in the requirements analysis, and serves as the determinant of all hardware and software requirements. After the decision requirements are specified, each decision is analyzed to identify the information needed to make the decision. "Information" implies data that has been processed and reduced to just the elements needed for the decision. Next, the data necessary to provide decision-specific information is defined. "Data" refers to data, (e.g., target contact reports) that is needed to derive decision-specific information. These three steps are critical because they serve as the basis for developing all detailed requirements in the system specification.

Once the data necessary to provide decision-specific information have been defined, the hardware and software (both decision augmentation and other support software) requirements are specified. As Figure 4 shows, there are three parallel, but not independent, paths for specification of decision augmentation software, support software and hardware requirements. The emphasis on the decision function suggests that the middle path, that of defining decision augmentation software, is the leading path, First, it is necessary to identify the source of each data element. Then, decision functions and their information requirements must be allocated to organizational units/individuals and subsequently to human (specific organizational units/individual) versus computer (decision augmentation software). The human/computer allocation should be based on the relevant capabilities and limitations of each. Decision augmentation requirements, including HCI requirements should then be derived based on an analysis of the decision problem, and the techniques to assist that particular decision problem.

The left-hand path, that of defining support software requirements, is based on the data necessary to provide decision-specific information as well as the decision augmentation software requirements. The support software might include operating systems, device handlers and data base management systems.

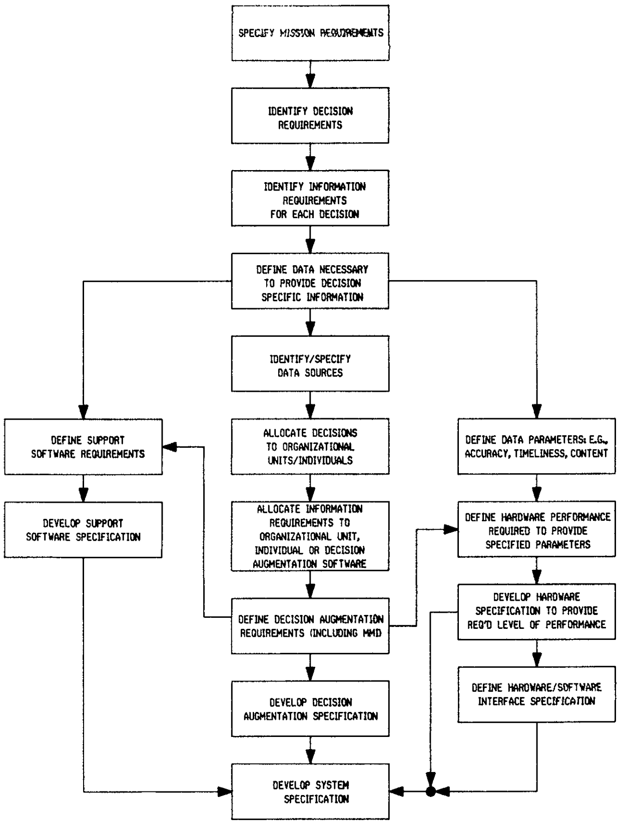

SPECIFY MISSION REQUIREMENTS
IDENTIFY DECISION REQUIREMENTS
IDENTIFY INFORMATION REQUIREMENTS FOR EACH DECISION
DEFINE DATA NECESSARY TO PROVIDE DECISION SPECIFIC INFORMATION
IDENTIFY/SPECIFY DATA SOURCES
DEFINE SUPPORT SOFTWARE REQUIREMENTS
ALLOCATE DECISIONS TO ORGANIZATIONAL UNITS/INDIVIDUALS
DEFINE DATA PARAMETERS: E.G., ACCURACY, TIMELINESS, CONTENT
DEVELOP SUPPORT SOFTWARE SPECIFICATION
ALLOCATE INFORMATION REQUIREMENTS TO ORGANIZATIONAL UNIT, INDIVIDUAL OR DECISION AUGMENTATION SOFTWARE
DEFINE HARDWARE PERFORMANCE REQUIRED TO PROVIDE SPECIFIED PARAMETERS
DEFINE DECISION AUGMENTATION REQUIREMENTS (INCLUDING HMI)
DEVELOP HARDWARE SPECIFICATION TO PROVIDE REQ'D LEVEL OF PERFORMANCE
DEVELOP DECISION AUGMENTATION SPECIFICATION
DEFINE HARDWARE/SOFTWARE INTERFACE SPECIFICATION
DEVELOP SYSTEM SPECIFICATION

FIGURE 4. DECISION-ORIENTED SYSTEM DESIGN

The design approach shows hardware requirements, the right-hand path, also being dependent on the "Define Data" task. First, the data parameters are defined. Hardware performance requirements are stipulated based on the hardware's capability to provide the data parameters (or performance) specified. This, then, determines the hardware specification, i.e., the ability to provide the data needed to produce information required for decision making. The hardware elements include sensors, which must furnish specified data at a certain rate and accuracy to allow a meaningful and timely decision to be made; computers which must have the requisite capacity and computational speed; and communications systems which must have the bandwidth to handle data transmission load. Developing hardware requirements based on the decision maker's needs will provide a better fusion between these system components, resulting in a higher level of performance than previously achieved by using the current system design approach.

After the hardware, support software and decision augmentation specifications have been defined, the hardware/software interface specification can be defined. By combining these specifications appropriately, the overall system specification can be developed. What will result is a C^2 system (or weapon system) that has a greater likelihood of meeting its theoretical performance level.

ORGANIZATION ANALYSIS

Reorientation of the system design approach to emphasize the human and his decision making responsibilities requires analyses that are not considered in the current approach. These analyses are qualitative and concentrate on factors that contribute to the framework of decision making in an organizational environment. The analysis methods and data sources for the initial requirements analysis are shown in Figure 5. Mission requirements are determined by a functional analysis of operational problems and deficiencies, including tactics, sensor utilization, sensor performance, available resources, and enemy order of battle. Decision requirements are identified by going directly to the decision makers. Non-quantitative behavioral/social science methods such as questionnaires, interviews, verbal protocols and observation of the decision-maker in his operational environment should be employed. Analysis of the operational and organizational environment is also crucial. It should include identification of the following elements:

- informal as well as formal communication links
- apportionment of decision-making responsibility to components of an organization
- relationships between organizations
- what decision aids (automated or not) are currently being used or could be used if available.

Analysis of these and other organizational elements will enhance the ability to correctly define decision requirements and to determine the appropriate level of decision augmentation.

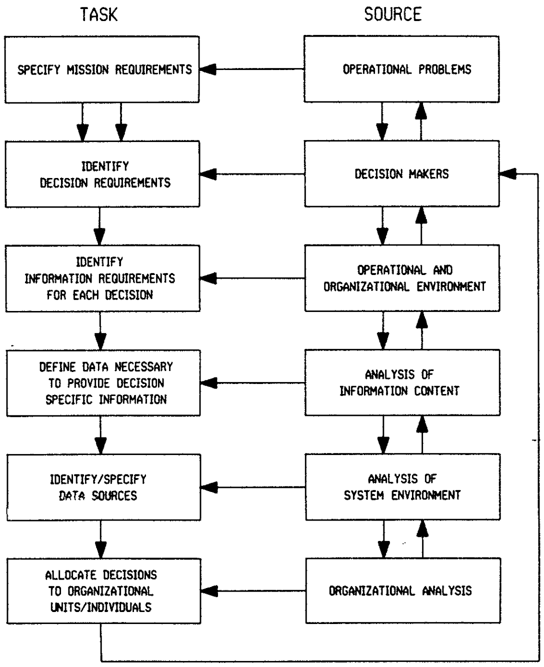

TASK
SOURCE
SPECIFY MISSION REQUIREMENTS
OPERATIONAL PROBLEMS
IDENTIFY DECISION REQUIREMENTS
DECISION MAKERS
IDENTIFY INFORMATION REQUIREMENTS FOR EACH DECISION
OPERATIONAL AND ORGANIZATIONAL ENVIRONMENT
DEFINE DATA NECESSARY TO PROVIDE DECISION SPECIFIC INFORMATION
ANALYSIS OF INFORMATION CONTENT
IDENTIFY/SPECIFY DATA SOURCES
ANALYSIS OF SYSTEM ENVIRONMENT
ALLOCATE DECISIONS TO ORGANIZATIONAL UNITS/INDIVIDUALS
ORGANIZATIONAL ANALYSIS

FIGURE 5. ANALYSIS METHODOLOGY

TECHNOLOGY IMPLICATIONS

Emphasis on decision making functions has a number of implications for the design and implementation of C^2 system (and weapon systems). Five are discussed below.

First, a basic understanding of human cognitive processing is required to realize fully decision augmentation potential. C^2 systems should be designed to provide assistance in those areas where human capability is limited while still capitalizing on human strengths. For example, humans have attention and memory limitations, inherent heuristics which they employ in information processing and limited ability to process numerical data in complex, stressful and data overload situations. These are areas in which decision augmentation can provide significant improvements over unaided decision making.

Second, HCI design can affect decision behavior; therefore, its effects should be considered in system design. Decision augmentation system design is often concerned with methodological validity, without sufficient attention to the relationship between it and the user (e.g., the amount and kind of user-augmentation interaction, dialogue style, information presentation formats, Schwartz and Jamar, 1983). If results of decision augmentation are not presented in a directly useable format with due consideration to the user's needs and desires, they will not be effectively used. Importance of the interface between human and computer has received increased attention in recent years and is particularly critical in situations in which decisions must be made under time pressure or conditions of high data volume. Some examples of good interface design features are:

- use of graphics to represent situational overviews, particularly geographic representations
- provide only that information needed to support a decision situation
- display historical data on request
- provide embedded training and on-line tutorials to facilitate use by both novices and experts
- provide easy means of user-computer communication
- make the knowledge base and decision rules in decision augmentation systems accessible so the user can query them
- insure computer response speeds are commensurate with user expectations
- provide automatic mode settings that users can override (e.g., number of alternatives displayed, what utility measure to order alternatives on)
- provide suggestive rather than authoritative output
- provide succinct rather than conversational output.

Third, improved understanding of and attention to innovation acceptance is needed. Whether a decision system is used or not will depend not only on its design but also how it is introduced (Mackie and Wylie, 1985). The system must be designed so that it meets the user's needs rather than introducing additional workload. Even if the system is actively involved in the decision-making process, and in some cases excludes the human, it should not *appear* to erode individual control or decision making authority. System operation should require only minimum knowledge of computers and should not involve complex operating procedures. The user community should be involved throughout the development process to facilitate the introduction of new decision automation systems (Adelman, 1982). They should be involved early in the development cycle, when system requirements analysis is being conducted. Also, prior to system introduction, potential users should be briefed on system capabilities and operating procedures so they know what to expect and can take full advantage of what is provided.

Fourth, AI technology is rapidly becoming accepted as a major tool for implementing decision augmentation systems. It allows the use of more complex decision rules in addition to numerical computations and algorithms. Furthermore, the knowledge of experts can be acquired and encoded into the system knowledge base.

Finally, decision augmentation systems should be able to adapt to both user and environment. In order to make the system adaptable, the system architecture in which the knowledge base and decision rules reside must permit change. It should be possible to update when new tactical situations arise or other environmental changes occur. Also, there are individual differences among decision makers both in decision making style and the importance they place on different criteria in determining a final decision. This individuality should be accommodated to the extent possible using innovative architecture until technology has evolved to the point where learning can be an integral part of the decision augmentation system. The current system design approach does not bring into focus the technology implications discussed. It is only through a decision oriented design approach that full advantage can be made of the new technology.

SUMMARY

As a consequence of the increased data volume in current and future C^2 systems and the limitations in human cognitive processing capacity, a reorientation of the system design process has been proposed. In the approach proposed, system design is based on decision requirements rather than on hardware performance. Also, decision augmentation techniques and innovative information presentation are needed to reduce the data overload. The higher degree of integration and automation possible with decision augmentation systems coupled with the emphasis on decision functions should greatly reduce the incoming data load so that the decision maker can devote more time to thinking about operational problems rather than merely reacting to the task environment. If these changes can be accomplished, C^2 system capabilities should improve and, as a consequence, mission effectiveness should also improve.

REFERENCES

Adelman, L. Involving users in the development of decision-analytic aids: the principle factor in successful implementation. *Journal of the Operational Research Society*, 1982, *33*, 333-342.

Keen, P. G. W. & Scott-Morton, M.S. *Decision Support Systems: An Organizational Perspective,* Reading, MA: Addison Wesley, 1978

Mackie, R. R. & Wylie C. D. *Technology Transfer and Artificial Intelligence: User Considerations in the Acceptance and Use of AI Decision Aids.* Goleta, CA: Human Factors Research Division Essex Corporation, Technical Report TR 51231-1, November 1985.

Rouse, W. B. Design and evaluation of computer-based decision support systems. In S.J. Andriole (Ed.), *Microcomputer Decision Support Systems.* Wellesley, MA: QED Information Sciences, 1984.

Schwartz, J. P. & Jamar, P. Lack of guidance for decision aid interface design. *Association for Computing Machinery SIGCHI Bulletin,* 1983, *15,* 13-17.

Wohl, J. G. Force management decision requirements for Air Force Tactical Command and Control. *IEEE Transactions on Systems, Man, and Cybernetics,* 1981, SMC-11, 618-639.

Zachary, W. A cognitive based functional taxonomy of decision support techniques. *Human-Computer Interaction,* 1986, *2*, 25-63.

NEW CONCEPTS IN THE BRL ADDCOMPE FIRE SUPPORT APPLICATION

SAMUEL C. CHAMBERLAIN

US Army Ballistic Research Laboratory
Aberdeen Proving Grounds, MD 21005-5066

OVERVIEW

The Ballistic Research Laboratory's (BRL) System Engineering and Concepts Analysis Division (SECAD) is developing a fire support application for the Army/DARPA Distributed Communications and Processing Experiment (ADDCOMPE). This application addresses the unique, very dynamic data distribution challenges associated with brigade-level fighting forces (and below) and considers several of the concepts detailed in the *Advanced Field Artillery Tactical Data System (AFATDS) Organizational and Operation Plan.*[1] Controlled laboratory experimentation will be conducted to evaluate several new data distribution concepts described below. The fire support application will include portions of two (of the five) key AFATDS fire support functions, *fire support control & coordination* (FSCC) and *field artillery tactical operations* (FA TAC OPS), and will be implemented on Sun Microsystems workstations for five key field artillery nodes: a maneuver brigade fire support element, a direct support field artillery battalion operations element, and three maneuver battalion fire support elements, see **Figures 1 & 2**. These five players will use this application to respond to several tactical vignettes that could occur as a result of a critical situation (e.g., a surprise attack) that would force them to change plans on-the-run, reconfigure forces and the command chain, conduct an operation in a silent mode, and dynamically respond to battlefield losses.

A major goal is to design a fire support control system that can support "fighting level" commanders and soldiers who must contend with a highly dynamic, unpredictable, and hostile tactical environment. In this application several new concepts will be explored in an effort to study various techniques to provide more ***flexibility*** and ***survivability*** in these systems. As will be described shortly, flexibility is enhanced via a free-form, distributed knowledge base and its associated fire support control capability profile. Survivability is enhanced through minimizing radio transmission emanations by taking advantage of "overheard" information, providing a "radio silence" (emission control, or EMCON) mode of operation, and using multicast transmissions when possible. Also included is a knowledge exchange protocol to facilitate the exchange of information using these features, see **Figure 3**.

[1] Draft Revised O & O Plan for AFATDS, US Army Field Artillery School, 23 June 86.
NOTE: AFATDS is the follow-on to the US Army's Tactical Fire Direction System, TACFIRE.

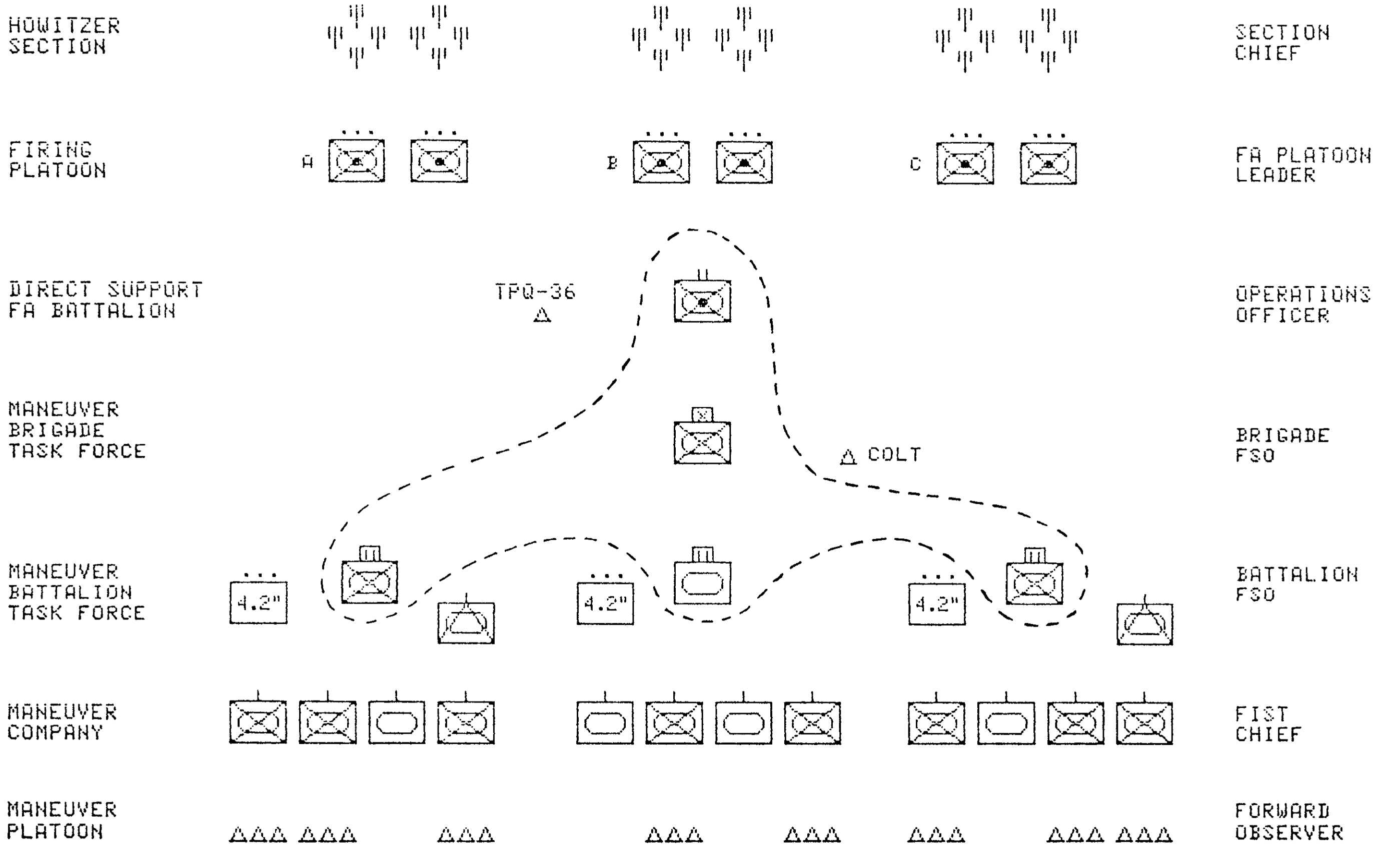


Figure 1. Organizational View of BRL Fire Support Application Nodes (Nodes Located Within Boundary)

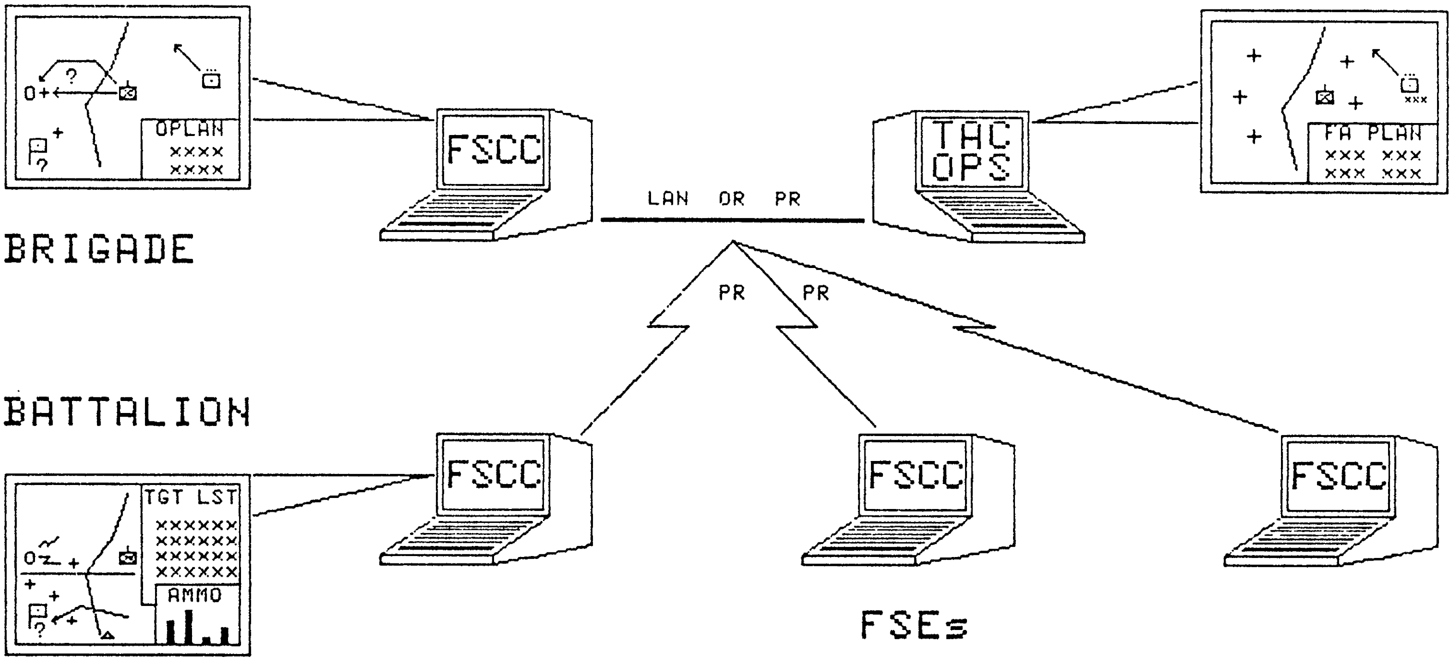


Figure 2. Functional View of BRL Fire Support Application Nodes

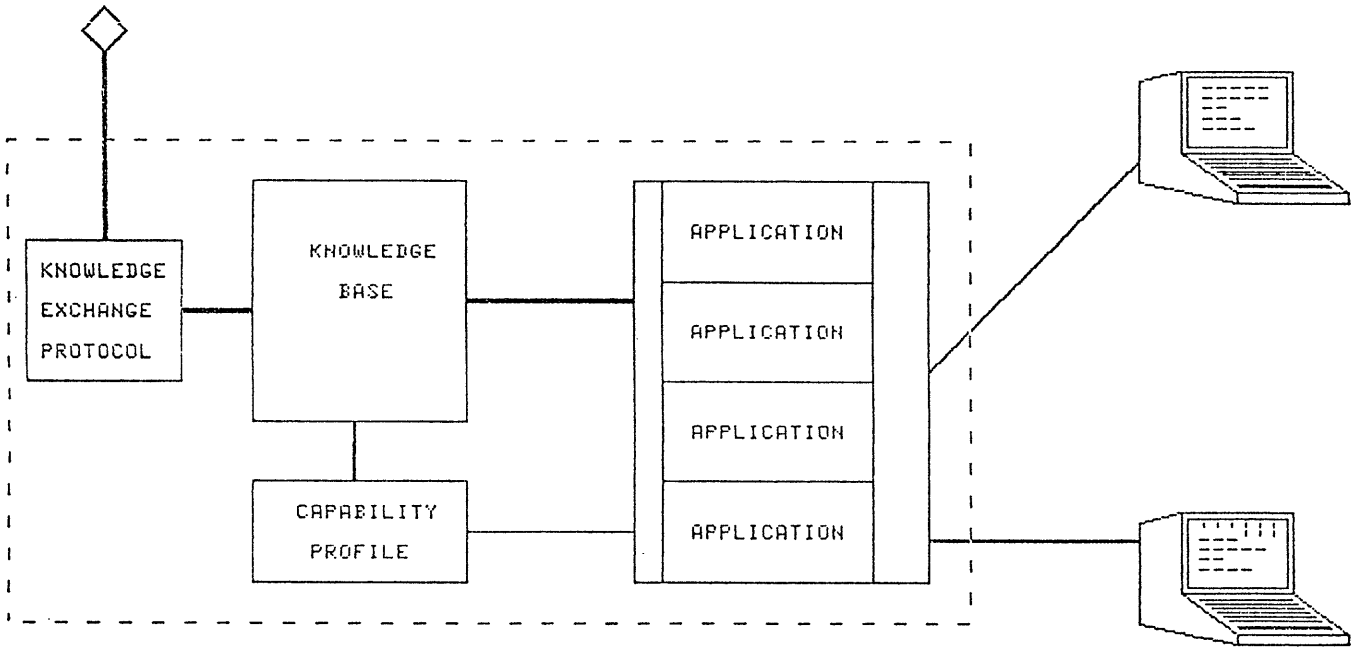


Figure 3. Block Diagram of Basic Application Software Modules

BASIC TENETS AND OBSERVATIONS

Through past experiences with both fielded and prototype tactical automatic data processing (ADP) systems, some basic tenets and observations have evolved concerning these digital systems.

1. At the lower, "fighting" levels (brigade and below) concurrent and consistent databases are not practical nor even required.

   The underlying reason for this is the very limited bandwidth offered by single channel, VHF-FM radios (Very High Frequency, Frequency Modulated) that will remain the primary form of communications for a long time to come. This fact, combined with the rapid pace at which pertinent battle information changes (e.g., unit locations, ammunition status, casualties, etc.) makes the constant updating of such information impractical. But in addition, the numerous small updates required in current fighting level systems are usually of little worth to decision makers (e.g., minute changes in status, such as the number of rounds just fired, or a small location change); simply put, they are normally too expensive in bandwidth for their worth. In other words, the requirements for *flexibility* and *responsiveness* often outweigh the requirement for accuracy. It is simply not necessary to communicate small modifications; it is more important that initial, new information be rapidly processed and distributed. (Note: this refers to information used for tactical decision making and planning, not that for directing weapons. In contrast to naval warfare, the close-in battle of ground warfare depends on human rather than electronic sensors to direct, fire, and control weapons.)

2. Several effective voice communication techniques have been lost in the transition to digital communications:

   A. Monitoring ("overhearing") - Decision makers have passively listened to voice radio communications to track battle situations for years. Suddenly, with the strict addressing schemes used in digital systems, perfectly good messages (and potentially useful information) are thrown away simply because they were not addressed to the individual host (computer).

   B. Listening Silence (EMCON Mode) - This has also been a common practice in voice communications. This requires a protocol that supports the reception of information without requiring that an acknowledgement be immediately transmitted. Current network protocols require an immediate acknowledgement.

   C. Multicast - It is very common in voice communications to expect several recipients to hear a single message (e.g., firing commands to all the guns in a howitzer battery). Multicast is now an active research area in computer networking.

3. Computers are communicating with computers, not computer terminals.

   Computer technology has shrunk components both physically and in price; consequently, for the old price of a computer terminal one can now purchase a computer system. Mainframe computers with remote terminals are rapidly disappearing since each battlefield node can have its own computer. Therefore, computers are communicating with each other (rather than with terminals), and more sophisticated protocols should be developed to exploit this fact.

4. The electronic signature that results from numerous digital transmissions is huge and can be significantly decreased using more efficient protocols.

5. Processing power is growing rapidly, bandwidth is not.

   Based upon recent history, the amount of computing power available will continue to be more than anticipated. Therefore, system developers should focus their attention inward and design systems that are *processing intensive* rather than *communications intensive.* For this BRL application, it has been assumed that any processing power required will be available, and thus, this is not considered an obstacle (as it could be if fielding were imminent).

6. Getting *at* the data is usually harder than getting the .data.

   The tough problem is making sense out of even a portion of what is available. It is imperative that *imaginative data retrieval* be supported so that experts can use their ingenuity to spontaneously query the available information using complex combinations of criteria.

7. Tactics is software.

   To support flexibility, information concerning tactics (and doctrine) should be stored in a format familiar to the user and be accessible so that modifications can easily be made to support the rapidly changing and often unanticipated events that occur in battle. Tactics information should be explicitly represented, that is, stored separately from the other programs and data within a computer system. (This is the basic concept behind the "knowledge base" found in expert systems.)

8. Finally, partition tasks to let people do the things people do well, and let computers do the things computers do well.

   Although this may be an obvious (and even trite) statement, it is easily forgotten. When presented with tough command and control problems, this simple sounding heuristic can often provide the impetus for a satisfactory solution: let the person do part of the task! The design of a *balanced* system that effectively partitions tasks between human and computer resources is a tough problem, especially since this partition is dynamic. A healthy perspective is to attempt to maximize the power of human decision capabilities through the exploitation of computer technology, not the reverse.

NEW CONCEPTS

Considering the above tenets, six basic concepts are being explored in the BRL fire Support Application for ADDCOMPE.

- Distributed Knowledge Base (DKB)
- Fire Support Control Capability Profiles (CAPs)
- Overhearing Capability
- Listening Silence
- Multicast Datagrams
- Knowledge Exchange Protocol

The foundation of the application is a fact-oriented, free-form, **distributed "knowledge base"** that is simply a collection of many interconnected *facts*. (This is not to be confused with the traditional term "knowledge base" used in expert systems.) Each fact is an indivisible piece of information about a particular item, activity, or event and is identified by a unique fact identification number (fact ID). Its structure is defined by the user and is free-form (like any abstract data type) in that it can be composed of integers, real numbers, strings, enumerated types, or references to other fact IDs. Facts also include a quality indicator because a key tenet of this application is that it is too expensive in communications bandwidth, as well as unnecessary, to provide replicated databases at the fighting level; therefore, some measure of information quality must be provided (e.g., overheard information could be assigned a lower data quality than directly addressed information). There is one DKB per host and all application programs communicate with each other by entering and retrieving information from their local DKB. This information is exchanged as facts (chunks of knowledge) whether between a DKB and an application program or between DKBs. The common "message" paradigm is discarded for more efficient facts in order to to minimize the effect on the already limited bandwidth.

Two particularly powerful features of the DKB are its query language and triggers. The *query language* appears similar to the C programming language and efficiently allows complex (imaginative) criteria to be defined (e.g., "show me all armor targets east of x with speeds greater than y and heading between a and b at time greater than t"). The DKB will then return the fact IDs of all facts that satisfy the criteria. Advanced techniques to propagate queries across a network of DKBs are being explored that will eventually take into account bandwidth availability and the potential worth of the information. A *Trigger*, as the name implies, notifies other programs when its query-like criteria are matched. Each time a new fact is entered into a DKB a list of triggers is checked for a match; when this occurs, the program that entered the trigger is notified and is presented with the fact ID of the triggering fact. It is expected that these two features, the powerful query language and triggers, will significantly enhance the users ability to cleverly retrieve information from the DKB.

The **fire support control capability profiles**, or "CAPs", contain explicit fire support control information that describe the "personality" and capabilities of a particular node. This includes the *data dictionary* that defines the form of the free-form knowledge base; the knowledge *distribution information* that dictates what, when, and where knowledge is sent, replicated, or retrieved; and the *event triggers* that alert other application programs when certain combinations of facts are entered into the DKB. The CAPs can be modified dynamically (even from remote locations) thus allowing all the items just mentioned to be maintained in a form most suitable for a particular situation; of course, this also has serious security ramifications that are also being considered. It is envisioned that the CAPs, which basically implement standard operating procedures (SOPs), would be developed by doctrine and tactics experts (e.g., US Army Training and Doctrine Command schools) and perhaps modified by specialists in the upper echelons of a particular unit (e.g., corps, division, or brigade); they would not normally be accessible to the soldier in the field.

Although CAP distribution rules can be very specific (e.g., send all unit location reports to some unit), they will normally be very general. *Thresholds* are used to determine when information is worth reporting. Thresholds can be defined in a manner applicable to the data; for example, status information may not be transmitted until it changes by more than ten percent from the last update, while enemy sensing may need to be a certain type and size to warrant reporting. How *much* information gets reported may also be defined in general terms. It could be decided that each unit should only keep information about two echelons below. To implement this policy, each unit would report status information to its parent about the units one level below. For example, battalions would send status information about their subordinate companies to their parent brigade; therefore, the brigade DKB would be limited to information concerning the companies subordinate to it. If information about platoons were desired by the brigade node, then a conscious decision would have to be made to use more of the bandwidth allocated by the data distribution policy currently in effect. When combined, these two simple features will allow decision makers to control when and what information gets distributed thus providing them the power to control the nature and volume of the digital traffic emanating from their subordinate units.

In the past, commanders and their staffs have kept themselves informed by simply listening to the voice transmissions occurring on several radio nets. Similarly, the collection of "free" digital data (at no cost in bandwidth) is another feature of this system. Through the use of event triggers all "**overheard**" information is screened for possible inclusion into the local DKB. This allows pertinent information (as defined by the event triggers active in a DKB) to be collected locally without any additional burden on the data communications system. Having this information locally might prevent it from being requested from a remote DKB at a later time (thus saving bandwidth). Although much of the overheard information might not have been requested because its need was not predicted, in reality it could be important in an unforeseen context (e.g., "real estate" allocation conflicts during a planning phase). It is anticipated that overhearing will be a very advantageous feature.

Listening silence has also been a common practice in voice communications. This application will allow *others* to send information to a node that has entered into *EMCON mode* even though acknowledgements are not being returned. Upon leaving EMCON mode, bulk acknowledgements are transferred using an appropriate media (e.g., radio, motorcycle transported floppy disk, etc.). The major task in this feature is maintaining a list at the transmitting node of what needs to be acknowledged when the receiving node leaves EMCON mode. Only the latest updates need be acknowledged; older facts can be retracted from this list when a newer update is entered. Many of the implementation details of this feature are handled by the Knowledge Exchange Protocol (described below).

Multicast is the ability to send the same message to several recipients with only one transmission. This capability is commonly used in military voice communications. Because it reduces bandwidth requirements, it is of specific interest to this application, especially in relation to the EMCON and network overhearing schemes. Since this is currently an active area of networking research, the technique used in this application will most likely be one of those being studied at various institutions.

Interactions between individual knowledge bases is accomplished by the **Knowledge Exchange Protocol** (KEP). The KEP is a datagram-oriented protocol built on top of the standard DoD IP and UDP protocols. Among other things, it provides facilities for selective and bulk acknowledgements (to support EMCON mode). Several other subtle network protocol features are also required for this application. For example, the DoD IP protocol may arbitrarily fragment messages. This presents a problems when overhearing information since partial facts are usually meaningless. The KEP interacts with the DKB and CAPs to insure that information is transmitted in logical chunks (facts), which if overheard, are meaningful to the receiver. When combined, these features (the free-form, distributed knowledge base, fire support control capability profiles, network monitoring, EMCON mode, multicast datagrams, and the knowledge exchange protocol) produce many significant, and often subtle, interactions whose synergistic effects will be a major focal point of these research concepts.

APPLICATION PROGRAMS

The preparation, maintenance, and dissemination of the information associated with a fire support plan for a maneuver operations order (OPORD) is an excellent vehicle to demonstrate the above concepts in a dynamic, real-time environment. The application software will exploit the aforementioned features to assist the soldier by: identifying critical information and alerting the operator, extracting current situation information from the DKB, displaying the unit's mission, providing tools to develop or modify the fire support plan information and update the prescribed CAPs, insuring that the appropriate information is in the knowledge base, controlling the dissemination of the fire support plan information, and making maximum use of graphics application "tools" and other software available; see **Figure 4**.

There will be two main application programs that reside above the DKB. The first is named *WORK MAP* and it will provide tactical map and overlays information. Standard 1:50,000, 1:100,000, and 1:250,000 scale military maps have been digitized and are presented on the Sun workstation monochrome display. Graphical information, such as unit locations, can be displayed on top of the map as a result of queries or fact entries to the DKB. A DKB read-only version, named SIT MON, may also be provided to serve as a situation monitor. The second application program is named *ORG FACTS* and is a graphics program to display organization charts and status information. This will include both generic and actual unit information for both friendly and enemy units. This application also provides a synopsis of personnel and equipment below each of the bottom units (the leaves of the tree structure) displayed in the organization structure. The user can use the mouse to expand and contract the tree, to build the task organization by attaching and detaching units, and to identify specific properties of the units displayed. Other applications include: *FIRE PLAN* that will use the other two programs along with its own display to assist the user in building the unit's fire support plan and *MOD CAPS* that will contain a capability profile editor to allow CAPs to be modified and also allow alternative CAPs to be activated. Access to MOD CAPS will be a restricted to selected, specially trained personnel.

CONCLUSION

Although a significant amount of work is being expended in developing these application programs, it must be emphasized that their purpose is to demonstrate the underlying concepts being explored. Since one cannot see or experience the underlying software, this fact is easily overlooked. It is not *what* is displayed that is important, but *how* the information that is displayed is obtained and distributed that is unique.

The overall goal is to build an experimental system that is responsive enough to the user that it will still be preferred during degraded modes of operation. Hopefully, the combination of a distributed knowledge base, capability profiles, overhearing, listening silence, and a new knowledge exchange protocol will provide such an environment. If not, the equipment will be thrown aside, and commanders and their staffs will continue to huddle in a circle and make figure drawings in the dirt during a crisis situation.

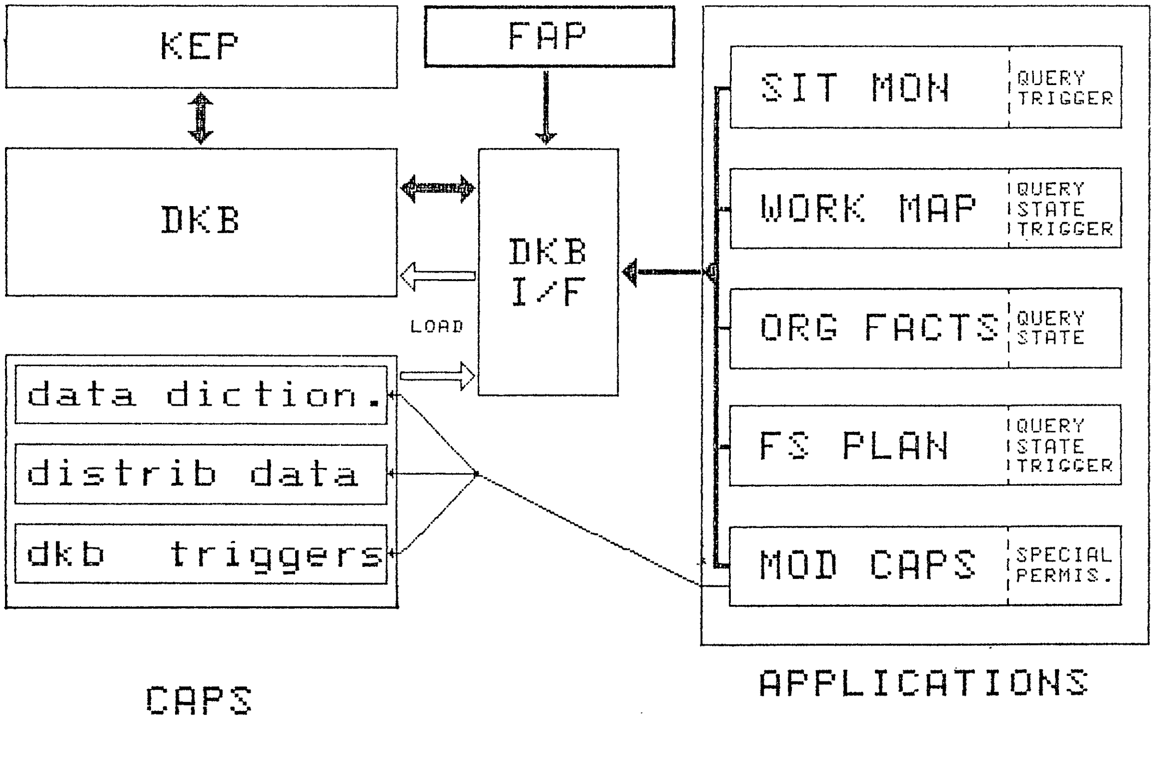


Figure 4. Block Diagram of Application Programs

A Prototype Expert Assistant for Tactical Intelligence Battlefield Situation Assessment

William H. King, Patricia A. Tao and Michael J. Becker

Hughes Aircraft Company
Ground Systems Group
P. O. Box 3310 MS 618/M306
Fullerton CA 92634

Barbara S. Larsen

Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Drive
Pasadena CA 91109

ABSTRACT

Given a database of correlated and fused intelligence, the job of the situation analyst is to determine current enemy activities and predicted enemy intent, and to report this assessment to a higher level commander for the purposes of planning and decision-making. In the tactical battlefield environment, this assessment must be accomplished in real- or near-real-time. The proliferation of sensors (collectors) and resulting volumes of data, combined with the increasing complexity (speed; mobility, sophistication) of the threat, renders automated decision support mandatory for the situation analyst. Because situation assessment is a complex, ill-structured task that relies heavily on the judgement, perception and experience level of the analyst, traditional algorithmic approaches have failed due to their requirement for well-defined problem structure. A promising solution lies in the expert system, which relies on a large body of knowledge about the field of expertise, rather than on pre-specified algorithms. This paper describes a prototype expert system for tactical battlefield situation assessment. This rule-based system employs a data-driven, exhaustive breadth-first search to consider all possible clues of potential enemy activity in forming the situation assessment. The system utilizes reports on both air- and ground-based activities, with attention directed at enemy division and regiment levels. Key features of the system are 1) an architecture which effectively models the problem solving strategies of expert analysts; 2) an integrated text and graphics explanation capability; and 3) successful adaptation of the Dempster-Shafer technique for evidential reasoning.

INTRODUCTION

Background

For a commander to make tactical decisions responsive to rapidly changing events on the battlefield, he must have current and accurate information tailored to his needs. He requires an overall description of the operational situation to include indications and warnings analyses, threat assessments, target acquisition judgements, and inferences of enemy activities and intentions, and relies heavily on intelligence to accomplish this.

In battlefield situations, the timeliness and accuracy of the intelligence product is critical. However, technical advances in military operations have made these goals more difficult to attain. As the speed, mobility and sophistication of the enemy have increased, shortened decision timelines have emphasized the need for real time and near-real time intelligence. At the same time, intelligence collection systems have proliferated over the past several years, increasing the variety, complexity and volume of message traffic that the analyst must digest. Indeed, one of the most critical questions facing defense planners today is whether commanders and their staffs will be able to manage the deluge of data which is expected during a crisis or conflict. This constantly growing volume of time-critical information threatens to push workloads beyond human capacity.

Further compounding the problem is the fact that intelligence analysis has historically been a primarily manual process, depending heavily on the technical and operational knowledge of the personnel. But in the more intense electronic battlefield of the near future, the intelligence analyst requires automated support in order to provide a timely and accurate assessment of the operational situation to the battlefield commander. High volume, on-line message processing can substitute for paper input, and color displays can replace grease pencil on acetate map overlays. Automated correlation and data fusion can reduce the sensitivity of the tactical database to the speed and experience of the watch personnel. Finally, automated support for the more interpretive task of **situation assessment** will improve both the quality and quantity of the product which the analyst is tasked to provide.

Domain Objectives of Situation Assessment

In a greatly simplified description of the SIGINT analysis process, initial reports (e. g., TACREPS, TACELINTS) of current enemy activity are generated at several sites by single source sensor analysts using raw intelligence sensor input. These reports may be either manually or automatically generated, and include amplifying comments supplied by the reporting analyst. These are transmitted to an all-source (multi-source) analysis center at which data correlation and fusion are accomplished. The resultant correlated database is interpreted by a situation analyst, whose primary objectives are to provide his commander with both a timely and reliable portrayal (snapshot) of current enemy activities as well as insight into likely future enemy actions.

In manually developing the battlefield picture, the analyst relies on two sources of input. First, static information on enemy doctrine and tactics is available either through published reference materials or through his own experience. The other source of input is the incoming message traffic which supplies the dynamics of the evolving situations. As the analyst processes the incoming messages, a significant battlefield indicator may be detected based on message content, track history, unit location (absolute, or relative to other enemy units, friendly units, or key terrain features), or as a result of the correlation and fusion process. The analyst acts on the determination of this indicator to determine its impact (alone or in combination with similarly-derived indicators) on the current and projected battlefield picture. When such an assessment is made, the analyst will draft the appropriate report and transmit it to his commander and/or other interested activities. As subsequent reports arrive, the analyst must recognize the most critical indicators and prioritize his activities appropriately.

The primary activities of the situation analyst in accomplishing the above objectives are shown in Figure 1, and form the model for the prototype expert subsystem for situation assessment support described in the following sections.

SYSTEM ARCHITECTURE

Driving the system architecture definition and the design of the key system components (knowledge base, inference engine, user-interface) is the model of the domain task (Figure 1). This set of tasks maps onto a hierarchy of five primary information levels (key data types or objects): messages, tactical units, battlefield indicators, current combat situations, and predicted enemy intent.

As intelligence reports (messages) enter the system, they are "correlated" with existing tactical units currently in the database or are used to initiate new unit records. (This correlation is done using a unique unit identifier supplied in each simulated message. Although no claims are made of a correlation capability, this function must be modeled to provide the knowledge base dynamics.)

The next key step in the process is the inference of intermediate clues (indicators) of current battlefield activities. The rule-based paradigm was chosen for this step. Based upon key fields (unit type, unit identification, parent unit identification, location, speed, activity) within the updated/created enemy unit record, relevant rules are selected as potentially applicable to be invoked. Incorporated in each rule are antecedents which consider environmental data (weather, terrain), technical and doctrinal data (force structures, unit attack radii), as well as locational information on related enemy units and friendly units potentially under attack. Distance filtering and other functions necessary in establishing the rules test conditions are included as LISP functions and procedural attachments (daemons). A rule for inferring a RED-SUPERIOR indicator is shown in Figure 2. It is a typical rule, requiring the conjunction of several antecedents to be true before the conclusion is drawn. The successful satisfaction of all rule antecedents causes the rule to fire and generates an instance of an indicator which is associated with the enemy unit in question. The battlefield indicators considered in the prototype are shown in Table 1.

Since one message can trigger the instantiation of several battlefield indicators, the system must synthesize this collection of indicators into a higher level interpretation of the combat situation. As explained in the section on "Possibilistic Inference," below, the Dempster-Shafer technique is employed -- implemented as a table -- at this step, treating the indicators as evidence and the combat situations as the competing hypotheses. The set of possible combat situations available to the system are given in Table 1.

Certain critical situations (Assembly, March and Attack) are subject to more stringent criteria prior to being presented to the operator and added to the database. These spatial templates are based on well-known doctrinal rules, and include units which are doctrinally-related to the enemy unit being reported on. Successful matching of these templates results in a higher confidence being assigned to the potential situation.

The above processes result in a snapshot of the current battlefield picture, which is portrayed graphically to the analyst as well as textually in the form of situation alerts. A "predicted situation knowledge base" is then used to predict potential future enemy activity, based on current and past activities. Not only are the

RULE: GROUND-ACTIVITY AND RED-SUPERIOR

IF	A GROUND-ACTIVITY-INDICATOR HAS BEEN INFERRED FOR A CRITICAL ENEMY UNIT
AND	THE TIME OF THE LAST MESSAGE WITH LAT/LONG FOR THE CRITICAL UNIT IS LESS THAN 4 HOURS OLD
AND	THERE ARE 2 OR MORE FRIENDLY UNITS WITHIN PROXIMITY OF THE CRITICAL UNIT
AND	THE TANK, ARTILLERY, AND INFANTRY OPPOSING FORCES RATIOS ARE ALL GREATER THAN 1
THEN	INFER RED-SUPERIOR-INDICATOR FOR THE CRITICAL UNIT WITH CONFIDENCE CERTAIN

Figure 2. Rule for Inferring RED-SUPERIOR Indicator.

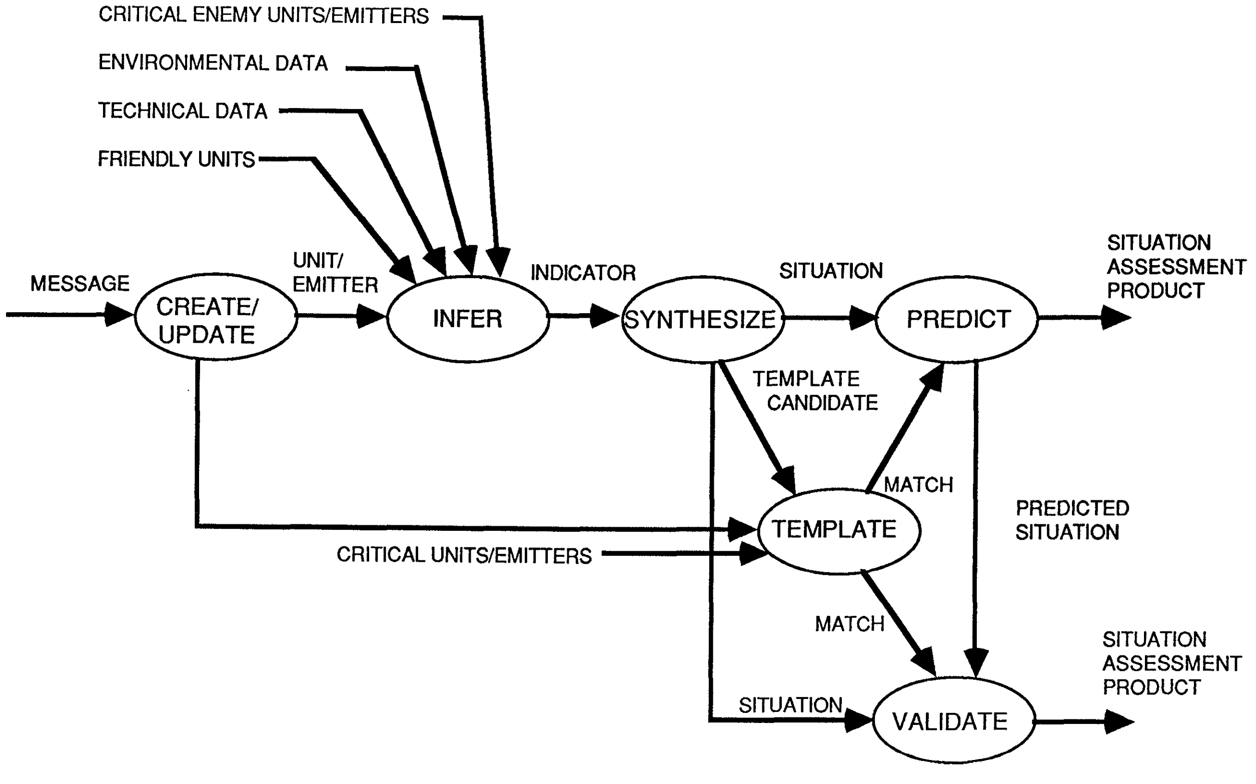


Figure 1. Top Level Model of Tactical Intelligence Situation Assessment.

BATTLEFIELD INDICATORS	COMBAT SITUATIONS
FWD-CMD-POST-FORMED	ATTACK-OF-DEFENDING-ENEMY
WATER-BARRIER	MARCH
HEAVY-VOLUME	ASSEMBLY
RED-STATIONARY	MOVE-OUT
AIR-DEFENSE-SUPPORT	STALLED-MARCH
LANDING-RADAR	MEETING-ENGAGEMENT
BOMBING	PURSUIT
AIRSTRIKE	RELIEF-OF-FORCES
CONVOY-REPORT	ASSAULT-CROSSING-WATER
RECON-REPORT	DEFENSE
NBC-REPORT	COUNTER-ATTACK
ATTACK-REPORT	DISENGAGE-AND-WITHDRAWAL
STALLED-REPORT	
ASSEMBLY-REPORT	
MOVEMENT-REPORT	
AIRSTRIKE-REPORT	
RED-SUPERIOR	
BLUE-SUPERIOR	
NEITHER-SUPERIOR	
NO-BLUE	

Table 1. Indicators and Situations.

current and future activities portrayed to the user, but they also serve as an adaptable filter which serves as a backdrop for subsequently inferred current situations, thus providing additional detection of a situation which appears abruptly and unexpectedly as a result on the inferencing process.

As the scenario develops, the analyst is immediately made aware of enemy activity and potentially developing situations. As discussed in the section entitled "Explanation Facility," the analyst can quickly and easily review the system's conclusions regarding battlefield activity by back-tracking through situations, indicators, data on tactical units, and messages to any level of detail he requires in order to substantiate the inferences that the system has drawn. This capability assists him in accomplishing his primary objective of providing situation assessment product (in the form of a written report) to his commander.

The paradigm after which knowledge has been organized is that of object-oriented programming. The knowledge base includes both observed objects (units and emitters), inferred objects (indicators, situations and predicted situations), and rules. These objects are represented as frames, allowing object specialization and inheritance of attributes. In addition, procedural attachments (methods and daemons) are included to model and respond to changes in spatial relationships among units.

The inference engine for the prototype system is a forward-chaining, breadth-first system which utilizes a relevant-rules index to increase rule-selection efficiency. This design was based on the model of the expert task. Situation assessment is largely data-driven (hence the forward chaining) and relies on the relationships between an exhaustive set of intermediate clues (hence breadth-first). The relevant-rules indexing scheme automatically generates pointers from generic frames of tactical units to that subset of rules which might potentially fire when such a frame is instantiated. In addition, meta-knowledge is incorporated within the control structure to focus the system's attention on the most appropriate tasks (e. g., inferring indicators, synthesizing indicators, applying templates, etc.).

EXPLANATION FACILITY

One of the key features of an expert system is its ability to elaborate on, justify and explain the conclusions that it has reached. This is especially important when the resulting command decision might be to maneuver forces or launch a weapon. In an advisory system such as the prototype developed under this effort, the analyst requires complete confidence in the system inferences. Thus, development of an integrated textual/graphics explanation capability has been a prime focal point of this effort.

In developing the prototype, three specific requirements were imposed on the explanation subsystem. First, critical information must be displayed to the analyst in such a way that critical decisions may be made without additional input from the user. This implies that the large amounts of data relating to system conclusions must be extracted, reduced, and presented to the analyst in concise and unambiguous terms, allowing him to interpret these conclusions in light of all relevant information. Thus, the supporting displays must be rich in content, rich in utility, and lacking redundant or extraneous data.

A second requirement is that of both graphical and textual explanation support displays. Because the underlying rules leading to most conclusions can be expressed in English terms, it is natural that a textual explanation capability be available. However, many of the rules incorporate spatial and temporal information (see Figure 2), which is best displayed graphically.

Finally, the level of explanation must be tailored to the particular real-time requirements of the analyst. Sources of these varying requirements include different levels of expertise or experience using the system. Even for a single analyst, requirements may change depending upon the intensity of his workload or on the level of fatigue he is experiencing.

The approach taken to meet the above requirements was to define an integrated graphic/textual explanation capability. In this approach, neither display is considered the primary explanation support tool, although the textual display is used as the primary interface to system-level processes. Developed on a Symbolics 3670 processor, the hardware supporting the display capabilities are a Symbolics monochrome display for textual information and a Mitsubishi color monitor for graphical and textual display. Although most of the user-interface software was developed specifically for this application, a graphics map data tool kit, GEOFLAVORS-2, was procured from Verac, Inc. This package uses World Data Bank II to provide geo-political boundaries and other features.

The user-interface subsystem includes multiple windows as shown in Figure 3. The windows serve to alert the analyst of high-confidence, high-priority enemy activity and to allow his investigation into the underlying rationale for such alerts. Driving the analyst's activity is the situation alert window, a part of the textual display. Based on a threshold set by the analyst, combat situations whose confidence exceed the threshold will be displayed. This display summarizes the key attributes of the inferred battlefield situation, thus providing the analyst with sufficient data with which to determine his next action. The situation alert includes the time, location and confidence

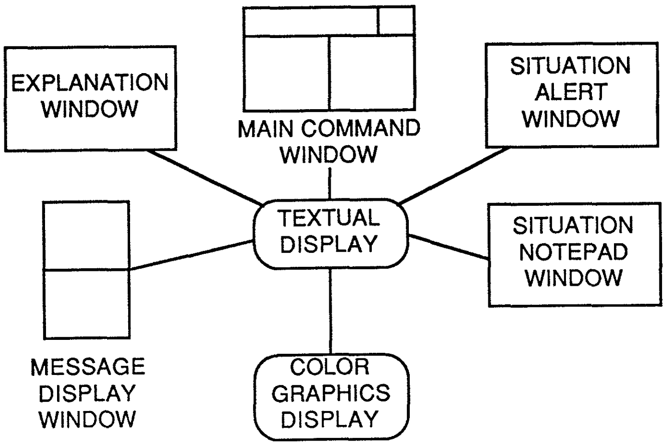


Figure 3. Independent Window System for User-Interface.

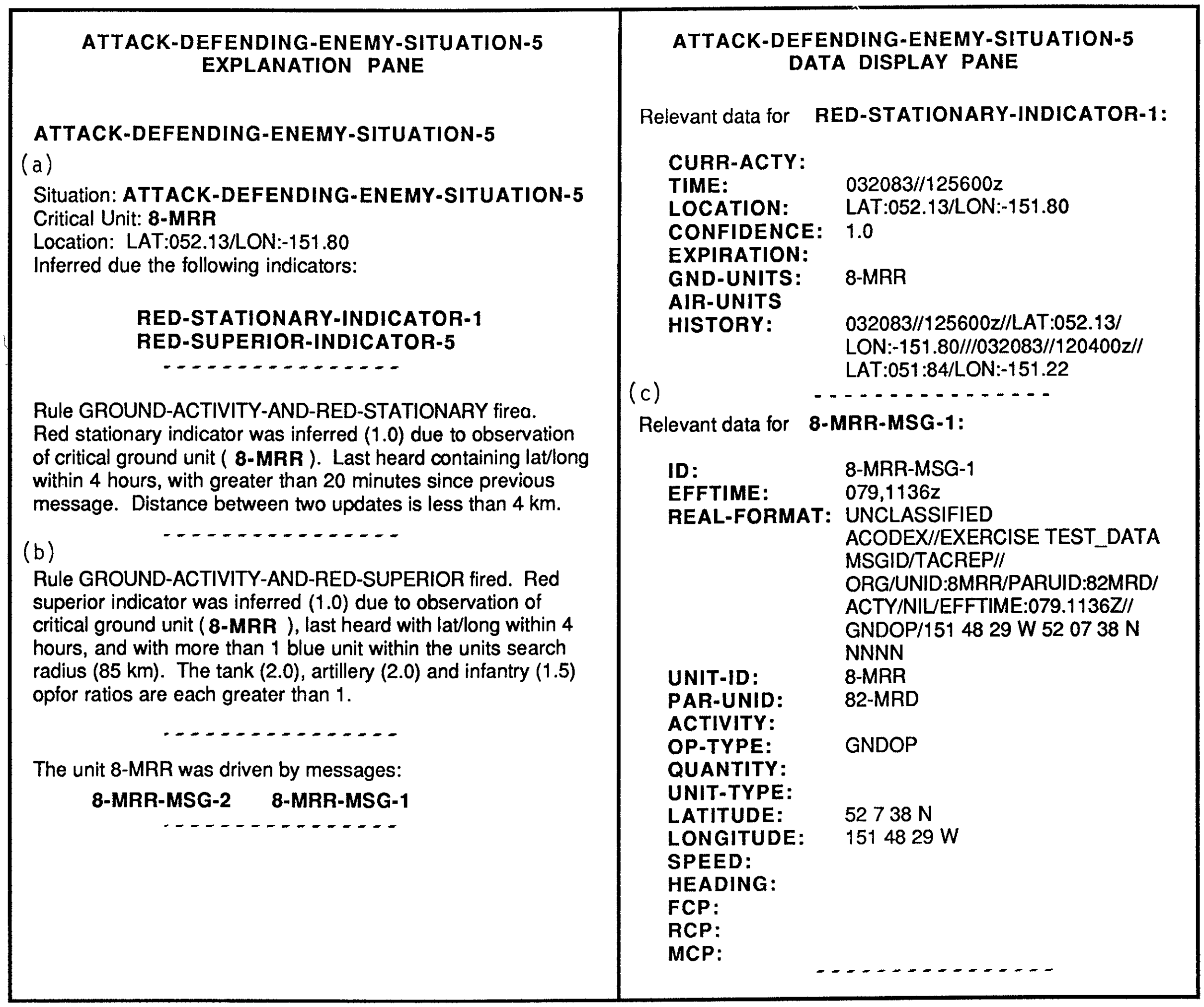
associated with the situation, as well as the identification of critical units driving the situation. Supporting this conclusion, the same information is highlighted on the graphic display, thus providing quick indication of the position of the hostile forces relative to own forces, which may be more valuable than the absolute position presented in the situation alert. This is particularly desirable when the inference process has used spatial or temporal relationships between battlefield entities.

Utilizing the extensive windowing capabilities of the host machine, a functionality has been implemented which allows the analyst to investigate in a top-down hierarchical manner the rationale underlying system-generated conclusions at any level of detail he requires. By activating an explanation window, the system provides the intermediate indicators of enemy activity which were used in inferring the battlefield situations. If the analyst requests a justification of these indicators, he is provided with additional explanation concerning the indicator in question. This explanation essentially articulates an English version of the rule which fired, with specific information about the unit(s) and other technical data which activated the rule embedded in the explanation. The explanation window (Figure 4) contains an explanation pane and a data display pane. The explanations appear in the former, while more detailed supporting data on situations, indicators, units and messages is provided in the data display pane.

The following example provides an illustrative subset of the explanation facilities. Suppose the system has alerted the analyst of a possible attack by the 8th motorized rifle regiment (8-MRR). By calling up the explanation window, the analyst is presented with a brief explanation (in the explanation pane) of the battlefield indicators supporting this situation (Figure 4.a). At this point, he has the option to access more detail on the indicators themselves or to request an explanation of the reasoning behind those indicators. The explanation of the RED-SUPERIOR indicator (Figure 4.b) indicates that the 8-MRR can be expected to engage friendly units within an 85 km radius within a reasonable period of time (nominally four hours), and that the opposing forces (opfor) ratios indicate decisive enemy superiority when numbers of tank, artillery and infantry units organic to the attacking enemy units and defending friendly units are compared. By selecting the boldfaced 8-MRR from either explanation, the analyst may gather historical data on the unit or see a list of the messages recently reporting on this unit, from which he may access the contents of the actual messages (Figure 4.c) reporting on the 8-MRR.

Figure 5 shows an example of graphical explanation support. Referring to the RED-SUPERIOR indicator explained in Figure 4, the 8-MRR is shown in the center of a circle of radius 85 km, and the defending friendly units are easily observed within this radius of potential attack. The analyst's attention is quickly drawn to the units and areas of interest by presenting units critical to the inference of a specific battlefield situation in highlighted or blinking modes. The graphic explanation subsystem also include display of spatial and temporal templates, which allow the analyst to visually perceive the degree of template match/mis-match calculated by the corresponding template matching functions embedded within certain rules.

Contained within every textual explanation are system-supplied mouse-sensitive items, such as unit names, which



ATTACK-DEFENDING-ENEMY-SITUATION-5
EXPLANATION PANE

ATTACK-DEFENDING-ENEMY-SITUATION-5

(a)

Situation: **ATTACK-DEFENDING-ENEMY-SITUATION-5**
Critical Unit: **8-MRR**
Location: LAT:052.13/LON:-151.80
Inferred due the following indicators:

RED-STATIONARY-INDICATOR-1
RED-SUPERIOR-INDICATOR-5

Rule GROUND-ACTIVITY-AND-RED-STATIONARY fireo. Red stationary indicator was inferred (1.0) due to observation of critical ground unit (**8-MRR**). Last heard containing lat/long within 4 hours, with greater than 20 minutes since previous message. Distance between two updates is less than 4 km.

(b)

Rule GROUND-ACTIVITY-AND-RED-SUPERIOR fired. Red superior indicator was inferred (1.0) due to observation of critical ground unit (**8-MRR**), last heard with lat/long within 4 hours, and with more than 1 blue unit within the units search radius (85 km). The tank (2.0), artillery (2.0) and infantry (1.5) opfor ratios are each greater than 1.

The unit 8-MRR was driven by messages:

8-MRR-MSG-2 **8-MRR-MSG-1**

ATTACK-DEFENDING-ENEMY-SITUATION-5
DATA DISPLAY PANE

Relevant data for **RED-STATIONARY-INDICATOR-1:**

CURR-ACTY:	
TIME:	032083//125600z
LOCATION:	LAT:052.13/LON:-151.80
CONFIDENCE:	1.0
EXPIRATION:	
GND-UNITS:	8-MRR
AIR-UNITS	
HISTORY:	032083//125600z//LAT:052.13/ LON:-151.80///032083//120400z// LAT:051:84/LON:-151.22

(c)

Relevant data for **8-MRR-MSG-1:**

ID:	8-MRR-MSG-1
EFFTIME:	079,1136z
REAL-FORMAT:	UNCLASSIFIED ACODEX//EXERCISE TEST_DATA MSGID/TACREP// ORG/UNID:8MRR/PARUID:82MRD/ ACTY/NIL/EFFTIME:079.1136Z// GNDOP/151 48 29 W 52 07 38 N NNNN
UNIT-ID:	8-MRR
PAR-UNID:	82-MRD
ACTIVITY:	
OP-TYPE:	GNDOP
QUANTITY:	
UNIT-TYPE:	
LATITUDE:	52 7 38 N
LONGITUDE:	151 48 29 W
SPEED:	
HEADING:	
FCP:	
RCP:	
MCP:	

Figure 4. Explanation of RED-SUPERIOR Indicator

are dynamically supplied by the system. The analyst may access these bold-faced items to obtain additional detail on the item in question. This allows the analyst to proceed logically through a rationalization process, and the system supplies him only the level of detail that he requires.

Although most of the initial cues to expected enemy activity are provided in alphanumeric form on the textual display, the analyst will be able to rely on the graphic display to verify or strengthen his conclusion, or -- in some cases -- to reverse his decision. In other cases, the analyst will be able to visualize on the graphic display a situation developing which has not yet prompted an alert condition. Similar to the capability discussed in the preceding paragraph, the analyst can access relevant data by selecting unit icons from the graphic display in order to gain enough detail to assist him in validating or rejecting the system-derived conclusion.

In addition to the integrated explanation capability described above, the analyst may at any time access information from the knowledge base through a fertile knowledge base access facility resident in the main window of the textual display. Exploiting the hierarchical structure of the knowledge base, the analyst may review rules, tactical data, messages, inferences and predicted situations to support his analysis process.

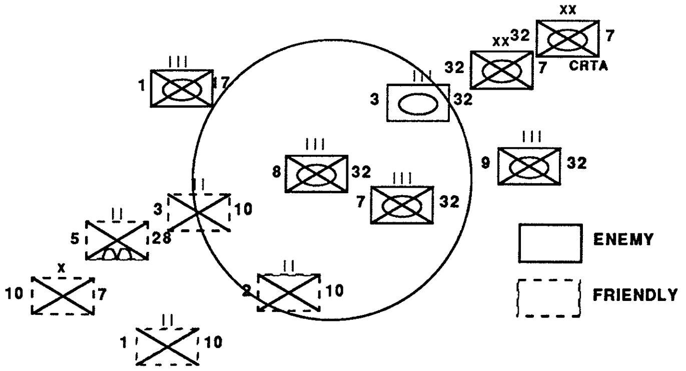


Figure 5. Graphical Support of RED-SUPERIOR Indicator Explanation.

POSSIBILISTIC INFERENCE

One of the key technical challenges in expert systems development is accounting for the manifold sources and types of uncertainty which contribute to the confidence in the inferences that the system derives. The system essentially attempts to determine an ordered 7-tuple (unit-id, location, activity, activity-conf, predicted-activity, predicted-activity-conf, predicted-activity-time), where each component has an inherent error which contributes to the overall uncertainty of the result. This effort focused only on the confidence components (activity-conf, predicted-activity-conf) of this "uncertainty vector"; i. e., we did not not address measurement errors associated with location, for example.

In battlefield situation assessment, this uncertainty in the nature of the activity itself is manifest at nearly every stage of the process. Referring to Figure 1, note that the system must consider unreliability in the message data, the technical data and the rules when inferring the intermediate clues (indicators) of enemy activity. The synthesis of these indicators, each providing evidence toward at least one potential situation, must also take into account the resulting uncertainty in the indicators. Rules for increasing the confidence due to doctrinal template matching must consider the degree of match and factor this into the confidence calculation.

Table 2 summarizes the choice of uncertainty management techniques for each step of the situation assessment process. An important point to make is that each technique captures the experts' belief regarding the relative importance that a single piece of evidence has toward confirming or disconfirming a given hypothesis. For example, the Dempster-Shafer technique for evidential reasoning requires -- for each piece of evidence -- definition of a basic probability assignment (bpa) which maps a probability mass of 1 into the set of all possible subsets of a frame of discernment (hypothesis space). [1] Although the technique is powerful and flexible in its representation of ignorance, the definition of the bpa depends entirely on the judgement of the expert, since no empirical results exist to establish the more desirable a priori conditional probabilities.

TASK	TECHNIQUE
Infer indicators	Confidence in indicator - embedded in rule(s) - assigned by experts. Contribution of incoming reports degrades over time.
Synthesize indicators into situations	Dempster-Shafer technique for evidential reasoning. Basic probability assignments defined by experts.
Apply templates	Match/mis-match: Increase/decrease situation confidence for all relevant units using ad hoc techniques.
Predict subsequent situations	Confidence propagation multipliers defined by experts.
Validate inferred situations	Filter situations based on previous expectations.

Table 2. Uncertainty Management Techniques Implemented.

The Dempster-Shafer technique was implemented for the synthesis of battlefield indicators into situations. Each indicator can be viewed as a piece of evidence giving support to any subset of the universe of likely situations. The algorithm suggested by Barnett handles the somewhat restricted domain in which each piece of evidence (indicator) supports or refutes exactly one singleton hypothesis (situation), but fails when this condition does not hold, i .e, when a piece of evidence supports or refutes more than one hypothesis. [2] A technique was developed which extends the Barnett algorithm to include this more general case, while still avoiding the computational burden of the full-blown Dempster-Shafer model. (This technique will be the subject of a subsequent paper, currently in progress.)

In the application of spatial templates to an aggregation of doctrinally related units, the synergism of tactical units acting in concert with one another is modelled by an ad hoc technique employed and suggested by the experts. These templates are defined and applied for ATTACK, MARCH and ASSEMBLY situations. A template match results in the confidence C in the template situation T being increased by updating the corresponding belief by $\Delta C = (1\text{-Bel}(T))/2$. Confidence in the other competing hypotheses not supported by the template are reduced by an appropriate normalization factor. In the case of a template mismatch, the belief in the situation T is halved, with the remaining confidence assigned to the universe of possible situations. In the validation step, situation confidences are handled implicitly by the filtering of observed and inferred situations against a list of situations previously predicted.

FUTURE DIRECTIONS

Several applications have been identified for expanding and demonstrating the knowledge-based system technology developed for this prototype. The enhanced prototype situation assessment expert system for intelligence analysis support will incorporate the integrated textual/graphic man-machince interface discussed above, a refined inference engine to include backward chaining to allow user-initiated queries regarding potential enemy activities, and an expanded knowledge base to encompass higher-level "interpretive" rules that synthesize system-derived conclusions about current and predicted activities of doctrinally related enemy units. These are discussed in greater detail in the sections below

User-Interface. In moving toward a comprehensive intelligence analysis support capability, we plan to develop a reporting facility by which the analyst could access previously generated explanations from the explanation display window and supporting technical data from the data display window in preparing a summary to his commander.

Further enhancing the rich explanation environment will be a system query function, in which the user can make inquiries regarding the potential activities of the enemy before they occur. This capability will be implemented using a goal-directed reasoning approach called backward chaining, and will not only provide qualitative answers to the queries but will identify potential evidence which would further substantiate the analyst-specified goal.

Meta-Level Knowledge. When building expert systems, the initial emphasis is on the ease of adding rules. As the complexity of the system increases, domain-specific knowledge often finds its way into the inference engine, thus encumbering that part of the system which has a great impact on runtime effieciency. Domain-specific control knowledge must be incorporated external to the actual inference process in order to streamline the search for the most likely conclusions.

The current control strategy of the inference engine is breadth-first, implemented via a driver list or queue for storing incoming data and resulting inferences. Meta-knowledge intelligently controls higher level invocation of procedures such as the Dempster-Shafer confidence combination scheme. This procedural knowledge, currently included implicitly in the inference engine, will be explicitly specified in meta-rules. This approach will yield an inference engine that makes control knowledge more accessible to the system builders and that is much less domain-specific, thus increasing the efficiency of this major system component.

Uncertainty Management. Another major system refinement will be the extension of the system's current capability to reason under uncertainty. The current system employs the Dempster-Shafer method for the combination of multiple sources of evidence to arrive at confidence values for a set of competing hypotheses, with unlikely hypotheses removed from contention when their confidence falls below a user-specified threshold. Based upon subsequent evidence, the confidence in these hypotheses are modified using ad hoc methods. A more comprehensive approach will be taken in the enhanced version, and will improve the strategy for updating the confidence due to the matching of aggregated doctrinally related units with pre-stored spatial templates for certain situations. The criteria for and degree of template match must be considered in such an approach. As the rule base expands to include higher level rules, the mechanism for uncertainty management will be modified to synthesize the confidence values from all supporting evidence.

REFERENCES

1. J. Gordon and E. Shortliffe, "The Dempster-Shafer Theory of Evidence," in B. Buchanan and E. Shortliffe (eds.), Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project, Reading, MA, Addison-Wesley, 1985.

2. J. A. Barnett, "Computational Methods for a Mathematical Theory of Evidence," Proceedings of the 7th International Joint Conference on Artificial Intelligence, August 1981.

SYSTEM BOUNDING - A CALIBRATION THEORY ON C^3I MISSION ANALYSIS

by

LT Bruce M. Nagy, USN

NRL Code 9110
Washington, D.C. 20375
(202) 767-3184
15 September 1986

ABSTRACT

System Bounding is a theory on command, control, communications and intelligence mission analysis by qualitatively defining the potential hardware and software limitations of a point design interfacing with the command and control environment. System Bounding has three phases. Phase I consists of establishing a benchmark to calibrate the system. In phase II, a hardware and software network model is developed from the design specifications. Using the "soft" benchmark, modular dependencies from functional paths are created. With phase III, applicable hardware and software analysis is accomplished using the simulation models data. The result of phase III is the calibration of the existing design's potential performance.

1. Introduction

1.1 Purpose

System Bounding theoretically provides a structure to analyze a command, control, communication and intelligence (C^3I) system consisting of computer hardware and software units. The result of System Bounding is a group of grades associated with the performance of the system as related to its operational use. The operational use is defined by oracles describing its performance characteristics. Oracles are derived from Operational Requirements (OR), Top Level Requirements (TLR), and Required Operational Capability (ROC) documents. The oracles are used to support the Modular Command and Control Evaluation Structure (MCES) in defining the performance characteristics. In other words, System Bounding provides a method to analyze the quality of the design. System Bounding also provides the flexibility to analyze large systems, comprised of smaller subsystems, or subsystems comprised of smaller units. Therefore, the purpose of System Bounding is to analyze a hardware, software structure objectively and qualitatively.

1.2 Background

System Bounding is a theory on mission analysis of an integrated hardware and software product by defining the potential quality of the product in terms of its user capabilities and limitations. It was developed in conjunction with the MCES for data synthesis and data aggregation when using simulation [Ref(1)]. The MCES is used as a methodology for defining the measures of force effectiveness (MOFEs), measures of effectiveness (MOEs), and measures of performance (MOPs) for C^3I Systems. These measures aid in defining a benchmark in chich to gauge the capability of the design.

1.3 Requirements

There are two requirements critical for System Bounding success. The first is that the operational interface between the system under analysis and its corresponding environment must be rigorously defined. Definition must be in the form of stimulus to the system and its desired response. A detailed description of the stimulus, response pairs is required for an adequate "soft" benchmark. Much of the information can be synthesized from system/functional specifications. Yet insight into the validated operational requirements, from sources outside the specification, may be valuable (e.g. ORs, ROCs, and TLRs). Additionally, the large amount of research involved with developing the system specification can also aid in describing the stimulus, response pairs. Without a detailed stimulus, response pair, the benchmark used in System Bounding will not provide the calibration standard necessary for thorough analysis. The MCES provides the methodology to organize the material to define the stimulus, response pairs in C^3I terms.

The second critical requirement is that the system, under analysis, must have adequate design specifications. The design specifications must be able to support a point design. The hardware specifications must describe each software module's function and input/output (I/O) requirements. This is necessary in order to adequately represent the system in a hardware and software network model. The model will be used to describe the functional paths resulting from the stimuli associated with the benchmark. If the model does not accurately represent the design, the functional paths may not accurately represent the responses to the stimuli. Therefore, the accuracy of the results is directly related to the accuracy of the model.

2. Theory

2.1 Assumptions

System Bounding has two sets of results. The first is the identification of the functional paths through the hardware and software under analysis, via the model, using the original stimulus, response pairs. The second is the results of the initial stimulus, response pairs to determine the potential bounds of the design. Therefore, System Bounding assumes that a hardware and software network model can be created from the design specifications. The model must have the following properties:

- o accurately reflecting point design detail,
- o accepting stimulus to produce functional path execution,
- o providing accurate response(s) resulting from functional path execution,
- o automated data aggregation techniques associated with functional path execution, and
- o timeliness of representing a point design into the model.

Once the functional paths are created, there is a variety of techniques available to the analyst in finding potential problems during the execution of the path [Ref(2-7)] by relating it to the stimulus, response paire (i.e. applicable oracles). Some software techniques include symbolic analysis [Ref(8,9)], functional testing [Ref(10)], and perturbation testing [Ref(11)]. Modeling tools provide analysis techniques to identify hardware faults [Ref(12,13)]. The System Bounding approach provides the functional paths resulting from the stimuli. Additionally, it provides the desired responses to compare with the model responses in order to aid the analyst in determining hardware and software faults. It is assumed that the hardware and software analysis techniques will provide the appropriate tools to determine the potential operational capability of the point design.

2.2 Implementation

System Bounding consists of three phases. In phase I, the system is defined in its operational environment. The operational environment and system interfaces are described in detail. The result of phase I is stimulus, response pairs in the form [s,r] where,

$s:s \in S$, $S = \{$Environmental Stimuli/System Specification$\}$
$r:r \in R$, $R = \{$System Responses/System Specification$\}$

Each pair will represent a "soft" benchmark to compare with the design's performance. An environmental stimulus may have a variety of responses. Likewise, stimuli may be required to produce a single response. Therefore, a pair can be written in the form $[(s_1,s_2\ldots,s_j),(r_1,r_2\ldots,r_k)]$ where n and k are determined from system analysis using the operational interface requirements.

Phase II is the translation of the design specifications into a hardware and software network model of the system. The model must represent software modules, processing elements, data transfer devices, and data storage devices. Figure 1 represents a basic structure of a network model.

Example of Intermodular Paths for Stimulus/Response Pairs

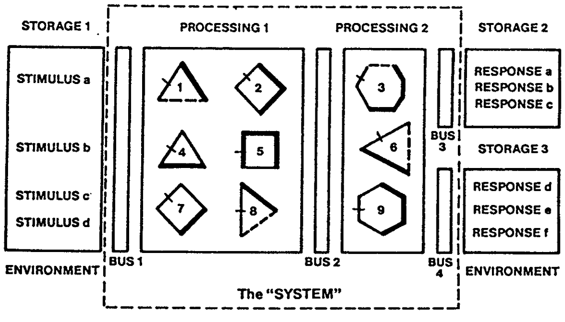


Figure 1. Basic Structure of a Hardware and Software Network

In phase II, the model is stimulated and analyzed using the stimulus, response pairs defined in phase I. Analysis consists of tracing the stimulus input through the software modules, processing elements, data transfer devices, and data storage devices to the resultant output. The resultant output is then compared with the desired result. Figure 2 shows an example of the functional paths created by the stimuli and resultant responses. The software modules in the figure are geometrically represented by polygons. The sides of the polygons are either input arguments (thin lines), output arguments (dark lines), or not applicable (dotted lines). Each polygon represents a software module function.

Example of Intermodular Paths for Stimulus/Response Pairs
{[a, (b, f)], [b, c], [c, (a, e)], [d, d]}

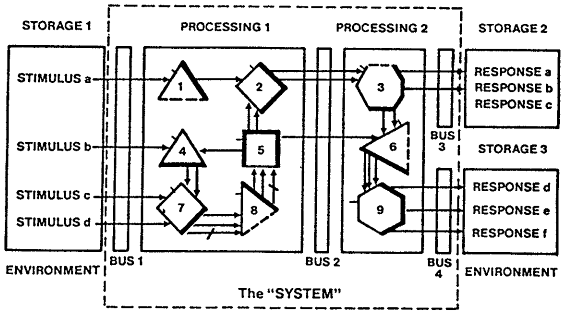


Figure 2. Functional Paths within a Hardware and Software Network

With phase III, the functional paths, identified by the stimuli, are analyzed for software and hardware limitations with respect to the desired responses. Hardware and software analysis techniques will be selected based on the type of discrepancy between the model executed response and desired environmental response. The analysis will result in fault stimuli and fault responses as a potential capability bound for the system. A fault defines the potential limitation characteristic(s) of the design. Each fault will relate to a software module associated with a hardware unit. This relation is represented as:

$$mh,\ldots mi_{[(s_{n,1}\ldots s_{n,j}),(r_{n,1}\ldots r_{n,k})](al,\ldots al)}$$

$m:m \in M$, $M = \{$Software Modules$\}$,

$a:a \in A_m, A_m = \{$Input/Output Arguments passed per Module$\}$,

$s_n:s \in S_s, S_s = \{$Fault Stimuli$\}$,

$r_n:r_n \in R_r, R_r = \{$Fault Responses$\}$

where S_s and R_r separates each set of test pairs under the parent [s,r] pair.

As an example, in Figure 2 there are four stimulus, response pairs, These pairs would be derived using the MCES and oracles discussed earlier.

(1) [sa,(rb,rf)]
(2) [sb,rc]
(3) [sc,(ra,re)]
(4) [sd,rd]

$S = \{sa,sb,sc,sd\}$
$R = \{ra,rb,rc,rd,re,rf\}$

Each stimulus, response pair has a functional path related to software modules.

(1) $m1,2,3,6,9_{[sa,(rb,rf)]}$

(2) $m4,7,8,5,2_{[sb,rc]}$

(3) $m7,8,5,2,3,6,9_{[sc,(ra,re)]}$

(4) $m7,8,6,5,4,7,8,5,6,9_{[sd,rd]}$

where

$M = \{m1,m2,m3,m4,m5,m6,m7,m8,m9\}$

During functional path analysis, each mocule or its associated hardware can be identified with a potential fault by using $[(s_{n,1}\ldots s_{n,j}),(r_{n,1}\ldots r_{n,k})]$ pairs. Since hardware is driven by software, an appropriate module can also be defined for fault stimulus in identifying possible hardware limitations in the design. Using Figure 2, there are fault stimulus and response pairs identified for each module. Therefore,

(1) For ${}^{m1,2,3,6,9}[sa,(rb,rf)]$, fault pairs are

${}^{m1}[sa_1,(rb_1,rf_1)]_{a(1,1),a(1,2)}$,

${}^{m2}[sa_2,(rb_2,rf_2)]_{a(2,1),a(2,2)}$,

${}^{m3}[sa_3,(rb_3,rf_3)]_{a(3,1),a(3,4),a(3,6)}$,

${}^{m6}[sa_4,(rb_4,rf_4)]_{a(6,1),a(6,3)}$, and

${}^{m9}[sa_5,(rb_5,rf_5)]_{a(9,2),a(9,6)}$.

(2) For ${}^{m4,7,8,5,2}[sb,rc]$, fault pairs are

${}^{m4}[sb_1,rc_1]_{a(4,1),a(4,3)}$,

${}^{m7}[sb_2,rc_1]_{a(7,2),a(7,3)}$,

${}^{m8}[sb_3,rc_1]_{a(8,1),a(8,2)}$,

${}^{m5}[sb_4,rc_1]_{a(5,4),a(5,2)}$, and

${}^{m2}[sb_5,rc_1]_{a(2,4),a(2,7)}$.

(3) For ${}^{m7,8,5,2,3,6,9}[sc,(ra,re)]$, fault pairs are

${}^{m7}[sc_1,(ra_1,re_1)]_{a(7,1),a(7,3)}$,

${}^{m8}[sc_2,(ra_2,re_2)]_{a(8,1),a(8,2)}$,

${}^{m5}[sc_3,(ra_2,re_2)]_{a(5,4),a(5,2)}$,

${}^{m2}[sc_4,(ra_3,re_3)]_{a(2,4),a(2,2)}$,

${}^{m3}[sc_5,(ra_4,re_4)]_{a(3,1),a(3,3)}$,

${}^{m6}[sc_6,(ra_5,re_5)]_{a(6,1),a(6,3)}$, and

${}^{m9}[sc_7,(ra_6,re_6)]_{a(9,2),a(9,4)}$.

(4) For ${}^{m7,8,5,4,7,8,5,6,9}[sd,rd]$, fault pairs are

${}^{m7}[sd_1,re_1]_{a(7,4),a(7,3)}$,

${}^{m8}[sd_2,rd_2]_{a(8,1),a(8,2)}$,

${}^{m5}[sd_3,rd_3]_{a(5,4),a(5,1)}$,

${}^{m4}[sd_4,rd_4]_{a(4,2),a(4,3)}$,

${}^{m7}[sd_5,rd_5]_{a(7,2),a(7,3)}$,

${}^{m8}[sd_6,rd_6]_{a(8,1),a(8,2)}$,

${}^{m5}[sd_7,rd_7]_{a(5,4),a(5,3)}$,

${}^{m6}[sd_8,rd_8]_{a(6,1),a(6,3)}$, and

${}^{m9}[sd_9,rd_9]_{a(9,2),a(9,3)}$.

where

$S_{sa} = \{sa_1,sa_2,sa_3,sa_4,sa_5\}$,
$S_{sb} = \{sb_1,sb_2,sb_3,sb_4,sb_5\}$,
$S_{sc} = \{sc_1,sc_2,\ldots sc_7\}$,
$S_{sd} = \{sd_1,sd_2,\ldots sc_9\}$,
$R_{ra} = \{ra_1,ra_2,\ldots ra_7\}$,
$R_{rb} = \{rb_1,rb_2,\ldots rb_5\}$,
$R_{rc} = \{rc_1\}$,
$R_{rd} = \{rd_1,rd_2,\ldots rd_9\}$,
$R_{re} = \{re_1,re_2,\ldots re_6\}$,
$R_{rf} = \{rf_1,rf_2,\ldots rf_5\}$

and

A = {a(1,1), a(1,2), a(1,3), a(2,1), a(2,2), a(2,3), a(2,4), a(3,1), a(3,2), a(3,3), a(3,4), a(3,5), a(3,6), a(3,7), a(4,1), a(4,2), a(4,3), a(5,1), a(5,2), a(5,3), a(5,4), a(6,1), a(6,2), a(6,3), a(7,1), a(7,2), a(7,3), a(7,4), a(8,1), a(8,2), a(8,3), a(9,1), a(9,2), a(9,3), a(9,4), a(9,5), a(9,6)}

where

a(1,3), a(3,2), a(6,2), a(8,3), a(9,1), a(9,6) are not applicable sides. Additionally, a(3,5) and a(3,7) are not part of a functional path (see Figure 2) and describe class two and three observations.

The arguments are labeled per each module in a clockwise direction where a(m,k) is m = module number and k = I/O argument for the module. k = 1 is indicated by a notch in the polygon.

Before using analysis techniques, four classes of observations can be made in analyzing the functional path. Each class is represented in Figure 3. Class one deals with paths that have deadends, i.e. paths that end with a module and without a response to the environment. Class two concerns a module with no output path. Class four deals with hardware that stops a path. Each observation class can result in a $[r_n, s_n]$ pair to identify a potential fault in the design.

Example of Intermodular Paths for Stimulus/Response Pairs
{[a, (b, f)], [b, c], [c, (a, e)], [d, d]}

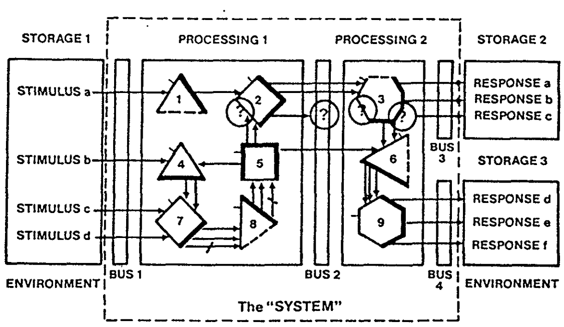


Figure 3. Observations of Potential Faults within a Hardware and Software Network

By using the above notational relationships, overlapping paths resulting in multiple $[r_n, s_n]$ pairs can be identified and eliminated. An overlap is defined as a module having the same I/O requirements among various. modules. Every fault pair resulting from the four observations should take priority in determining the potential bounds of performance. This is because each observation reflects gross design errors which could cause multiple problems associated with path execution. Additionally, due to this type of potential flaw in causing multiple functional path execution errors, the deadend path should be checked along the entire route. Consequently, overlapping fault pairs with identifying the potential limitations of the entire route is derived.

Using Figure 3 as an example, all fault pairs for the m6 module are eliminated except for an arbitrary choice of the fault pair derived from [sc,(ra,re)]. Obviously, in actual analysis, elimination would be based upon the worst case $[s_n, r_n]$ pair for that module. Even though overlap occurs for fault pairs associated with [sb,rc], all overlap $[sb_n, rb_n]$ pairs remain. This is in addition to other fault pairs from other [s,r] environmental pairs. Finally, the fault pair for module 8 is under [sc,(ra,re)] is eliminated (arbitrarily for this example) due to the same path execution for [sd,rd].

If a fault pair is desired to specifically identify a hardware problem, the following relationship occurs:

$$hw(ni,...nj)_{[s_n, r_n]a(ni,k),...a(nj,l)}$$

where

$$hw:hw \in HW = \{Storage_{1,...}, Processing_{1,...}, Bus_{1,...}\}$$

The notation can become cumbersome, yet it lends itself to uniquely identifying the relationship between the system environment with its software and hardware component. Since fault identification is derived, the notation also relates potential faults of the system with its operational requirement. The notation is an aid in describing the relationships. A database management scheme may fascilitate the actual System Bounding analysis.

After all fault pairs have been defined, the fault stimulus is executed in the network model. The responses are analyzed to determine system bounding from the data. The responses are then associated with the fault stimulus. Therefore a $[s_n, r'_n]$ pair is created, where

$$r'_n : r'_n \in R'_r,\ R'_r = \{\text{Performance Bound Responses}\}$$

When the performance bound response is compared with the determined fault response a determination can be made as to the quality of the design as related to its potential performance. Additionally, the quality can be directly related to the hardware units involved as per the general notation

$$hw(mi,...mj)_{[s_n, r'_n]a(mi,k),...a(nj,l)}$$

Obviously s_n and r'_n can represent multiple fault stimulus and performance bound response pairs. Although the performance bound results have a binary answer, i.e. pass or fail, fuality should not be represented by a binary result. The binary answer is used to determine the variations of stimulus associated with the design, where the variations are determined by the values used during the functional path execution of a [s,r] pair. Therefore, the value contained in the [s,r], $[s_n, r_n]$, and $[s_n, r'_n]$ can be compared to determine the quality of the design by using the following nine definitions:

Definition 1: S_s (S iff $\forall s_n, \forall s$: minbound(s) $< s_n \vee s_n <$ maxbound(s)

where $\text{maxbound}(s) \rightarrow (\lim_{\Delta s -) 0} \frac{\Delta f(s)}{\Delta s} > 0)$,

$\text{minbound}(s) \rightarrow (\lim_{\Delta s \rightarrow 0} \frac{\Delta f(s)}{\Delta s} < 0)$,

and f(s) is the function describing the system associated with the environmental stimulus, where f(s) = r.

Definition 2: $S_s = S$ iff $\forall s_n, \forall s: s_n = s$

Definition 3: S_s) S iff $\forall s_n, \forall s: s_n >$ maxbound(s) $\vee\ s_n <$ minbound(s)

Definition 4: R_r (R iff $\forall r_n, \forall r$: minbound(r) $< r_n \vee r_n <$ maxbound(r)

where $\text{maxbound}(r) \rightarrow (\lim_{\Delta r \rightarrow 0} \frac{\Delta g(r)}{\Delta r} > 0)$,

$\text{minbound}(r)$ -) $(\lim_{\Delta r \rightarrow 0} \frac{\Delta g(r)}{\Delta r} < 0)$, and g(r) is the function describing the system associated with the required environment response, where g(r) = s.

Definition 5: $R_r = R$ iff $\forall r_n, \forall r: r_n = r$

Definition 6: $R_r\ )\ R$ iff $\forall r_n, \forall r: r_n > \text{max-bound}(r) \vee r_n < \text{minbound}(r)$

Definition 7: $R,r\ (\ R_r$ iff $\forall r'_n, \forall r_n{:}\text{min-bound}(r_n) < r'_n \vee r'_n (\text{maxbound}(r_n)$

where $\text{maxbound}(r_n) \rightarrow \left(\lim_{\Delta r_n \rightarrow 0} \frac{\Delta g(r_n)}{\Delta r_n} > 0 \right),$

$\text{minbound}(r_n) \rightarrow \left(\lim_{\Delta r_n \rightarrow 0} \frac{\Delta g(r_n)}{\Delta r_n} < 0 \right),$

and $g(r_n)$ is the function describing the design associated with the required fault response, where $g(r_n) = s_n$.

Definition 8: $R'_r = R_r$ iff $\forall r'_n, \forall r_n: r'_n = r_n$

Definition 9: $R'_r\ )\ R_r$ iff $\forall r'_n, \forall r_n: r_n > \text{max-bound}(r_n) \vee r'_n < \text{minbound}(r_n)$

With the above definitions, the following theorems result:

Theorem 1: $(R'_r\ )\ R_r \wedge (R_r\ )\ R) \rightarrow (R'_r\ )\ R)$

Proof: From Definitions 9 and 6, $r'_n > \text{max-bound}(r_n)$
and $r_n > \text{maxbound}(r)$, respectively, therefore
$r'_n > \text{maxbound}(r)$. Likewise,
$r'_n < \text{minbound}(r_n)$, $r_n < \text{minbound}(r)$, therefore
$r'_n < \text{minbound}(r)$. $(R'_r > R)$ representation is
$r'_n < \text{minbound}(r) \vee r'_n > \text{maxbound}(r)$.
(*)

Theorem 2: $(R'_r= R) \wedge (R_r = R) \rightarrow (R'_r = R)$

Proof: From Definitions 8 and 5, $r'_n = r_n$ and
$r_n = r$, respectively, $\therefore r'_n = r$.
(*)

Theorem 3: $(R'_r\ (\ R) \wedge (R_r\ (\ R)$ -) $(R'_r\ (\ R)$

Proof: From Definitions 7 and 4, $r'_n < \text{max-bound}(r_n)$
and $r_n < \text{maxbound}(r)$, respectively, therefore
$r'_n < \text{maxbound}(r)$. Likewise,
$r'_n > \text{minbound}(r_n)$, $r_n > \text{minbound}(r)$, therefore
$r'_n > \text{minbound}(r)$. $(R'_r\ (\ R)$ representation is $\text{minbound}(r) < r'_n \vee r'_n <$ maxbound(r).
(*)

Caliber assignment can now result:

- ° Caliber 1 $\rightarrow (S_s\ )\ S) \wedge (R'_r\ )\ R)$
- ° Caliber 2 $\rightarrow (S_s\ )\ S) \wedge (R'_r = R)$
- ° Caliber 3 $\rightarrow (S_s\ )\ S) \wedge (R'_r\ (\ R)$
- ° Caliber 4 $\rightarrow (S_s = S) \wedge (R'_r\ )\ R)$
- ° Caliber 5 $\rightarrow (S_s = S) \wedge (R,_r = R)$
- ° Caliber 6 $\rightarrow (S_s = S) \wedge (R'_r\ (\ R)$
- ° Caliber 7 $\rightarrow (S_s\ (\ S) \wedge (R'_r\ )\ R)$
- ° Caliber 8 $\rightarrow (S_s\ (\ S) \wedge (R'_r = R)$
- ° Caliber 9 $\rightarrow (S_s\ (\ S) \wedge (R'_r\ (\ R)$

where S and R are the calibration benchmarks.

Now if $\text{dmn}(e) = \{s{:}[s,r] \in e\}$,
$\text{rng}(e) = \{r{:}[s,r] \in e\}$,
and $\text{dmn}(t) = \{s_n{:}^{u}[s_n,r'_n] \in t\}$,
$\text{rng}(t) = \{r_n{:}^{u}[s_n,r'_n] \in t\}$
where $u{:}u \in \text{Unit}$, $\text{Unit} = \text{HW} \cup \text{M}$

Then, in grouping [e,t] pairs by dmn and rng according to one of the nine caliber relationships, the hardware and software units will likewise be grouped. Consequently, the system's software modules, processing elements, data storage devices, and data transfer devices are defined in accordance with the quality of their performance as related to their design dimensionality. A caliber cumulation of the results with respect to each hardware and software component in the system can describe a profile of the design's potential performance bounds (caliber) as related to its operational requirements. Figure 4 represents an example of each categories profile. Obviously, a cumulation of caliber 1 profiles is desired. When the profiles are complete, the calibration process is finished.

3.0 Significance

The accomplishment of phase I forces a thorough understanding of the system being designed. This understanding is in the form of identifying the stimulus, response pairs associated with the environmental interfaces. This result may direct specification changes or enhancements. Additionally, the synthesis of stimulus, response pairs formally reflects the operational requirements of the design by the user/developer. This initial perception may be different from the customer's viewpoint.

The completion of phase II allosw a dynamic interpretation of the operational requirements to the design specifications.

The hardware and software network model creates an execution representation of the dynamic application of the design specification to its operational interface. Obviously the specification dynamics affects both customer, user and developer in providing more effective products.

The main result of phase III is the identification of the potential performance bounds of the system using calibration in terms of hardware and software dimension limitations with respect to its operational requirements. This provides the developer with specifics concerning potential faults in the design. Likewise, it provides the customer with an understanding of the system's potential operational performance. Therefore, System Bounding can create a major impact on product development and operational capability.

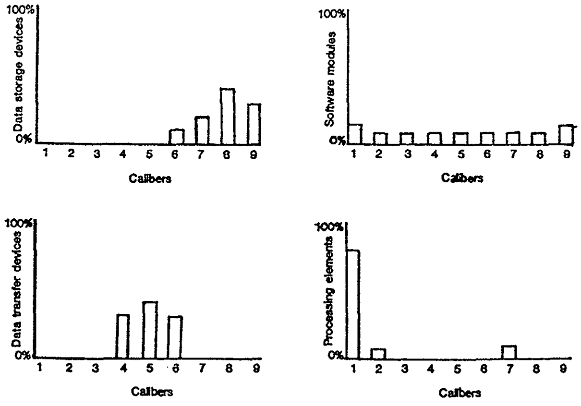


Figure 4. Example of four calibration profiles

References

1. Ricki Sweet, Preliminary C^2 Evaluation Architecture, AFCEA Transaction in Signal, January 1986. pp. 71-73.
2. John S. Gourlay, Introduction to the Formal Treatment of Testing, Symposium on Software Validation (1983 : Darmstadt, Germany) Software Validation, Elsevier Science Publishers B. V., 1984. pp. 67-72.
3. Daniel P. Siewiorek, C. Gordon Bell, Allen Newell, Computer Structures: Principles and Examples, McGraw-Hill Computer Science Series, 1982.
4. Daniel P. Siewiorek, Robert S. Swarz, The Theory and Practice of Reliable System Design, Digital Equipment Corps., 1982.
5. William Hetzel, The Complete Guide to Software Testing, QED Information Sciences, Inc., 1984.
6. S. E. Goodman, An Introduction to the Analysis of Algorithms, Applied Mathematics and Computer Science, University of Virginia, ±984. pp. 169-173.
7. Sabina H. Saib, Stephen W. Smoliar, Software Quality Assurance For Distributed Processing, Proceedings of the Fifteenth Hawaii International Conference on System Sciences, Volume I, Software, Hardware, Decision Support Systems, Special Topics, 1982. pp. 79-85.
8. Lori A. Clarke, Debra J. Richardson, Symbolic Evaluation - An Aid to Testing and Verification, Symposium on Software Validation (1983 : Darmstadt, Germany) Software Validation, Elsevier Science Publishers B. V., 1984. pp. 141-166.
9. Erhard Ploedereder, Symbolic Evaluation as a Basis for Integrated Validation, Symposium on Software Validation (1983 : Darmstadt, Germany) Software Validation, Elsevier Science Publishers B.V., 1984. pp. 167-185.
10. William E. Howder, A Functional Approach to Program Testing and Analysis, Electrical Engineering and Computer Sciences, University of California at San Diego, 1986.
11. Steven J. Zeil, Testing for Perturbations of Program Statements, IEEE Transactions on Software Engineering, SE-9, 3, 1983. pp. 335-346.
12. Domencio Ferrari, Giuseppe Serazzi, Alessandro Zeigner, Measurement and Tuning of Computer Systems, Prentice Hall, Inc., 1983.
13. William J. Garrison, Network II.5 User's Guide, CACI, Inc., September 1985. pp. 1-1 to 1-6.

MEASUREMENT OF VALUE ADDED BY THE MANEUVER CONTROL SYSTEM

Philip Feld, Director C^3I Division

Defense Systems, Inc.
7903 Westpark Drive
McLean, Virginia 22102

INTRODUCTION

The Maneuver Control System (MCS) is an automated computer system being developed to assist the G3/S3 and their staffs in support of command and control (C^2) responsibilities pertaining to the maneuver control function. Moreover, MCS is expected to be the initial node fielded in the Army Command and Control System (ACCS) and must interface with both joint and allied C^2 systems.

Evaluation of MCS is a challenging assignment. Previous operational evaluations have not provided adequate information or the data necessary to assess the combat utility MCS brings to the commander and his staff. Moreover, as a C^2 system, the MCS is being developed under an evolutionary acquisition (EA) concept. On the one hand, this provides the flexibility necessary to adapt the system to changes in its operating environment, and to use lessons learned in the evaluation of one phase to improve the system in later phases. On the other hand, it complicates evaluation because the MCS is changing across evaluation opportunities. Finally, achieving reliable, valid measures of the performance of C^2 systems and their impact on military operations has, in the past, proven difficult in itself.

Recognizing these difficulties, the U.S. Army Operational Test and Evaluation Agency (USAOTEA) is adapting the Headquarters Effectiveness Assessment Tool (HEAT) and applying it to measure the value added by the MCS to Army tactical C^2.

HEADQUARTERS EFFECTIVENESS ASSESSMENT TOOL

HEAT is a set of consistent procedures which measures the effectiveness of a military headquarters or command node. In the context of an MCS evaluation, the "headquarters" is considered to be the aggregate of the individual command nodes (i.e., TAC, MAIN, and REAR Command Posts) that support a commander in the performance of C^2 functions, in combat operations. HEAT supports quantitative, objective, and reproducible assessment of both the quality of the processes by which information is used by the commander and his staff in decisionmaking (and of the systems which support the processes), and the overall effectiveness of the decisions made and their implementation. The essence of HEAT is a set of definitions of measures, including a small number of measures covering overall headquarters effectiveness, and a much larger menu of diagnostic measures covering specific parts of the headquarters process.

HEAT has been applied to field exercises involving joint forces (including Army Division size elements), naval battle group and fleet exercises, exercises by the Military Airlift Command, nuclear control elements in NATO exercises, and a variety of laboratory experiments in C^2. It has proven remarkably robust across this range of applications. While the technique must be modified for applications with different purposes, the core methodology consistently provides quantitative, objective, and reproducible assessments of the quality of the C^2 processess observed.

CONCEPTS UNDERLYING HEAT

The HEAT measures are based on a view that headquarters are analogous to an adaptive control system which seeks to impact the environment (own forces, enemy forces, and physical elements such as weather and distance) by means of the plans (or directives) that it issues to its subordinates. This view implies that the effectiveness of the headquarters can be judged by the viability of its plans. Good plans can be executed without need for modification beyond the contingencies built into them and remain in effect throughout their intended life. Alternatively, the headquarters may find that its plans (in decreasing order of effectiveness):

- require minor adjustments in the course of their execution, without change to the basic plan;
- require execution of a contingency, significantly different from the intended course of action, but provided for in the initial plan; or
- require cancellation and issuance of an entire new plan.

As illustrated in Figure 1, HEAT breaks down the task of C^2 (i.e., preparation and execution of a plan) into six basic processes:

- monitoring what is happening in the environment (physical situation, enemy situation, friendly situation);
- understanding what is happening (hypotheses about the characteristics of current or emerging situations that have tactical or strategic significance);
- generation of alternate courses of action, i.e., actions by which those significant features of future situations might be changed or preserved (one course of action is no action);

- predicting the impact of each course of action on the future situation, including likely adversary reaction;
- deciding what course of action or combination of courses of action to take; and
- directing the execution of those decisions.

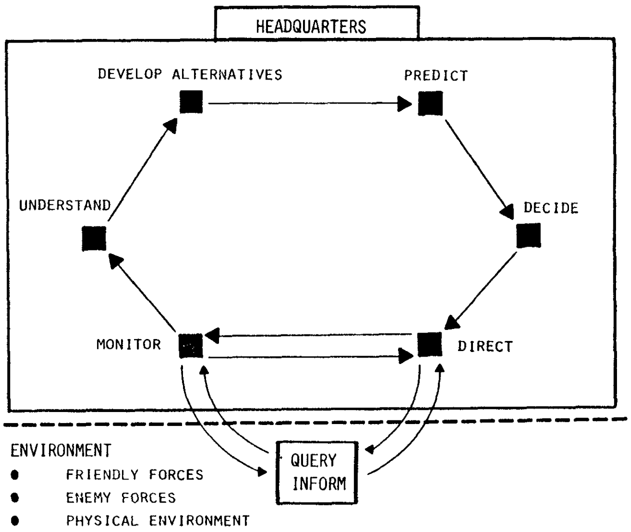


FIGURE 1. The Headquarters Cycle

Together, these steps constitute a full headquarters cycle or planning cycle. If a plan is being modified rather than created afresh, some of the steps may be omitted. As a further part of the headquarters' interaction with its environment, it may find itself:

- informing superior, subordinate, and adjacent headquarters;
- querying its monitoring elements about incomplete information; and
- responding to queries from superior, subordinate, and adjacent headquarters.

HEAT addresses the performance of a headquarters both in the planning cycle as a whole and in the separate process steps.

Figure 2 shows a set of notional scores for selected monitoring functions developed over a period of five days during a field exercise involving a joint command. Note that the initial problems of training and lack of procedural expertise depressed scores for most functions early and that the scores then generally rise over time to some fairly stable level. Some functions, such as understanding the enemy's intentions, never do get high on an absolute scale. The scores (which can be disaggregated into much greater detail if desired) create a picture of performance of key C^2 activities.

HEAT PROCESS FUNCTION	1	2	3	4	5	COMMENT
● ENEMY GROUND LOCATION	.52	.66	.50	.45	.50	FLAT LEARNING CURVE: ● WEAK PROCEDURES
● ENEMY GROUND STRENGTH	.60	.62	.74	.77	.82	HIGH START, STEEP LEARNING CURVE: ● STRONG TRAINING ● STRONG PROCEDURES
● ENEMY AIR STRENGTH	.13	.49	.52	.60	.65	LOW START, STEEP LEARNING CURVE: ● WEAK TRAINING ● STRONG PROCEDURES
● UNDERSTANDING ENEMY	.25	.32	.32	.40	.35	LOW START, FLAT LEARNING CURVE: ● WEAK TRAINING ● WEAK PROCEDURES ● POOR INTEL AND COMMS
● OWN GROUND LOCATION	.11	.43	.63	.62	.70	LOW START, STEEP LEARNING CURVE: ● WEAK TRAINING ● STRONG PROCEDURES
● OWN AIR STRENGTH	.20	.50	.74	.68	.32	LOW START, STEEP LEARNING CURVE, LOW END: ● WEAK TRAINING ● STRONG PROCEDURES ● ENDEXITIS

FIGURE 2. Summary of HEAT Scores for Monitoring Functions Observed During an Exercise Involving a Joint Command

MCS EVALUATION STRATEGY

As stated earlier, the MCS is the first in a series of developing Command, Control, and Subordinate Systems which form a subset of the ACCS. The context of these developmental efforts is the streamlined material acquisition process; the technology integration process; and the continuous, comprehensive evaluation (C^2E) process. The goal of the evaluation strategy is to provide for initial evaluation of value added by the MCS to the C^2 process, followed by the continued evaluation of MCS value added as it evolves from a single subsystem into an integrating node of the force-level system. Because the system is being developed under an EA concept, it is essential that the evaluation strategy provide continuing feedback to permit diagnosis of problems and identification of opportunities for additional value added, as well as answering the fundamental value added issue over time, with different versions of MCS.

There are at least four approaches that can be used to evaluate the performance of C^3I equipment, among them:

- "Happiness" indices, wherein operators and commanders are surveyed as to their opinion of the value of the equipment.

- Limited functional testing, wherein specific facets of the equipment are tested against design standards, usually in a laboratory or test bed configuration.

- Measurement against design specifications in a field environment.

- Measurement of performance in an operational or exercise environment wherein the equipment is tested for its overall performance in simulated combat conditions.

Of these, the most definitive and comprehensive evaluation is provided by measuring performance during an exercise, either a command post exercise (CPX) or field training exercise (FTX), involving units employing the equipment. Short of war, evaluation during an exercise provides the most realistic assessment of whether the equipment will work in combat, and how well it will work in its intended environment.

The basis of the MCS evaluation strategy is the use of CPX/FTX for the conduct of a series of Follow-on-Evaluations (FOE). However, exercises themselves should not be the only methodology considered for evaluation over time. If the EA strategy is to work, feedback must be provided from testing and evaluations of later developmental stages of equipment to the earlier stages of development of follow-on equipment. This implies a requirement for evaluation of equipment at developmental stages other than fully operational. The money and manpower required for the conduct of a CPX/FTX argues against their use for other than full operational evaluations. Therefore, four alternate, lower cost, more focused approaches to support evaluation between full field evaluations have been identified.

- Test Beds provide a unique opportunity for the testing of new systems and procedures in a controlled man-machine environment. They also provide the capability to evaluate performance of a command node using new equipment by allowing human interaction with a system model.

- Computer simulations are useful to explore machine attributes in a system whose overall structure and behavior are well-known.

- Mathematical models to identify optimal solutions and key parameters for system redesign may prove useful. Such models can save time and effort, but these results must be validated in simulations, laboratories, or field exercises.

- Analysis of previous exercise reports could be valuable to the extent that available data could somewhat reduce the time and expense involved in the conduct of new exercises.

The evaluation strategy assumes that in addition to CPX/FTX opportunities, test beds and laboratories will be available to support MCS evaluations. Typical of the facilities envisioned are the Advanced Technology Test Bed (ATTB) and the ACCS Division Level Test Bed (ADLBT) located at the Army Development and Employment Agency (ADEA), the Total Systems Tactical Validation Test Bed (TSTV) located at III Corps, and the Experimental Development Demonstration and Integration Center (EDDI), being developed at Fort Leavenworth.

The basic philosophy is that the evaluation program will be iterative, consisting of an overall sequence of tests and evaluations that provides feedback from each phase of MCS development to the early phases of follow-on MCS and other ACCS elements. It is highly unlikely that any one test for evaluation in the sequence will produce results that are conclusive and totally satisfactory. For example, analysis of results may imply the need for further evaluation to improve confidence in the results, modifications to the evaluation plan in order to investigate new topics, modified or new equipment to correct deficiencies, or some combination of these approaches.

The concepts of value added and of iterative testing and evaluation require the establishment of a performance baseline against which to quantitatively compare subsequent improvements. Once a solid baseline has been established, operational evaluations will be conducted on units equipped with succeeding versions of the MCS participating in a CPX/FTX. Test bed experiments and simulations will be used in conjunction with exercise results to verify insights obtained and to aid in the determination of high payoff areas for future MCS development. Figure 3 contains the current schedule to evaluate the value added by MCS using HEAT. The baseline will involve the manual C^2 system employed by the 1st Infantry Division; FOE I will employ elements of the 1st Armored Division and VII Corps using current versions (i.e., current at the time of the evaluation) of both MCS hardware and software.

• BASELINE	SEP '86	1ST INFANTRY DIVISION
• FOE-I	JAN '86	VII CORPS
• FOE-II	JUL '88	VII CORPS

FIGURE 3. Current MCS HEAT Evaluation Schedule

DETERMINATION OF VALUE ADDED

The primary purpose of applying HEAT to evaluate the performance of commands using the MCS, is to determine the value added to C^2 performance by the MCS. Therefore, it is imperative that rigorous methods be employed to ensure that any change in performance can be attributed specifically to the MCS or to some other factor. This is particularly true with an evaluation strategy which includes considerable differences (as it will be between the baseline and FOE) in the forces employed and the exercise environment and scenario used. Since there are a number of factors or variables that can contribute to a change in performance with a headquarters or to a difference in performance between two similar commands, appropriate data must be collected on each of these variables during each exercise. These variables include:

- Control variables--these are factors influencing the values of the MOE, but not a result of MCS performance. These are divided into two groups: environmental variables--those that describe the environment in which MCS operates, and human factors variables--those that describe those factors directly affecting the ability of personnel to accomplish their tasks.

- System activity variables--those variables affected by employment of MCS which in turn affect the value of the MOE. These are defined in terms of workload.

The actual process of determining the value added by the MCS is based on a comparison of the change in MCS HEAT scores from one exercise to another (e.g., from baseline to FOE I) with developed hypotheses. Two sets of hypotheses are developed:

- Hypotheses which predict how the MCS would impact performance are developed prior to an exercise.

- Hypotheses which predict how differences in control and system activity variable data obtained from each exercise would impact performance. These are developed after the exercise, but prior to development of MCS HEAT scores.

Once the exercise is completed, changes in individual MCS HEAT scores are compared with the hypotheses. In cases where the hypotheses are not validated, further analysis is used to determine why the scores were not as predicted.

MCS HEAT evaluations of the 1st Infantry Division (Baseline) and then the 1st Armored Division (FOE I) will not result in a single number which reflects the absolute value added by MCS. Rather, these evaluations will describe how MCS impacted (for good or ill) overall performance and the individual processes which support decisionmaking, and provide insights into the significance of these impacts and the reasons behind them.

COMMAND AND STAFF DECISION AIDS

Martha L. Robinette, MAJ Steven R. Accinelli,
Derek J. Konczal, and MAJ Jerome A. Jacobs

U.S. Army TRADOC Analysis Center-Ft. Leavenworth
ATTN: ATOR-CSC-F
Ft. Leavenworth, KS 66027-5220

Summary

Command and Staff Decision Aids is a U.S. Army Training and Doctrine Command (TRADOC) project being worked by the U.S. Army TRADOC Analysis Center at the Combined Arms Center, Ft. Leavenworth, Kansas. Project purpose is to improve U.S. Army command and control (C2) effectiveness by research, analysis, and development of automated applications to support critical C2 functions. Completed project products include a technical report, G3 Analysis, which documents analysis and prioritization of analytic aiding opportunities for the operations officer (G3) at corps and division levels, a unit movement planning aid prototype, MOVEPLAN (1.0), for corps level and below, and a technical document the MOVEPLAN users manual. An ongoing part of the project is development of an enhanced aid, MOVEPLAN (1.5).

An analysis of the G3 section of U.S. Army corps and division main command posts (G3 Main) was performed to identify and prioritize analytic aiding opportunities to support the G3 during tactical operations through the use of computer applications. The analysis and assessment process was based on the near-term (five-year) automated environment of main command posts and current U.S. Army doctrine. A structured functional analysis was performed to identify specific G3 Main tasks and products and then to assess opportunities to aid G3 performance. A prioritization methodology was refined and exercised to develop a recommended priority to conduct research and to develop analytic aids. The G3 Analysis may be helpful in refining requirements for software support of automated C2 systems.

A microcomputer-based aid to assist in unit movement planning was also developed. The MOVEPLAN prototype, MOVEPLAN (1.0), is written in Basic and runs on an IBM or compatible microcomputer W/CRT, a disk drive, and a printer. Components include the MOVEPLAN (1.0) Users Manual and the 5 1/4-inch diskette, MOVEPLAN (1.0). Based on embedded equations and movement rate guidelines from field manuals, along with some user inputs, the program produces a movement table, a column analysis of pass times, and a description of the route (in terms of lengths and travel rates). MOVEPLAN (1.0) (A Unit Movement Planning Aid) Users Manual provides general information and a detailed tutorial to users of MOVEPLAN (1.0).

G3 Analysis

Purpose

The purpose of the G3 analysis was to identify opportunities for aiding the performance of the operations officer (G3) at corps and division levels during tactical operations through the use of computer applications. The G3 analysis was performed by the U.S. Army Combined Arms Operations Research Activity (CAORA), now called the TRADOC Analysis Center-Ft. Leavenworth, during the period January-July 1985. The G3 analysis was performed to assist the Combined Arms Combat Developments Activity (CACDA) to refine requirements for software applications on tactical automated systems.

Approach

The general approach employed to identify and prioritize opportunities for aiding the performance of the G3 during tactical operations was a structured functional analysis of the G3 section of U.S. Army corps and division main command posts (G3 Main). The structured function analysis focused on the doctrinal G3 Main tasks and products to develop qualitative assessments of aiding opportunities. The analysts recognized that the specific manner of task performance and the forms of products may vary from command to command, but that underlying opportunities for aiding performance have potential for transfer across commands.

Objectives

The following objectives were established to accomplish the G3 analysis: (1) identify the G3 Main critical tasks; (2) identify the G3 Main products which are supported by the critical tasks; (3) identify a taxonomy of aiding technologies; (4) assess the potential of identified technologies to aid G3 Main performance; (5) for those products which require analytic aiding technologies, assess the appropriateness of alternative analytic techniques; (6) develop a methodology for prioritizing analytic aiding opportunities; (7) prioritize analytic aiding opportunities based on appropriate criteria; and (8) document the analysis with appropriate findings and recommendations.

Methodology and Results

G3 Main Critical Tasks. Four primary doctrinal documents were analyzed to identify and define G3 Main critical tasks. A comparison matrix was organized to reflect the documents which validated each task. The matrix facilitated identification of gaps or differences between documents. Seven major G3 functions composed of a total of 43 critical tasks were identified and compiled. Differences across documents were not significant; however, the potential utility of a single comprehensive, doctrinally approved G3 critical task list was highlighted by the analysis. A separate reference sheet was prepared to delineate the key elements of each major function and critical task. The reference sheets were key documents which supported the assessment of aiding opportunities.

G3 Main Products. A detailed analysis of G3 tasks, doctrinal literature, tactical standing operating procedures (TSOP), and the Command Information Database (CID) (which maps tasks to the products they support) resulted in the compilation of G3 information products, both formal and implied. Formal products were defined as standard documents produced and disseminated by the G3. Implied products were materials generally developed by the G3 for internal use or for informal coordination. Forty-eight formal products and 11 implied products were identified for the G3 Main. The relationship of task, subtask, and product was key in the analysis process which led to the identification of opportunities to aid during the development of G3 products.

Assessment of Aiding Opportunities. A classification scheme of aiding technologies was developed using computer science, information system, and decision support literature. The taxonomy decomposed aiding technologies into information processing techniques, user interface techniques, and analytic techniques. Analytic techniques were further subdivided into categories of artificial intelligence (AI), mathematical models, optimization techniques, computer simulations, and decision analysis. Analysts made an assessment of the specific aiding technologies which could be applied to G3 Main products. In some cases, particularly in AI technologies, a positive assessment could not be made due to the relative immaturity of the technology. However, for most products and technologies, a positive assessment was possible. In some cases, a single product might be supported by more than one analytic aid. Analysts assigned a descriptor (name) to each analytic aiding opportunity. A total of 53 different analytic aiding opportunities were identified at this point in the analysis.

Prioritization of Analytic Aid Candidates. A prioritization methodology was developed based on an investigation of alternative techniques for structuring preferences. Thomas L. Saaty's analytic hierarchy process [10] provided an objective method to obtain a priority value for each individual aid. A hierarchy of separable criteria was formulated, and a method of pairwise comparisons was used to determine the relative utility (weight) of each criteria. Primary criteria were feasibility and importance, with three subcriteria under each. A commercial software application, "Expert Choice," facilitated the computation of criteria values. The 53 analytic aiding opportunities were prioritized based on adjusted ranks. The ranks were based on the total adjusted score for each aiding opportunity. A graphical display of aid scores was developed to examine the distribution of aids over the scoring spectrum. A leaf plot of adjusted scores showed that the adjusted scores had a single mode, were slightly skewed toward higher scores, and that the distribution of scores was approximately normal.

Sensitivity Analysis. A limited sensitivity analysis was performed to examine the relationships between adjusted scores, raw scores, scores based solely on feasibility, and scores based solely on importance. Graphical analysis was the primary technique employed to investigate sensitivity. Comparison of leaf plots showed that five analytic aids consistently scored in the top two cells across all scoring schemes. The specific aiding opportunities are: Air Movement Analyzer, Fuel Consumption Rates, Assign Critical Replacements, Unit Movement Planner, and Force Movement Analyzer. Comparison of scatter plots showed that the top four aids were dominant (low rank) for both raw and adjusted ranking procedures. Further, the bottom five aids were consistently inferior. Aids in the midrange are highly sensitive to the effects of alternative subcriteria weights.

Documentation. The G3 analysis was documented in a two-volume technical report, CAORA/TR-13/85, in December 1985.

MOVEPLAN (1.0)

Purpose

The purpose of MOVEPLAN (1.0) was to provide a flexible microcomputer-based aid to assist in unit movement planning. It was developed at the request of Commander, CAORA and was worked independently of the G3 analysis, which identified a unit movement planner as a high-priority analytic aiding opportunity. The requirement for a movement plan--ning or time/distance analysis tool was also identified during the AirLand Battle Study. MOVEPLAN (1.0) was designed to automate many of the manual procedures involved in unit movement planning.

Approach

Several alternative approaches were investigated to satisfy the requirement for a movement planning tool. The alternatives included simple spreadsheet equations and a U.S. Army Command and General Staff College (CGSC) application. However, the alternatives were restrictive in design and did not deal with the variety of movement conditions encountered during tactical operations. MOVEPLAN (1.0) is a computer simulation which accounts for most of the conditions for a single-route tactical movement. However, MOVEPLAN (1.0) does not do all the steps in the movement planning process. The user must gather data on units, the route, and unit movement TSOP and then must input this information on a microcomputer. The microcomputer computes the data and prints a report containing a movement table and a summary of inputs. MOVEPLAN (1.0) provides the benefits of responsiveness and accuracy over manual procedures. A primary source of doctrinal information used in MOVEPLAN's development was FM 55-30, Army Motor Transport Units and Operations. Most movement planning doctrine and "schoolhouse" procedures are taught with the fundamental assumption of constant pace or rate of march. While this assumption is acceptable for administrative moves, it is not appropriate for tactical movement across different types of roads and cross-country terrain under both day and limited visibility conditions. Consideration of varying movement rates over a route was a principal factor in the design of MOVEPLAN. As a result, the effects of queuing or backup along the route are represented in MOVEPLAN.

Model Operation and Organization

MOVEPLAN may be used by U.S. Army tactical movement planners at levels from section to corps. MOVEPLAN inputs are obtained from map inspection, subordinate unit reports, and unit TSOP. MOVEPLAN outputs are used by the movement planners and are an enclosure to the Movement Annex of an operations order (OPORD) or a Letter of Instruction to subordinate units. MOVEPLAN is an interactive software application. MOVEPLAN's software components are primarily an input routine, a computation routine, an output routine, and help and file maintenance routines. The software is driven by user-selected menu options and creation or recall of data files on units and routes.

Capabilities

1. Handles up to 10 serials with 10 march units in each serial.

2. Handles a single route with up to 20 road segments.

3. Provides a capability for rest/refueling halts for all units at specific checkpoints.

4. Provides a capability to limit speeds on each road segment.

5. Provides a capability to plan movement under three alternative march disciplines, described below.

a. Hasty with fixed start intervals: Each march unit travels as rapidly as conditions will allow but does not pass units ahead of it. Each march unit starts at a fixed interval behind the unit ahead of it.

b. Hasty, with staggered starts: Each march unit travels as rapidly as conditions will allow but does not pass units ahead of it. Starts are staggered automatically to eliminate congestion on the route.

c. Control move: Force/march column integrity is maintained. Units start at fixed intervals. Slowest movement condition affecting the force dictates the force march rate.

6. Provides a pace for each march unit for each road segment as a guideline for the pace vehicle and for the movement planner.

7. Computes the average speed for the lead march unit across the route.

8. Computes the (static) column length and (static) vehicle density.

9. Computes due-in and release times for each march unit for each checkpoint.

10. Summarizes the movement by serial.

11. Summarizes the route description.

Limitations

1. Does not handle multiple start points or multiple release points.

2. All march units must occupy designated rest/refueling halts.

3. Does not allow for a variable maximum pace for each type unit (wheeled vs tracked vs mixed vehicle units).

4. Does not consider oversize or overweight vehicles and route limitations.

5. Does not deal with transition conditions (day to night).

6. Does not perform route selection or analysis tasks.

7. Does not perform tactical movement planning on a multiple-route network, since it is a single-route processing model.

Inputs

MOVEPLAN (1.0) requires information to be provided by Army tactical movement planners according to a data input sheet, which the user can have printed out by the program and which facilitates organization of data prior to entering it into the machine. This information consists of data file name, maximum pace, vehicle interval, average vehicle length, march unit interval, serial interval, route name, number of road segments in the route, segment maximum rates and distances, number of serials, serial names, number of march units in each serial, march unit names, number of vehicles in each march unit, halt times at each checkpoint and release point, choice of march discipline, desired arrival or start time, and number of days to departure. Input intensity is highly dependent on the number of route segments and march units in the force.

Processing

Once inputs are completed, lengths of segments and movement rates on segments are established for the route. March unit lengths are computed and rest areas are established for the route. Due and clear times at each checkpoint are computed for each march unit and conflicts at checkpoints are resolved. Times are adjusted for user-defined arrival or start. The program is then ready to produce outputs.

Outputs

MOVEPLAN (1.0) is a responsive software application that provides real-time tactical movement information. Response time is less than five minutes for an average-size force on a route with less than six different route segments. MOVEPLAN (1.0) produces a road movement table, a column analysis of pass times, and a description of the route in terms of lengths and travel rates.

Summary

MOVEPLAN (1.0) is a flexible, responsive, transparent software application which significantly reduces the time and error associated with tactical movement planning. MOVEPLAN (1.0) was released through the Command and Control Microcomputer Users Group (C2MUG) at Fort Leavenworth, KS, in May 1986.

MOVEPLAN (1.5)

Purpose

The purpose of MOVEPLAN (1.5), an enhanced version of MOVEPLAN (1.0), is to provide a flexible microcomputer-based aid to assist in unit movement planning. The proposed MOVEPLAN (1.5) will plan ground movements of multiple units along multiple routes, from multiple assembly areas to multiple final positions.

Proposed Capabilities

Work on MOVEPLAN (1.5) is ongoing. Proposed capabilities include the following:

1. Permit multiple march units to travel from multiple starting positions to multiple final positions.

2. Generate a movement order.

3. Move march units using one of two march disciplines, fixed start or staggered start.

4. Account for rest/halt areas.

5. Select the "best" route for each unit to travel.

Summary

The Command and Staff Decision Aids Project has consisted of three major parts: (1) the G3 analysis, a functional analysis of the operations (G3) section of U.S. Army corps and division levels to identify and prioritize opportunities to support the G3 during tactical operations through the use of analytic computer software applications; (2) MOVEPLAN (1.0), a unit movement planning decision aid prototype, which assists in moves for multiple units over a single route with multiple segments, from common start point to common release point; and (3) MOVEPLAN (1.5), an enhanced movement planning aid, which will assist in moves of multiple units along multiple routes from multiple assembly areas to multiple final positions. Written products include a technical report, CAORA/TR-13/85, G3 Analysis and a technical document, CAORA/TD-4/86, MOVEPLAN (1.0) (A Unit Movement Planning Aid) Users Manual).

References

1. Andriole, Stephen J., et al. Decision aids for Command and Control (C2). Seminar Notebook. Burke, Virginia: The Armed Forces Communications and Electronics Association, September 1984.

2. Boar, Bernard H. Application Prototyping. New York: John Wiley & Sons. Inc., 1984.

3. Expert Choice. McLean, Virginia: Decision Support Software, Inc., 1983.

4. Hillier, Frederick S., and Lieberman, Gerald J. Introduction to Operations Research. Oakland, California: Holden-Day, Inc., 1980.

5. Hussain, Donna, and Hussain, K. M. Information Processing Systems for Management. Homewood, Illinois: Richard D. Irwin, Inc., 1981.

6. International Busines Machines Corporation. Information Systems Planning Guide. White Plains, New York: July 1984.

7. Miller, Allen C., III, et al. Analytic Procedures for Designing and Evaluating Decision Aids. DTIC Technical Report, April 1980.

8. MITRE Corporation. The Command Control Subordinate System (CCS2) Cross-Segment Functional Analysis. Working Paper. 2 vols. McLean, Virginia: 30 November 1984.

9. Moder, Joseph J., and Elmaghraby, Salan E., eds. Handbook of Operations Research. New York: Van Nostrand Reinhold Company, 1978.

10. Saaty, Thomas L. Decision Making for Leaders. Belmont, California: Lifetime Learning Publications, 1982.

11. Science Applications, Inc. SAI-84/1569, Guidelines for Autmating Command and Control Functions in Field Units. Interim Research Product 84-07. Ft. Leavenworth, Kansas: Army Research Institute, March 1984.

12. Sprague, Ralph H., Jr., and Carlson, Eric D. Building Effective Decision Support Systems. Englewood Cliffs, New Jersey: Prentice - Hall, Inc., 1982.

13. U.S. Army Combined Arms Combat Developments Activity. Division Commander's Critical Information Requirements (CCIR). Study Report, 30 April 1985.

14. U.S. Army Combined Arms Operations Research Activity. Division/Corps Information and Communication Flow Analysis. Technical Report CAORA/TR-1-85, January 1985.

15. U.S. Army Command and General Staff College. FC 71-100, Armored and Mechanized Division and Brigade Operations, May 1984.

16. ____________________. FC 100-8, A Guide to the Applications of the Estimate of the Sitaution in Combat Operations, April 1984.

17. ____________________. FC 100-34, Operations on the Integrated Battlefield, July 1984.

18. ____________________. FC 101-55, Corps and Division Command and Control, 28 February 1985.

19. ____________________. FM 100-15, Corps Operations, Final Draft, 10 October 1984.

20. ____________________. RB 101-5, Staff Organization and Operations, May 1983.

21. U.S. Army Institute for Research in Management Information and Computer Science. Contract No. DAHC06-82-R, ADP Technical Support Services - Task 1. Ft. Belvoir, Virginia: U.S. Army computer Systems Command, 1982.

22. U.S. Army Training Doctrine Command. TRADOC Regulation 71-2: Force Development, TRADOC Post-Deployment Software Support (PDSS) Program. Fort Monroe, Virginia: 12 February 1982.

23. ____________________. "Soviet Research on the Use of Computers in Troop Control and Decision Making," 271404Z Ft. Monroe, Virginia: February 1984.

24. U.S. Naval Academy Operations Analysis Study Group. *Naval Operations Analysis*. Annapolis, Maryland: Naval Institute Press, 1977.

25. U.S. Department of the Army. ARTEP 3-387, *Chemical Company (Heavy Division)*, 9 December 1983.

26. ____________________. ARTEP 5-145, *Engineer Battalion Mechanized and Armored Divisions*, 29 December 1980.

27. ____________________. ARTEP 100-2, *Division Command Group and Staff*, 15 June 1978.

28. ____________________. FM 21-40, *NBC Defense*, 14 October 1977. 29. . FM 90-4, *Airmobile Operations*, 8 October 1980.

30. ____________________. FM 101-5, *Staff Organization and Operations*, 25 May 1984.

31. ____________________. TOE 87004J410, *HHC Heavy Division (Army of Excellence)*, 1 April 1984.

32. Young, Donovan, *Brief Usability Survey of Operations Research Applications Software for Decision Support Systems*. Report in Fulfillment of Scientific Services Agreement, D.O. No. 0919, 22 December 1978.

AN EVOLVING C2 EVALUATION TOOL - MCES THEORY

Dr. Ricki Sweet
SWEET ASSOCIATES. LTD., P. O. Box 9196, Arlington, VA 22209

Introduction

There has been extensive interest in the continuing evaluation of C2 systems and architectures. The objective of either a C2 system or a C2 architecture is to fulfill a military mission. Attempts to grapple with the issues of the evaluation of C2 (or C3 or C3I, etc.) systems are widespread across DoD. These initiatives have generally been tied to the acquisition cycle, viewed broadly to include operational test and evaluation (OT&E) and interoperability issues. This paper presents the current MCES formulation. This evolving tool is viewed as an asset for use in solving DoD C2 problems, see Figure 1.

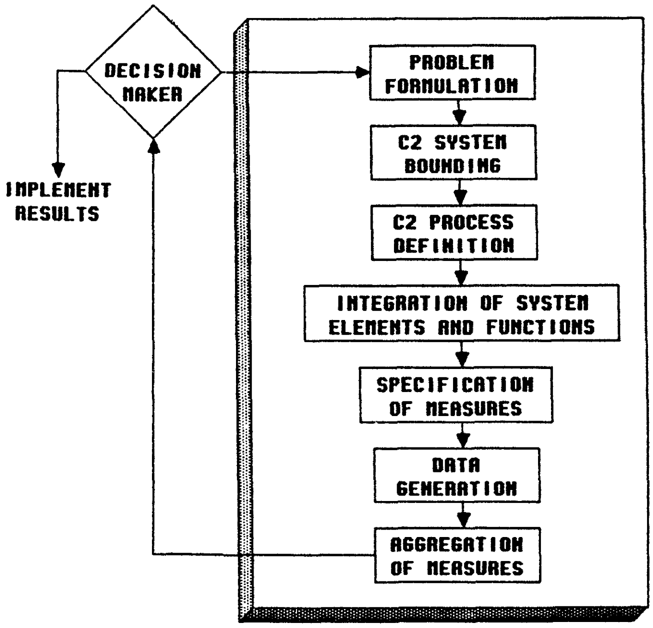


FIGURE 1
MODULAR COMMAND AND CONTROL EVALUATION STRUCTURE (MCES)

The MCES - What does it do?

The MCES expedites the analytic foundations for a whole range of reports used in all services in DoD. These reports include system design, requirements, interface and interoperability documents, critical issues reports, operational concepts, and prototype and full system evaluations.

The MCES may be viewed as two "products". "Product 1" focuses on the complete specification of the problem that is to be solved. It expedites the systematic specification of the problem by identifying essential characteristics of C2 systems. It eases the burden on decision and policy making resources by enhancing direction and reducing the time and personnel needed for both the specification and the analysis of the problem.

"Product 2" identifies, integrates and coordinates appropriate methodologies for the solution of the specified problem. A wide variety of existing tools and models are accommodated. (Two of these methodologies, whose utility was heightened through structuring with the MCES, are reported in the two papers following this presentation.) The MCES permits a senior analyst who must provide the supporting data for decision making to drive any C2 evaluation efficiently to a concise conclusion. It provides a set of standardized procedures which allow the resolution of commonly occurring analytic problems using pragmatic techniques.

These results are provided to the decisionmaker. Two courses of action are available to the decisionmaker. First, he may identify the need for further study, iterating the MCES once again. Alternatively, he may implement the results of the MCES-driven analysis to.

Module 1 - Problem Formulation

A C2 system consists of: (1) physical entities, (equipment, software, people and their associated facilities), (2) structure (organization, procedures, concepts of operation and information flow patterns), and (3) (C2) process (the functionality or "what the system is doing").

The MCES facilitates the evaluation of C2 systems. It does this by directly supporting the products necessary to make certain primary decisions by the military commander. The example provided in the Mensh paper addresses the program manager's problem of how to best integrate electro-optics technology into a C2 or C3I system.

The first MCES Module, called Problem Formulation, addresses the question of what are the objectives of the decision-maker posing the problem. Module 1 describes what these are from the standpoint of (1) the life cycle of a military (C2) system, and (2) the level of analysis prescribed. The implementation of this module results in a more precise statement of the problem being addressed, including the appropriate scenarios. The kind of problem as well as identified solutions are indicated.

The decision makers objectives generally mirror the various phases of the life cycle of a military system, namely: 1. Concept definition and/or development; 2. Design; 3. Acquisition; and 4. Operations. The selected phase must be related to the appropriate level of analysis, i.e,: 1. the mission which the system is addressing; 2. the system itself; or 3. the components of the system, the sub-systems.

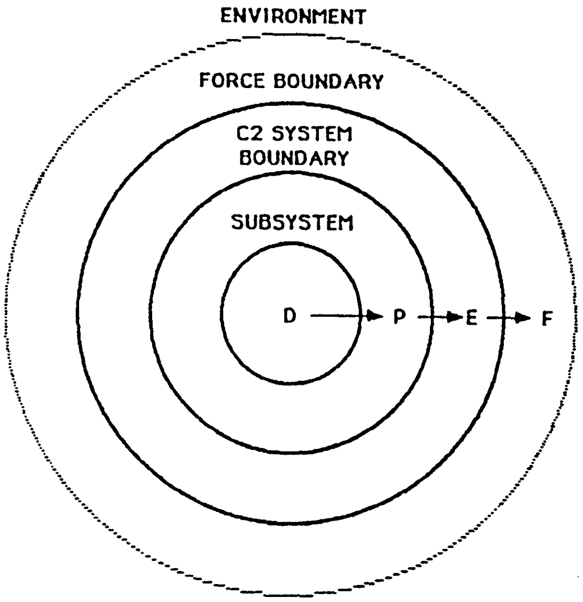
Taking the Design phase as an example, the MCES, as used in the Navy application to be presented, structures the analytic requirements for several documents. These include Required Operational Capabilities (ROCs), which support the system requirements, e.g., to detect low flying targets, and the mission areas, e.q., AAW and/or ASW. The analytic needs for both A and B System Specifications, which provide the conceptual and detailed equipment requirements, e.g., for an electro-optic device, are helped. Finally, Interface Requirements Definition (IRD), which support the interoperability of interfacing systems and sub-systems, e.g., the radar related to and improving the C2 process functions of search, identify and detect may also be enhanced. These documents are among the extensive set needed to trace through the morass of detailed studies required for decision support in the Design phase of a military system. (Indeed, both the definition of the interface and the tracing of information flow across that interface are specifically called out when using the MCES.)

Three steps take place in this module. First, the decisionmaker's needs, previously known as the Applications Objectives, are characterized. Next, the problem boundaries are selected. Finally, the remaining modules of the MCES are previewed for their potential impact on the problem statement. Working with the MCES in these problems, it was clearly found to be flexible and robust, both to the problem and the tools which exist in the mission area. The MCES also produces viable alternatives which are easily reconfigurable. In the final step in the Problem Formulation Module, the remaining modules are previewed in a quick run through.

Module 2 - C2 System Bounding

With the system elements of the problem identified and categorized, the C2 system of interest may be bounded by relating the "physical entities" and "structure" definitional components to the graphic representation of the levels of analysis, the "onion skin", see Figure 2.

There are at least two theoretical issues that are being pursued in regard to this Module. First, the issue of mapping to the C2 System Boundaries graphic will be discussed. Subsequently the relationship between system bounding and levels of analysis will be addressed. Since we have not resolved these issues, only the nature of the problem will be described.



WHERE:
D = DIMENSION
P = PERFORMANCE
E = EFFECTIVENESS
F = FORCE OUTCOME

FIGURE 2
MODULE 2: C2 SYSTEM BOUNDING

The applications of the MCES towards the problems being worked at the Naval Postgraduate School do not clearly indicate whether all of the C2 system definitional components can be bounded in the sense of this module, i.e., mappable to the "onion skin". It is clear that the physical entities are mappable to the "onion skin" graphic. The organizational structure has also been mapped with moderate success. The organizational structure has been mapped to the C2 process for the Navy Battle Group problem.

To date, the C2 process has been mapped to the types of measures prescribed, using a matrix approach to relating the process blocks or functions to the measurement typology. In this approach, the functions (subsets of the C2 process model) are related to an appropriate object prior to determining relevant measures of performance (MOP), measures of effectiveness (MOE), and measures of force effectiveness (MOFE). These objects help to make clear how the functional dimensions are related to generic MOEs, such as timeliness, accuracy, survivability, capacity, and/or percent completion.

The quantitative degree of the relationship and the actual measurement are next methodological challenges.

The other theoretical problem relates to the life cycle of a military system. Of the four phases, which are of interest, the Nagy paper being presented hereafter describes a potential technique for system acquisition. The extent of generalizability of this method of system bounding is not yet known in relation to either other acquisition phase applications or other life cycle phases in general. It is speculated that it is not appropriate at the concept definition/development phases due to fuzziness in the specification of the system. However, for design specification where a point design may be associated with an available advanced development model, this method of system bounding is both productive and cost-effective.

In Phase I of this method, the C2 system, represented by the hardware and software design specifications, is identified and related to the environmental C2 stimulus. This relationship is developed in terms of establishing boundaries to calibrate the system. Phase II translates the design specifications into a network model of the C2 system and creates modular dependencies from the functional paths of the system. Phase III analyzes the hardware and software and tests the actual system specifications against its design parameters.

Using this method of system bounding, interactions with other MCES modules also take place. There is interaction with the Specification of Measures Module in Phase I. Both the Data Generation and the System Integration modules are used in Phase II. In Phase III, some links are made to the C2 Process model.

Despite the current lack of resolution of these issues, the determination of the boundary helps us to identify what kind of measures are necessary. For the boundary between the force and the environment, MOFE are appropriate. Within the force boundary MOE are used. For the subsystem, i.e., within the boundary of the system, MOP should be employed. Finally, within the subsystem, dimensional parameters are the relevant descriptive terms.

Module 3 - C2 Process Definition

After the system is bounded and the system elements identified, the generic C2 process component of the system is identified in the next module, see Figure 3. This concept forces attention on (1) the environmental "initiator" of the C2 process, which result from a change from the desired state; (2) the internal C2 process functions (sense, assess, generate, select, plan, and direct); and (3) the input to and output from the internal C2 process and the environment, which includes enemy forces, own/neutral forces and the usual environmental components.

The applications studied, both in the scoping at the Workshop and in the subsequent expansion analyses, have shown that the relationships in terms of feedback and projection with respect to the functions, are extremely complex. At this time, the only general statement to be made is that the C2 Process will, in general, include functions such as those presented in the generic C2 Process Model. For each problem, the interactions found should be uniquely tracked.

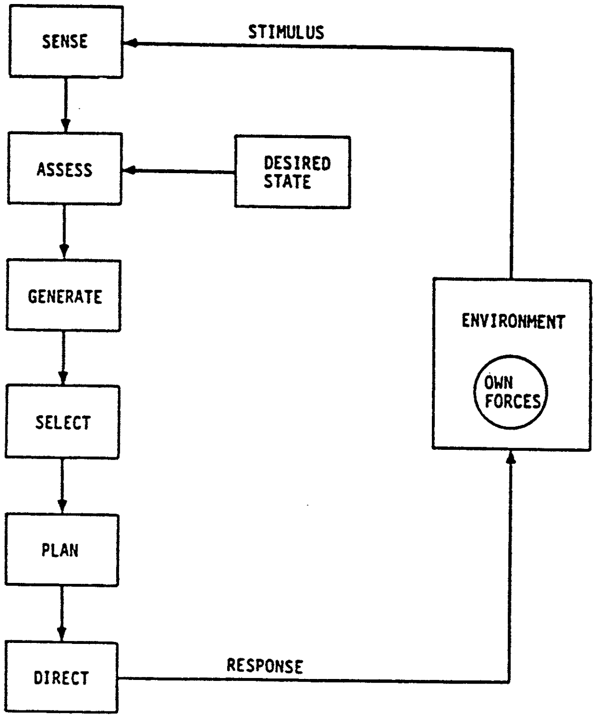


FIGURE 3
MODULE 3: GENERIC C2 PROCESS

In order to continue with the MCES Modules, it is necessary at this step to provide a translation of the vocabulary of the problem being addressed into the terminology of the C2 Process Model, e.g., Detect is equated with Sense; Track and Identify become Process.

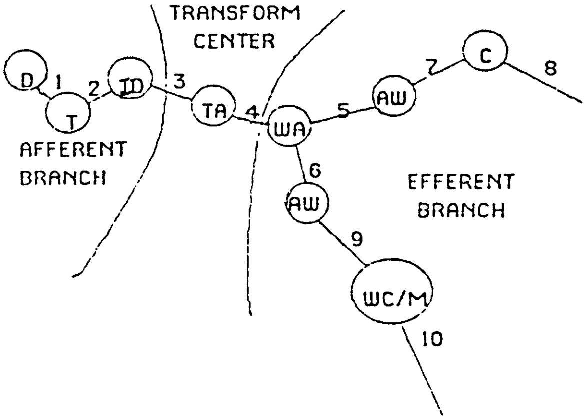
This equating of vocabularies keeps the analyst from overlooking critical process aspects, i.e., it acts as a check list on the functionality of the C2 system. Once the analyst has ascertained that all the appropriate pieces are present in the evaluation, such bridging theories as are represented by the Headquarters E Assessment Technique can be applied. The HEAT theory will provide, at the least, a set of static generic C2 organizational measures for the generic internal C2 process which therefore are mapped to the appropriate functions in the problem at hand.

In focusing on the functionality of the C2 system, the MCES may be used to indicate points of integration for new technologies, as is shown in the application study reported next. For at least one other application, the MCES was used to focus on the input and output to the C2 Process

Module, thus emphasizing the information needs into and out of the command and control system. This emphasis leads to expanding the analytic target from C2 to C3I.

Module 4 - Integration of Statics and Dynamics

Data Flow Oriented Design is a technique which can potentially form the relationships between the C2 processes, physical entities and structure, see Figure 4.



1 RADAR CONTACT
2 CONTACT/POSITION/DIRECTION
3 FRIEND/FOE/NEUTRAL
4 PRIORITIZE FOE TARGETS
5 ASSIGN TARGET TO FIGHTERS
6 ASSIGN TARGET TO SAMS
7 ALLOCATE TARGET TO FIGHTER AND CONTROLLER
8 PROVIDE DIRECTION AND INFORMATION TO FIGHTER
9 ALLOCATE TARGET TO SAM FIRE UNIT
10 MONITOR ENGAGEMENTS

FIGURE 4
MODULE 4: INTEGRATION OF STATICS AND DYNAMICS

First, Data Flow Diagrams (DFDs) are constructed to show information flow through the C2 process model. In a second step, a transform analysis is performed on the DFD. From this transform analysis you can determine the subordinate and superordinate relationships between the individual C2 functions and the transform center. Some information is coming into the transform center (afferent branch) and out of the transform center (efferent branch). Thus a hierarchical "structure" in terms of the information flow between functions within the C2 process has been defined.

The next step is to map those physical entities (man and/or machine), which perform functions and communicate output from the functions. This produces an organizational structure, which could reside in a single node where potentially all C2 functions could be performed. Alternatively, C2 process functions can be distributed between command nodes (i.e., Brigade and Battalion FDCs) or between command nodes and weapon systems (i.e., CRC and fighter.) The person and/or machine, which performs the function, may be related to the organizational structure. The result is a C3 system architecture. Physical equipment can be aligned to the same functions.

Equipment consoles could be configured to aid the operator in performing certain functions and allow the output to be addressed to other consoles. This alignment would also conform to the same structure. The operator would be aided in his ability to process information and communicate it through a machine structure that parallels an organizational structure.

All this gives us a first level model. However, many operational issues deal with the internal processing with C2 functions. In these cases, the DFD module description documents this internal processing and how the information is input and output from the function. This input/output relationship forms a description of the internal information flow between separate process functions, as required to perform the mission at hand.

Module 5 - Specification of Measures

Based upon the four prior modules, the fifth module specifies the measures necessary to address the problem of interest. The components of the C2 system definition may be employed to derive an exhaustive set of relevant measures, which are then subjected to further scrutiny. First, these are subjected to comparison with a set of criteria, which reduces the number to a more manageable set.

Then these are classified as to their level of measurement, i.e., dimensional parameters, MOPs, MOEs and MOFEs. Alternatively, instead of an exhaustive grouping, a minimum essential set may be sought. Regardless of the approach taken, the resulting measures may be used to determine the value added to the C2 system by alternative configurations of the physical entities, structure and/or processes.

Module 6 - Data Generation

Given that the measures for the functions have been identified, then we need to address the issue of how the data will be generated. In this timeline, see Figure 5, the time segments are mapped against the functions of the command and control process. Exercises, simulations, experiments and subjective judgments are all examples of data generators which can be used in the evaluation of command and control systems.

Module 7 - Aggregation of Data

The MCES enables the quantification of measures of effectiveness. However, we have not reached the stage of quantification

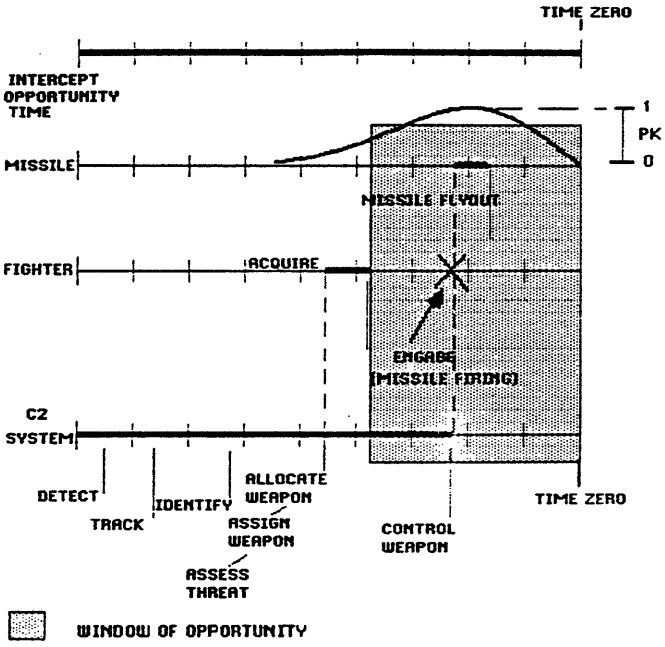


FIGURE 5
MODULE 6: DATA GENERATION

in most of the applications being worked. In fact, we have observed that when quantification of some of the concepts in some of the modules is attempted, the capability to structure analysis of the problem at hand is constrained. Therefore, our progress toward quantification of all concepts is understandably cautious.

From data generation you will presumably get values for the measures identified. Those values need to be aggregated in some way. One of the issues raised may be highlighted by exploring the desire to relate command and control systems to some measure of force effectiveness (a force multiplier effect). Thus, for MOFEs, the intent of aggregation is to relate the C2 system to combat systems. A vital question then must be addressed, potentially using the sufficiency analysis technique. Here, what we are testing is the question, "Is the probability of the combat outcome dependent just on the variables we have measured or do you need additional information from the real world or the scenario to make that decision?" Phrased somewhat differently the question can be asked " Are you willing to pick your C2 system simply upon exchange ratio or are there additional things that you would like to know about the battle.

Conclusion

The MCES will evolve over the next few years into a full blown analytic structure with which analysts within or supporting DoD can look at C2 problems efficiently and effectively. C2 problems include such questions as given several alternative C2 systems/architectures, which is "best"? For what? How do you know? What are you measuring to determine this? Against what standard? How complete is the alternative, i.e., are all the relevant aspects taken into account? How succinct is the description of the alternative, i.e., given decision-making time limitation, is the presentation at an appropriate level of detail.

Those of us who have worked toward developing the MCES and presenting its concepts to such expert audiences as attending this Symposium believe that such questions can be answered more readily by its utilization.

References

Gandee, Patrick L., "Evaluation Methodology for Air Defense Command and Control System", Naval Postgraduate School, March, 1986.

Mensh, Dennis R., "An Evolving C2 Evaluation Tool - MCES: Application."

Nagy, Bruce R., "System Bounding (C3I Mission Analysis Methodology."

Sweet, R., Metersky, M., and Sovereign, M. G., "Command and Control Evaluation Workshop", January, 1985, rev. June, 1986.

REAL-TIME DATA BASE MANAGEMENT

Dana L. Small

Naval Ocean Systems Center, Code 443
San Diego, CA 92152-5000

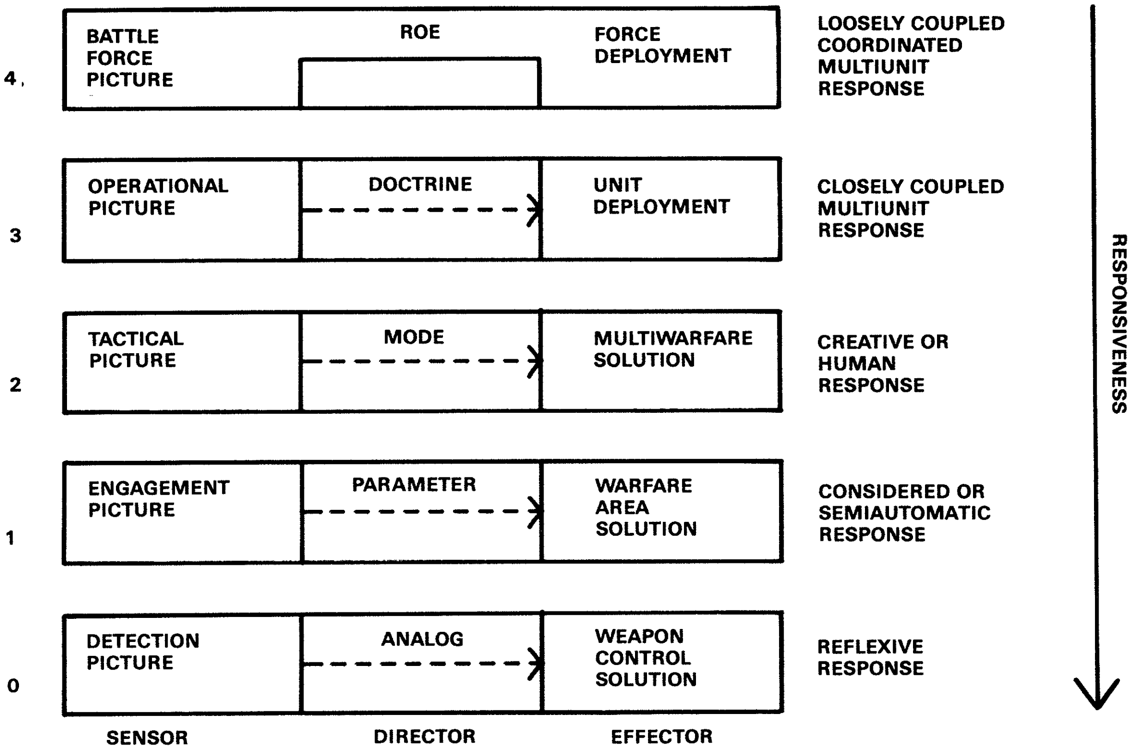
Introduction

Navy C^3 systems have at their core a requirement for a significant software methodology for managing a massive command, control, communications, and intelligence (C^3I) system encompassing land, sea, and airborne elements. Driving such systems are significant real-time requirements for tracking thousands of objects, discriminating the real threats among them, and tracking them by using an intelligent analysis of which objects are decoys and which are threats. The analysis is necessarily distributed and requires substantial data that must be consistent, always available, and accessible.

To succeed, a thorough and consistent logical data model must be used for all dispersed components of the Navy's C^3 system (see Figure 1 and D. Small's paper on "Machine Based Information Systems for Navy C^2" [1]). Real-time scheduling is the key to meeting the requirement for accessing all data described in a timely manner. For such scheduling, "process completion time is crucial to the correctness of application software [2]." (In our example, "process completion time is the time required for a target to be correlated.") The scheduling must be tempered by the need for keeping available and consistent copies of the various track data elements with which the C^3 system must cope.

Some of the more difficult issues that must be addressed to construct such a system are described below: (1) how to share data among parallel command control system processes such that the view of shared data is consistent among all processes; (2) how to attain improvements in performance by using a distribution of computer hardware (in the context of this paper, all assessments for performance improvement are made by using relative numerical data); and (3) how to accomplish consistent backup and recovery of distributed data in case of processing failure in a "real-time" environment. The last issue is addressed only in the context of assuring the availability of consistent data no matter where they are located.

Possible approaches for achieving real-time scheduling of data management for Navy C^3 systems are described next. Following that, an experimental methodology for assessing the best approach is outlined.



Responsiveness: Process completion time* critical to correctness at Reflexive Response Level

***In our example this is time for target to be correlated**

Figure 1. New Perspective: Levels of Conceptual Model for Navy C^2

Possible Approaches for Attaining Real-Time Data Base Management

(a) Use a centralized global data base to maintain all updates and access of data. Survivability and reliability issues immediately force this alternative into one of distributed data management to maintain "copies" of the data. This approach also does not manage "late" arrivals of similar data from differing sources. These "copies" somehow are maintained until they can be merged correctly into the global data base — again a distributed data base problem.

(b) The implementation of a real-time distributed data base system (DDBMS) to maintain massive amounts of dispersed command control data in a survivable and reliable manner forces the issue of efficient concurrency control over all data copies. Conventional models for DDBMS systems, such as Computer Corporation of America SDD1 [3] and TANDEM's transaction processing architecture [4], have been utilized in the past to assure that no data element is used until all its copies are correct. Loss of performance can be the price paid to ensure this level of data base consistency. A possible solution to this problem has been to use the first version of the data available and let other versions be copied while the first version is in use. The complexity of this solution multiplies when large numbers of data elements are present and a new version of a data element arrives while it is either still in use or its previous copies are being completed.

(c) It is our belief that concurrency can be managed efficiently and in a modular fashion by using a precedence of operations such that command control data can be consistently updated. The issue is how much data must be synchronized (sometimes done by locking the data from use) to keep the data consistent, and when such locking is required. An example of such data synchronization (published in L. Wong's paper on "Distributed Data-Base Management for Combat Systems" [5]) considers the alternatives of synchronizing copies of an entire relation versus defining each tuple as the element to be synchronized. An example of the latter may be to synchronize only significant updates (i.e., a change in threat status of a C^2 data element), or otherwise to do a projection of data element values as appropriate. Research at NOSC currently is being done to determine the best elements to be synchronized and when they should be synchronized.

The remainder of this paper describes an investigation into these alternatives. An experimental plan for methodical exploration of the issues involved in attaining real-time data base management is developed. Next, the results gained to date are described. Follow-on experiments suggested by these results are outlined in the last section of this paper.

Experimental Plan for Investigation of Real-Time Data Base Management Issues

A first version of a software model for the correlation of track data from dissimilar sensor sources, using a relational DBMS for data access, has been completed. Simulated target reports from radar, remote, and ESM sensors are used to generate an application process for initiating and/or updating global tracks from those local tracks. The end result is the creation of a consistent and, we hope, unique global track from any local report. The measure of correctness will be the time required for the target to be merged into a global track; i.e., will the target be merged before other versions of the target's data enter the system?

Using this baseline model, a distributed processing testbed of six SUN microcomputers was instrumented to evaluate different methods of distribution of the model's data and processes. A number of statistics were gathered to determine total processing time for each report, including correlation processing time, the queue time spent waiting for data base access, and data base processing time. The total of these three equals the total processing time per report. To analyze various ways of ensuring that a report does not get lost because of processing delays, a number of statistics also were gathered on processor and interconnect (Ethernet) utilization. Several configurations are described next, with analytical results and conclusions presented. An outline of further tests, which will attempt to show the best ways of accomplishing real-time data base distribution by using complete locking of relation copies and partial locking, will be described in the closing paragraphs.

Figure 2 shows a centralized configuration in which scenario generation, correlation processing, and data base processing are all on one SUN processor. The results with this configuration are not unexpected, and are tabulated in Table 1. The data base system is the most heavily used of all processing methods. The remainder of this report will focus first on the distribution of simulated sensor processing to establish our measurement methodology. Based on the results of that distribution, methods for improvement of data base processing by means of data distribution will be discussed. All numerical data presented are relative and based on the same scenario for all configurations. It is assumed that the numbers can be improved by using processors with better performance.

Table 1
RESULTS: CENTRALIZED MODULARITY EXPERIMENT PERFORMANCE STATISTICS

AVERAGE PROCESSING TIME TO CREATE A GLOBAL TRACK/REPORT

TCP	TDQT	TDPT	TPT
0.255	1.322	3.787	5.364

PERCENTAGE OF TPT/REPORT

4.8%	24.6%	70.6%	100.0%

CENTRALIZED CPU USE
(Clock Time Elapsed = 2,207)

	CPU TIME USED	UTILIZATION OF CPU
Scenario Generator	8.3	0.4%
Correlation Processing	334.9	15.2%
Data base	1,447.3	65.6%

(All Times in seconds)

Legend:
TCP = Total Correlation Processing Time/Report
TDQT = Total Database Queue Time/Report
TDPT = Total Database Processing Time/Report
TPT = Total Processing Time/Report

Figure 3, Configuration I, distributes the simulated sensor data such that each SUN processor is dedicated to processing data from only one data source. Figure 4, Configuration II, shows the same configuration except that the data base processing is not done on the same processor as that on which the data are located. The cost of doing that is about 40% more in average time to correlate a target. This cost is attributable mostly to the interval required to prepare messages for sending back and forth over the Ethernet. The data substantiating this are tabulated in Table 2.

Figure 5 is a diagram of the best of the two previous configurations,. Configuration I, using round-robin scheduling to determine which report is processed on which SUN. In this instance, the load distribution on the SUNs is more evenly dispersed because the next SUN takes the next report regardless of report type. Table 3 shows that comparison between distribution by data source and round-robin — in terms of report processing time and processor utilization. Correlation processing is evenly distributed among the three processors dedicated to the job; but there is no significant difference in processing time.

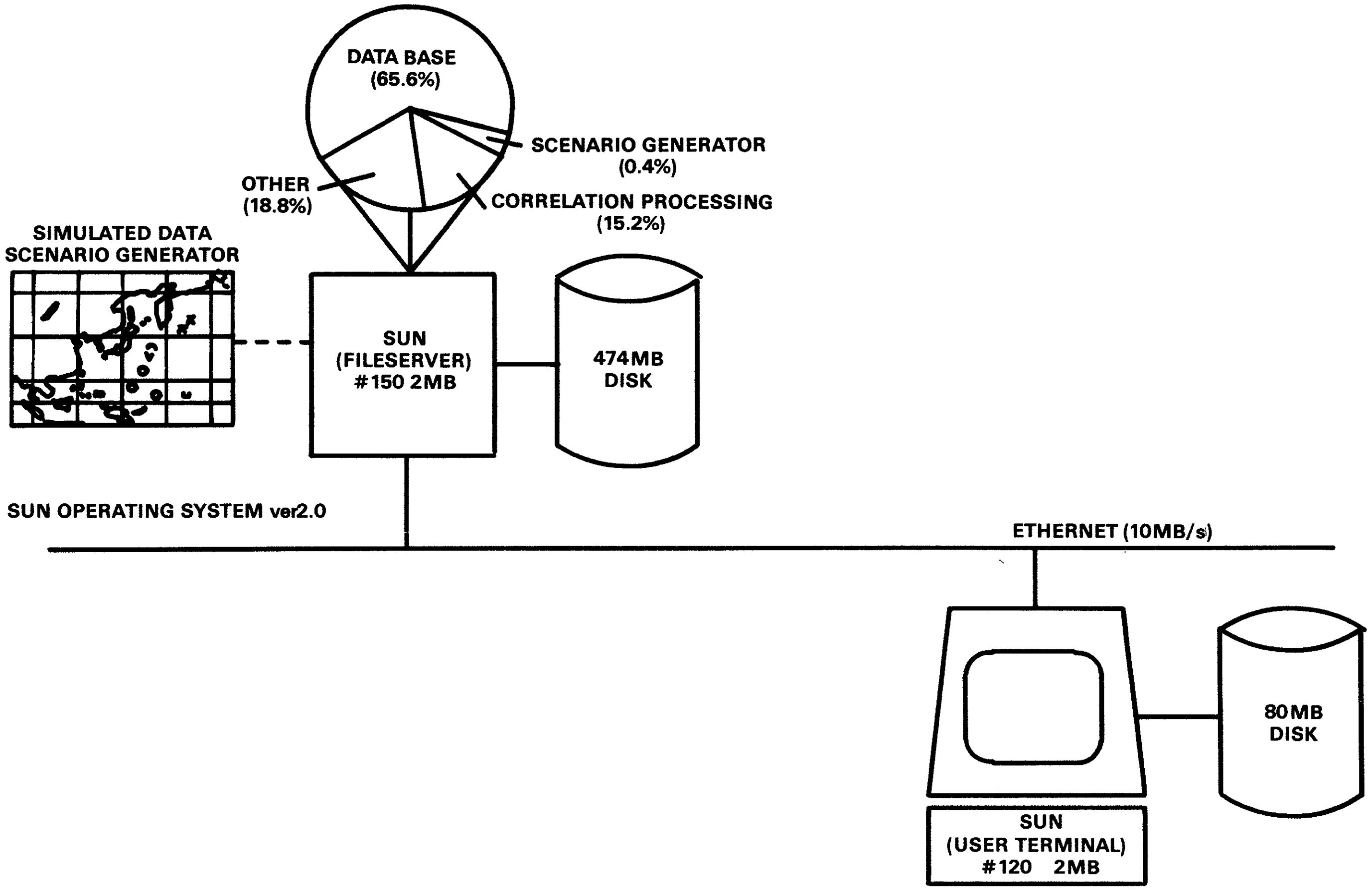


Figure 2. Centralized Modularity Experiment Configuration

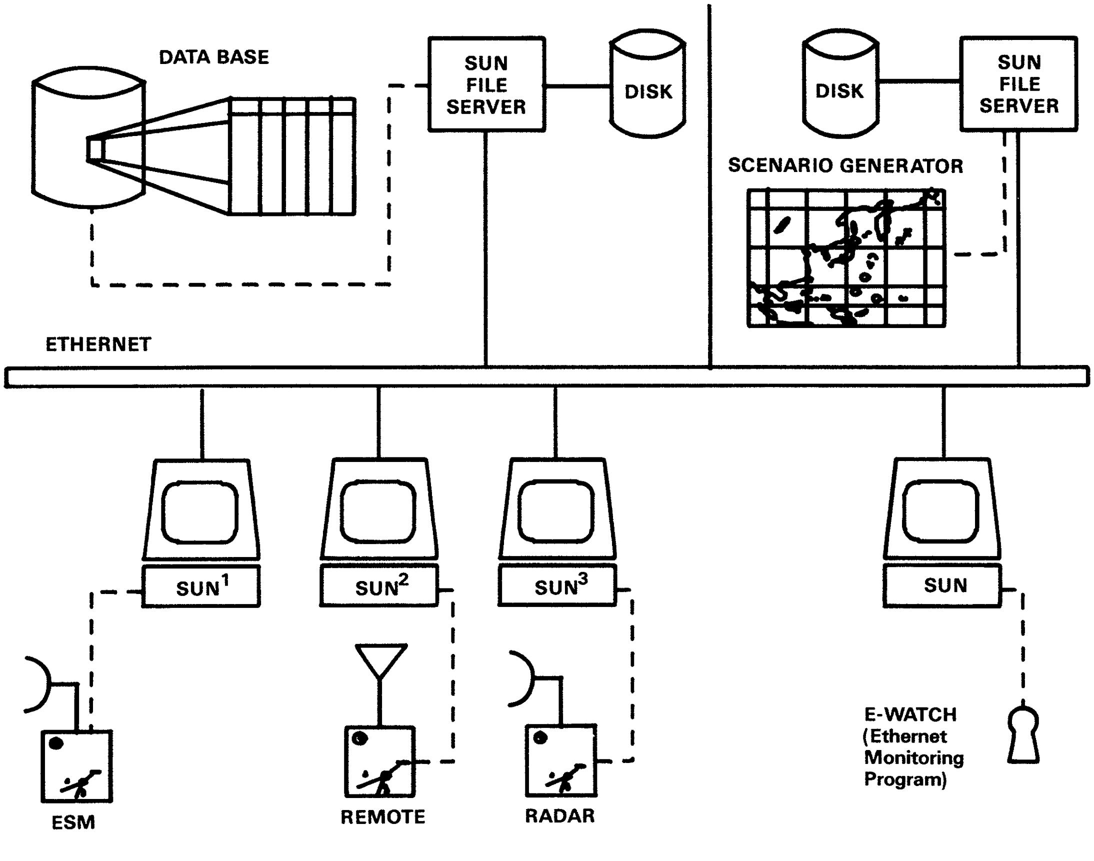


Figure 3. Distribution by (Simulated) Data Source Configuration I

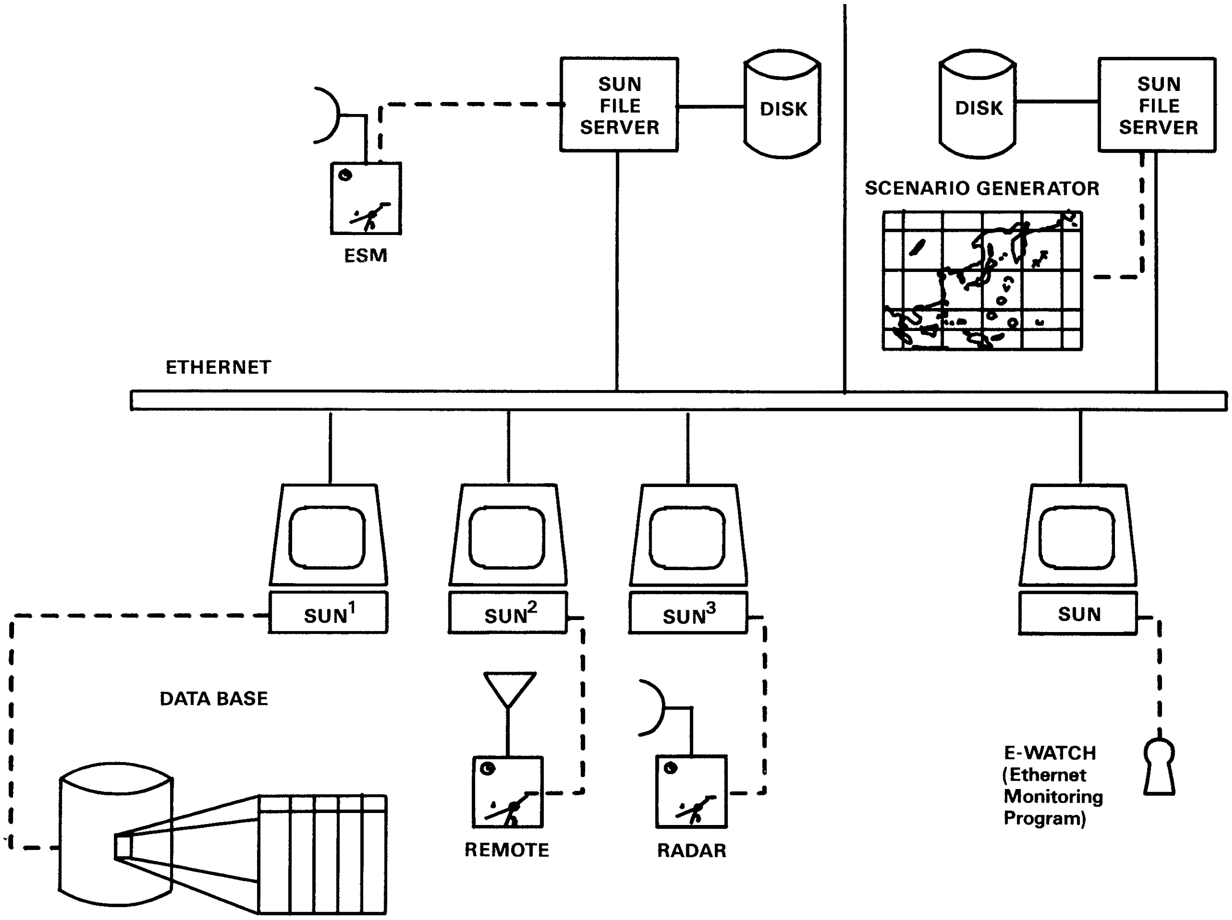


Figure 4. Distribution by (Simulated) Data Source Configuration II

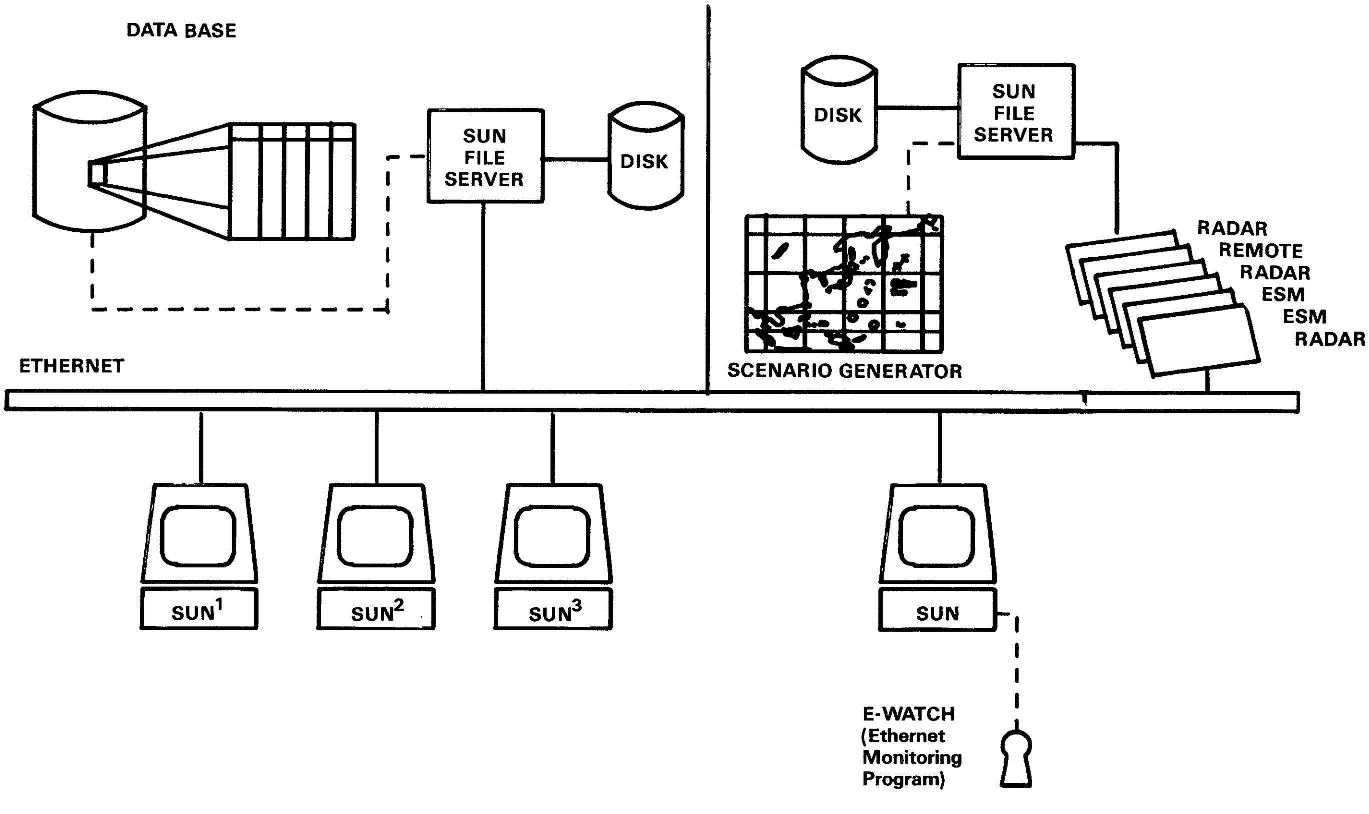


Figure 5. Round-Robin Configuration I (Simulated Data Inputs)

Table 2

DISTRIBUTION BY DATA SOURCE CONFIGURATION I

	TCP, s/rpt		TDQT, s/rpt		TDPT, s/rpt		TPT, s/rpt
SUN1 (ESM)	0.283	+	5.816	+	5.359	=	11.458
SUN2 (Radar)	0.339	+	9.599	+	5.021	=	14.959
SUN3 (Remote)	0.258	+	10.042	+	3.796	=	14.096

Total Time for All Reports = 2321.993 s
Ethernet Utilization = 0.95%

DISTRIBUTION BY DATA SOURCE CONFIGURATION II

	TCP, s/rpt		TDQT, s/rpt		TDPT, s/rpt		TPT, s/rpt
SUN1 (ESM)	0.357	+	9.529	+	10.186	=	20.073
SUN2 (Radar)	0.308	+	12.195	+	8.291	=	20.794
SUN3 (Remote)	0.237	+	16.042	+	6.649	=	22.927

Total Time for All Reports = 3646.647 s
Ethernet Utilization = 1.05%

Legend:
TCP = Total Correlation Processing Time/Report
TDQT = Total Database Queue Time/Report
TDPT = Total Database Processing Time/Report
TPT = Total Processing Time/Report

Table 3

DISTRIBUTION BY DATA SOURCE CONFIGURATION I
(Utilization Data)

	TCP	TDQT	TDPT	TPT	CPU Time Used, s	Elapsed Time, s	Processor Utilization
Scenario Generator					10.6	2952	<1%
SUN1 (ESM)	0.283	5.816	5.359	11.458	168.0	2864	5%
SUN2 (Radar)	0.339	9.599	5.021	14.959	106.0	2826	3%
SUN3 (Remote)	0.258	10.042	3.796	14.096	162.5	2795	5%
Database					1967.8	2887	68%

ROUND ROBIN DISTRIBUTION CONFIGURATION I
(Utilization Data)

	TCP	TDQT	TDPT	TPT	CPU Time Used, s	Elapsed Time, s	Processor Utilization
Scenario Generator					11.0	2262	<1%
SUN1	0.261	5.264	4.004	9.888	115.3	2180	5%
SUN2	0.266	6.963	4.306	11.535	114.0	2137	5%
SUN3	0.241	10.569	3.985	14.794	124.2	2111	6%
Database					1563.0	2201	71%

Legend:
TCP = Total Correlation Processing Time/Report
TDQT = Total Database Queue Time/Report
TDPT = Total Database Processing Time/Report
TPT = Total Processing Time/Report

Follow-On Experiments

Based on the results of the previous experiments depicted in Figures 3 through 5 and Tables 2 and 3, a series of experiments is now being performed in which each SUN does its own data base processing. Figure 6 shows an example in which each SUN is dedicated to processing for only one type of sensor source. In this diagram, all shared data are locked when updated. Other possibilities include allowing the use of older versions of data while the update is occuring [6] or sending copies of shared data when significant changes occur (as is being done in current NOSC research). These experiments also will be done by using the round-robin scheduling method. Measurement data will be collected for these experiments just as has been done for the experiments already described in this paper.

Once appropriate methods for distribution are established, this work will progress to investigation of recovery from failure and, secondly, investigation of better processors to keep from falling behind in target processing — i.e., to meet "real-time" criteria.

References

1. D. Small, "Machine-Based Information Systems for Navy C^2," Proceedings of Seventh MIT/ONR Workshop on C^3 Systems, December 1984.

2. E. D. Jensen, D. Locke, and H. Tokuda, "A Time-Driven Scheduling Model for Real-Time Operating Systems," Proceedings Real-Time Systems Symposium, December 1985.

3. J. B. Rothnie, etc., SDD-1: A System for Distributed Databases, Computer Corporation of America Technical Report CCA-02-79 (Revised), August 1979.

4. "Description of TANDEM 16 System," Combat System Experiments, Summary of Architectural Papers, Informal NOSC Working Document, September 1983.

5. L. Wong, "Distributed Database Management for Combat Systems," Proceedings of Seventh MIT/ONR Workshop on C^3 Systems, December 1984.

6. M. Singhai and A. K. Agrawala, An Algorithm for Update Synchronization in Replicated Database Systems, University of Maryland CS-TR-1518, July 1985.

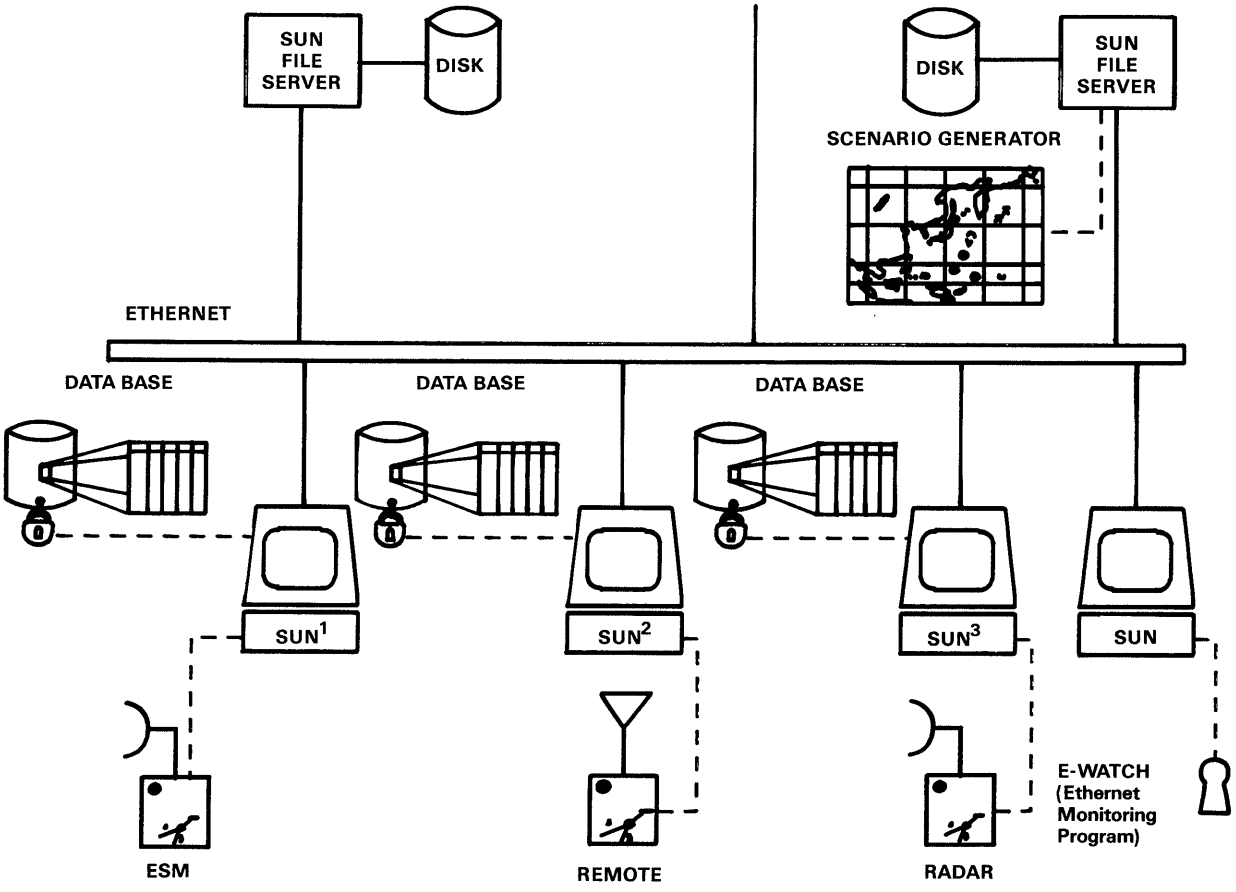


Figure 6. Distributed Data Base Processing (Simulated Data) (All Relation Copies Guaranteed Identical)

KNOWLEDGE BASED HIERARCHICAL UNDERSTANDING SYSTEM FOR CRITICAL COMBAT NODE ANALYSIS

F. Doug Deffenbaugh, John R. Miller, and J. H. Swaffield

TRW Defense Systems Group, Systems Engineering & Development Division
Redondo Beach, California 90278

SUMMARY

The input to critical combat node analysis is in the form of entity data extracted from opposing forces communications, non-communications emissions and other intelligence related events from the battlefield. The data is often incomplete, often unreliable, and changes with time. TRW's knowledge based data correlation (DACORR) system is a demonstration system for understanding hierarchical systems and is intended for use as a decision aid for critical combat node analysis. The system uses a blackboard knowledge representation framework, and was developed initially using Standford's AGE software design tool.

INTRODUCTION

The modern battlefield combat intelligence environment is flooded with information derived from many kinds of sensors and associated sensor processors. Such data is often untimely, ambiguous and duplicative. In recognition of this problem, nearly eight years ago DoD organized the initiative for tactical data fusion under the original program name: Battlefield Exploitation and Target Acquisition (BETA) as a joint service testbed, evolutionary development effort [1]. By the end of 1981, USA and USAF troop experimentations were commencing with new BETA testbeds at Ft. Hood, Texas and Hurlburt Field, Florida to evolve the system procedures and conceptual C^3 interoperability techniques to employ such innovative situation assessment and target nomination assets. Automated tactical data fusion techniques were not previously in the military inventory. Also, by this time DoD planning was underway to acquire and deploy to the European theater a "fixed plant" derived configuration of the BETA testbed together with the applicable threat data base and other P^3I attributes defined through the BETA troop experimentations at Fort Hood and Hurlburt Field. By 1982 BETA Derived Systems (Number One) known as the Limited Operational Capability - Europe (LOCE) System was operational at the Combat Operations Intelligence Center (COIC) at Ramstein Air Base, FRG. The LOCE system fielding within the EUCOM structure included NATO participation. It also encompassed field troop experimentation with the distributed remote workstations in such locations as 2ATF/NORTHAG at Rhiendahlen, FRG [2 and 4]. Finalization of DoD planning for acquisition of BETA Derived Systems (Number Two) was occurring by the end of 1983. This system, derived from the Hurlburt USAF testbed configuration, again updated with the appropriate threat data base and still further P^3I and evolving different sensor interfaces, was to be known as the Limited Enemy Situation and Correlation Element (LENSCE). LENSCE, fielded with the Ninth Tactical Air Force at Shaw AFB, S.C., is a mobile "fly away" configuration to provide the operational tactical data fusion support for the Air Force component of U. S. Central Command, the former Rapid Deployment Forces (RDF).

By late 1982 new state-of-the-art technology in the artificial intelligence expert on Knowledge Based System (KBS) domain waş maturing [3,4] to the point where a TRW IR&D aimed at providing expert ELINT weights and value thresholds for the numerous LOCE system parameters was accepted by the government through theater field trials for further LOCE system preplanned product improvement (P^3I). By 1984 other AI/KBS developments were occurring throughout industry. Examples were EXPRS - an Expert Data Fusion using PROLOG/POPLOG by Lockheed Research Laboratory [5], ANALYST by Mitre Corporation [6] and similar work in the Computer Science and Technology Division of SRI International [7], and TRICERO, [8] in another similar effort for multiple sensor signal understanding. By the end of 1984, after successful LOCE Joint Operational Effectiveness Evaluation (JOEE) by both AFOTEC and USA OTEA and commensurate with the mid-range P^3I planning for Derived Systems enhancements through the initial 1989 fielding of the long term service follow-on systems ASAS and ENSCE, DoD planned the extrapolation of the LOCE TUNER expert system concept to include not only ELINT but also added COMINT and other sensor discipline data.

Clearly, these systems and the continuing significant preplanned product improvements have blunted the data onslaught. However, we as a community have a long way yet to go. Let us not forget that clearly the worldwide threat growth in these ensuing years has not stood still either. Many dramatic advances have also been achieved over these same seven years in our own sensor collection and first instance" sensor processing technologies. This has encompassed not only sensor P^3I but also the introduction of new organic theater as well as national systems thus providing still larger sensor collection "chunks" for combat information exploitation in near real time. All this exploitation support is still expected with the same or less numbers of deployed battlefield analysts processing essentially the same skill capabilities. The bottom line is that a "technology multiplier" is required to gain the necessary, urgently needed, quantum leap forward in this crucial tactical intelligence data reduction and exploitation arena.

PROBLEM

It is the belief of these authors that there are some seven pacing technology issues which have plagued the cloistered tactical C3I/C3CM community for a decade now. These are:

1. Sensor Interface Automation and Cueing Feedback

2. Message Processing, Routing and Data Extraction Processing

3. Distributed Data Processing

4. Detection and Identification of Critical Combat Nodes/High Value Target Data

5. Soldier-Machine Data Assessment Processes

6. Timely Multi-Source Data Fusion Algorithm Enhancements

While significant enhancements have been achieved in six of these areas, and either have already or will soon result in new system/subsystem/P^{3}I equipment deployments to field, still two crucial remaining aspects have eluded the industry. These two critical (pacing) C^{3}I/C^{3}CM technologies are:

1. Detection and Identification of Critical Combat Nodes/High Value Target data on the Modern Battlefield in an expeditious manner, and

2. Timely Multi-Source Data Fusion Algorithm Enhancements to the "first generation" deployed systems such as the TRW BETA Derived Systems (LOCE and LENSCE).

It is the first of these two critical areas that this paper is focused upon. Slowly evolving national policy and direction relative to sanitized COMINT data exploitation and reporting will soon permit more definitive battlefield support through newer state-of-the-art expert approaches to the detection and recognition of various opposing forces (OPFOR) critical combat nodes or high value/high payoff data derived from near real time battlefield exploitation. While little can be found or said in the open sources, it is clear that clusters of intense research activities in this area are occurring through various DoD sponsorships at USAF RADC and the Electronic Security Command as well as by the C^{3}CM Joint Test Force, the U. S. Army Intelligence Center & School, the TRADOC Combined Arms Combat Development Activity, the National Security Agency, the Signal Warefare Center VHFS and the U. S. European Command together with its component and allied commands.

The critical combat node exploitation process may be thought of as consisting of two basic parts:

1. data reduction and order of battle (OB) preparation, and

2. situation assessment.

It must be recognized that OB production includes the detection and identification of battlefield entities and complexes which represent order of battle. The information processing of this data in context with the curent and/or projected battlefield situation(s) yields the isolation of battlefield combat nodes. Based on the friendly forces command battle plan objectives and Essential Elements of Information and Requests for Intelligence Information (EEI/RII), high value target data, frequently referred to as critical combat nodes, can be identified.

The input to combat node analysis is the form of entity data extracted from collected intelligence reports derived from OPFOR communications, non-communications emmisions and other intelligence related events from the battlefield. Node analysis relies heavily of attribute knowledge; knowledged concerned with the characteristics and composition of enemy equipment, units, etc. It involves the repeated processes of: self-correlation, cross-correlation, aggregation and component collection.

Self-correlation describes the process of deciding whether or not two reports describe the same entity across time and space. Cross-correlation assumes an existing entity to be a subordinate node and seeks an association to an existing superior node. Aggregation is the process of hypothesizing or creating new nodes from ensembles of existing entities. Component collection acknowledges the existence of a "superior" node (in a hierarchical structure) and tries to make an association to its subordinates which are loose in the data base. In today's systems these "correlation" functions are semi-automated and involve a process referred to as "templating". These "electronic templates" are file structures or tables which are descriptions of the OPFOR emmiter type(s), numbers and relative locations of all the components of a superior entity as well as some likelihood estimators. An association is made between superior and subordinate elements when a numerically valid association is found by comparing entity data and template data on types, numbers, locations, and likelihood estimators. The data upon which these computations are accomplished are basically intelligence reports which may exhibit:

1. varying degrees of position accuracy and object classification due to varying sensor capabilities and uncooperative collection conditions that always arise in tactical situations.

2. non-continuous flow of input data, due to problems in collection management and coverage.

3. arrival of data out of proper time order, due to receiving dissemination and problems that also always arise in tactical situations as well as use of multiple, nearly autonomous operating sensor systems.

4. lack of one or more types of data for extended periods of time as a result of sensor operation problems, as well as loss of sensor due to combat.

Experimentation at TRW with real-world field data indicates that current traditional algorithmic templating methods are inadequate because of the unreliability of the data itself. Artificial Intelligence (AI) techniques, however, are most appropriate when the data for solving the problems are incomplete, unreliable, or changing with time, when the knowledge about the domain is uncertain, and when the search of solutions is very large.

APPROACH

The objective of this TRW IR&D project is to develop a prototype expert or knowledge based system (KBS) to aid the combat analyst with order of battle preparation. Our approach to developing a decision aid for data correlation was to build a knowledge based hierarchical understanding system using a blackboard knowledge representation scheme. The blackboard architecture was originally developed as a problem solving framework for the Hearsay-II speech understanding system. The issues addressed by Hearsay-II which are applicable to this project include the restraint of search in a large solution space (the focus of attention problem), uncertainty in the data operated on, multiple, diverse, and knowledge sources which also are subject to error applied to that data, and the resolution of uncertainty between numerous potential solutions.

RESULTS

The initial data correlation (DACORR) demonstration system was implemented on a Xerox 1108/105T computer using Stanford's AGE [9] knowledge engineering software design tool. The original architecture was then tailored to the combat node analysis problem by modifying and adding processes coded in LISP to represent the architecture in Figure 1.

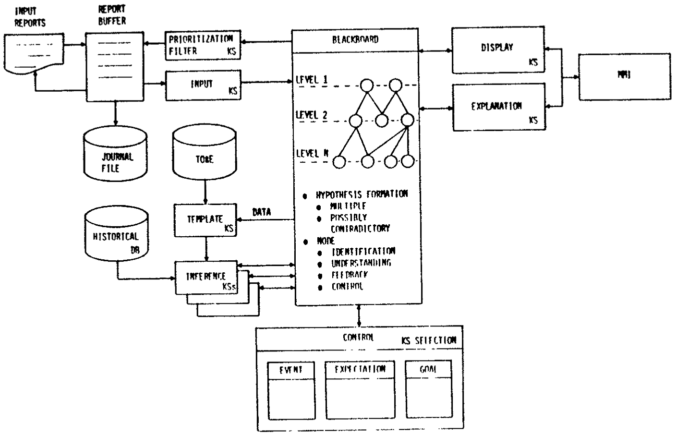


Figure 1. Architecture:
Knowledge Based System for Tactical Data Correlation.

Basically, processing in the DACORR system consists of additions, alterations, and deletions made to the data on the blackboard by the various knowledge sources. The current "state" of the blackboard is usually referred to as the current hypothesis. Knowledge sources (KSs) are production rules all organized around some object or concept. A KS can add nodes with support from lower level nodes, (data driven aggregation) or may hypothesize nodes at lower levels. Multiple, even contradictory, hypotheses will be carried along until enough confidence in new (updated) hypotheses is gained to resolve conflicts or the analyst intervens. The creations or modification of hypothesis elements (nodes or links) will always trigger one of more validation KS's. Their purpose is to examine the validity and to assign and update the certainty factors associated with the various links.

The control KS is responsible for selection and invocation of the KS's, and the selection of items on the blackboard for focus of attention. The control KS determines which type of step to take (an event, an expectation, or a goal), selects a particular step to process from the step-type list (evenlist, goalist, expectation list), and invokes a KS relevant to the selected step. When a KS is invoked, the rule(s) within that KS are executed and, if successful, perform one (or more) actions:

1. proposes changes in the hypothesis

2. indicates that some change in the hypothesis is expected

3. indicates that the KS wants a particular state in the hypothesis achieved.

Sensor reports represent observations or bodies of evidence about the solution space or ground truth. Doctrinal templates serve as the basis for generic types of entities that potentially exist. The hypothesis of the solutions space, i.e., the current state of the blackboard, represents sensor reports expressing partial beleifs over the possibilities in the templates relative to the current environment. The templates and the blackboard have the same hierarchy of levels. However, in the blackboard these levels contain bodies of evidence not generic types and structures.

The DACORR system currently assigns validity weights based on a combination of traditional electronic templating (BETA-like) methods and rule based methods. It is clear, however, that the domain size and complexity of the OB preparation problem requires a hypothesis generator with a mathematical framework such as the Dempster-Shafer theory of evidential reasoning [10]. The Dempster-Shafer approach provides a mathematical basis for combining the bodies of evidence that the sensor reports represent and the "knowledge" in the templates. The combined knowledge represents a deduction or belief of the output space [11].

In the DACORR system the combat analyst's knowledge encoded in the form of production rules is used to:

1. refine the templates to reflect the current situation, and

2. to validate, filter, and prune hypothesis which have been generated by algorithmic electronic templating or Dempster-Shafer like methods.

Processing by the current DACORR system begins by selecting a sensor report from the report buffer according to the objectives of the user. The prioritization KS monitors user goals, the current focus, and patterns on the blackboard to determine the priority and order in which reports are processed. Reports not selected for immediate attention may be routed to re-enter the report buffer at a later time or may be written to a journal file. Selected reports are processed by the input KS and are posted to the appropriate blackboard level which ranges from Theatre of Military Operations (TMO) to battalion echelons and an "unknown entity" level. Control then usually passes to the self-correlation KS which combines bodies of evidence about the same entity. Reports believed to be new nodes and the order of battle holdings on the blackboard are compared to the doctrinal templates and associations are formed with other nodes on the blackboard. However, because the doctrinal templates are not rigid stuctures but serve as incomplete axiomatic models of the solution space the templates may be "refined" before an association weight is produced. When the template KS is invoked by the control system, appropriate "static" templates from the TO&E background database are selected and modified depending on the data or situation on the blackboard. In this scenario, a motorized rifle regiment may be more likely to contain five subordinate motorized rifle battalions rather than the usual three if the regiment's location is near the FLOT.

After "templating" other KS's may be invoked by the control system to validate hypotheses which have been generated. Inferences may be made of unknown units echlon or type, and multiple hypothesis formed. The display and explanation provide the capability to review the hypothesis structure on the blackboard, examine the inference history and the corresponding rules.

EXAMPLE

As an example of how the DACORR KBS will perform as a decision aid for combat analysts in determining the opposing force order of battle 25 TACREP reports have been processed from a scenario of division level field training exercise (FTX). The purpose of this example is to elucidate the discussion of the previous section and demonstrate the system architecture. The amount of knowledge currently contained in the system is still sparse and is the subject of the current year's effort.

Elements of the 93rd GMRD, garrison location Brunne, have begun moving into the Lehnin PRA suggesting that this division plans to conduct a division level FTX. Figure 2 shows a copy of the display screen after processing 16 initial TACREP reports.

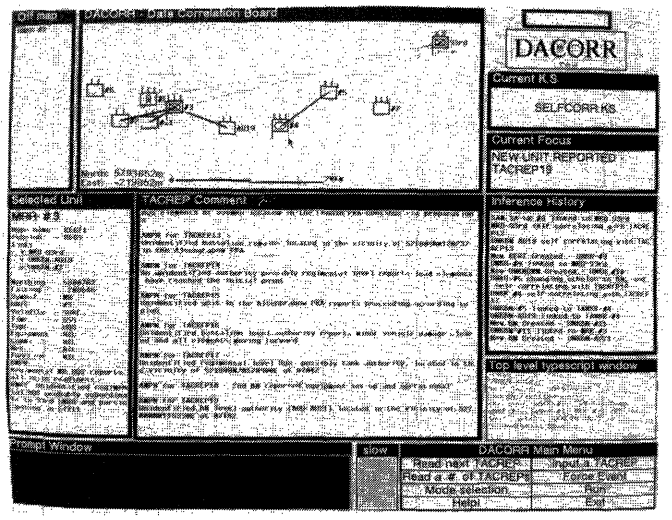


Figure 2. KBS System Display Features

The amplification lines of the TACREP's are reproduced at the center of the display. The "DACORR - Data Correlation Board" window shows a display of the intelligence reports which have been processed by the system and "posted" to the blackboard. The windows at the top right of the screen show the current KS which is active.

The "Current KS" box in Figure 2 indicates that production rules in the self-correlation knowledge source are processing TACREP 19 space (NEW,UNIT.REPORTED-TACREP19) to decide if Report 19 is about a new entity or one that already exists. The "Inference history" box in Figure 2 shows the reasoning which has previously taken place; i.e., new units have been created, nodes have been updated and information merged, and associations between entities have been hypothesized. The current best "Order of Battle" hypothesis is indicated by the lines linking the various elements shown in the "DACORR-Data Correlation Board" box.

The user may request information about any item on the blackboard by placing the cursor arrow on an icon using the "mouse" pointing device and pressing a button. In this example the operator has selected MMR-#3. The information displayed in the "Selection Item" window shows that a template match has occurred and that MMR-#3 is linked to two battalions unknown type as well as the 93rd GMRD. Note that the battalion designated AU19 is shown belonging to MMR-#3 even though it appears closer to tank regiment TANKR-#4. In actuality the system also maintains a link from AU19 to TANKR-#4, however, the "best guess" according to information stored in the templates is that AU19 belongs to MRR-#3. If later information were to establish that UNKBN-#2 (AU19) was actually a tank battalion, the system would compute a new hypothesis and the "best guess" link would be to TANKR #4. Even then, however, a multiple hypothesis would still be maintained with a "weaker" link to MRR-#3 by virtue of the fact that a tank battalion is still an organizational element of a motorized rifle regiment.

Multiple hypothesis structures may be shown at a later time with the view zoomed in on elements to the left of the screen in Figure 2. Units which are now off the display are shown in the "Off Map" window. The user can have the correlation board automatically redisplayed including any off map item by simply mousing the desired unit. Multiple link values may be computed. These are brought to the screen by mousing the "Links" item in the "Moused Item" window.

CONCLUSIONS

The DACORR system demonstrates an artificial intelligence approach to developing a system to aid the combat analyst with order of battle preparation. Efforts this year are focused on expanding and developing the baseline blackboard system. COMINT and ELINT knowledge sources have been expanded and the system will be tested using multi-sensor SIGINT data from a credible controled European scenario. Capabilities for maintaining multiple, possibly competing, hypotheses for automated aggregation, cross-correlation and component collection will be tested and evaluated for use as a combat analyst decision aid for identifying and exploiting critical combat nodes.

BIBLIOGRAPHY

1. BETA - An Idea Whose Time Had Come, Signal Magazine, pgs 11-13, October 1981.

2. Fusion Centers, Signal Magazine, pgs, 131-133, May 1984

3. Artificial Intelligence Applied to C^3I, Signal Magazine, pgs 27-33, March 1983.

4. Artificial Intelligence Comes of Age, National Defense Magazine, pgs 43-46, 50-52, December 1984.

5. EXPRS - A Prototype EXPERT System Using PROLOG for Data Fusion, AI Magazine, pgs 37-41, Summer 1984.

6. ANALYST, An EXPERT System for Processing Sensor Returns, R. P. Bonasso, Jr., Mitre Corporation Report MTP-83W00002, February 1984.

7. An AI Approach to Information Fusion, Journal of Electronic Defense, pgs 31-32, 34,36, 38 and 41, July 1984.

8. The Challenge of Technology for Military Intelligence Applications, Journal of Electronic Defense, pgs 29-30,32,34, 38-40, 42 and 50, January 1984.

9. AGE Reference Manual, AGE-1, Heuristic Programming Project Report HPP 81-24, Stanford University, October 1981.

10. A Mathematical Theory of Evidence, G. Shafer, Princeton University Press, Princeton, New Jersey, 1976.

11. A Set-Theoretic Framework for the Processing of Uncertain Knowledge, S. Y. Lu and H. E. Stephenou, Proc. National Conference on Artificial Intelligence, Aug. 6-10, 1984, 216-221.

EXPERT SYSTEM TECHNIQUES FOR RECONSTRUCTION AND POST ANALYSES

Robin A. Dillard
Naval Ocean Systems Center
San Diego, CA 92152-5000

ABSTRACT

A reconstruction and post-analyses system is outlined, and a representation scheme presented that will enable what-if reasoning in post analyses. The what-if situations typically involve postulating a different action than that which was taken. Since the action in question frequently will be a communication, the refinement, by post analyses, of rules for exchanging data among sites is discussed.

INTRODUCTION

A system for reconstructing and analyzing the flow of events of naval exercises and operations should be developed in conjunction with other C^2 subsystems. Human documentation is a slow and tedious process, and the result is not easily subject to query. Good records are needed for evaluating and improving fleet performance and for determining probable enemy reaction and probable outcomes. Artificial intelligence (AI) techniques for data fusion can be extended to perform the reconstruction. Other AI techniques can be developed to analyze the reconstructed data, provide useful interpretations, and improve the decision processes.

References 1 and 2 discuss the reconstruction and post-analyses processes. The reconstruction process consists of data fusion after all data are in. All reports would be organized and processed before inferencing, as opposed to being processed immediately after receipt of each report or batch of reports, as in data fusion. In general, we will interpret the reconstruction processes to be those processes requiring the same reasoning as data-fusion processes, the difference being the completeness of the information on which they operate.

Automated techniques should use the reconstruction to assist in interpreting, evaluating, forecasting, and modeling. Evaluations can be made of the data-fusion and planning processes and of the control strategies for communications, sensors, and weapons. What-if reasoning in post analyses can be useful in refining these processes and strategies.

THE PROBLEM

EMCON Example

An example of post analyses is that of evaluating the use of EMCON (EMission CONtrol) by a friendly platform__x. (Assume a single EMCON state: Emitters are off during EMCON and are on, otherwise.) One situation is where platform__x used EMCON (as ordered by a decision maker following doctrine or machine recommendation), and the analyses are to determine if this decision was the best one, based on the results. The other situation is where EMCON was not used by platform__x. The following are examples of questions to answer. Further analyses would measure the impact of each answer and involve tradeoff reasoning. The uniqueness of the emitter would enter into full analyses.

Platform__x EMCON-on questions:

- Did platform__x escape detection because of EMCON?
- Was platform__x detected in spite of EMCON?
- Was it because EMCON was initiated too late?
- What hostiles were not detected or located that could have been had platform__x's sensors been activated?
- Did EMCON initiation tell the enemy their presence was known?
- Could platform__x have targeted a hostile that no friend could have?
- Would platform__x likely have been the target of a radar-seeking missile if its radar were active?

Platform__x no-EMCON questions:

- If detected, could platform__x have escaped it with EMCON?
- Were platform__x emissions used to recognize it?
- Did platform__x detect hostiles actively which it would not have detected passively or would not have learned about from friends?
- Was platform__x able to target a hostile platform that no other friendly could have?
- Was platform__x the target of a radar-seeking missile?

Time-Late Information

The data-fusion system will receive much of its information well after the original time of observation. Some will be information known or available to humans at the time but not translated into machine language or not manually entered. Much of the information will be received from another site well after the observation time, i.e., time-late messages. The delay might be due to EMCON, higher priorities, propagation, or human error. Some delay occurs with air and space photographs and radar pictures; this may be processing time or the time until the aircraft returns or satellite downlink is possible. Occasionally there will be later sightings from which an earlier event can be inferred. This might be, for example, information from which an earlier contact is identified, observed damage from an earlier attack, or a port of destination.

Some of the late information (e.g., reports of major damage to an opponent) will still be of immediate value to the data-fusion system. Some will be of use only in post analyses. With a few exceptions, time-late sighting reports where newer position data are also available will not be useful when received. However, the lateness of the report does not decrease its value in reconstruction and post analyses.

To some extent, reconstruction and post analyses can be a continuing process during the exercise or operation, as time-late information is processed to improve the understanding of the situation several hours earlier. It may be many years, though, before computer resources are sufficient even for the situation assessment needs of the instant, so ongoing post analyses is only a thought for the future.

The delay in receiving sighting reports especially affects the usefulness of the position information at the time of receipt. For example, the position reported might be represented as a small ellipse, but unless the contact is known to be a merchant or its track merchant-like, the accuracy of the information when received would be represented with a greatly expanded ellipse, the axes having grown in proportion to the time delay and estimated speed. The accuracy of positions enters into determining targeting possibilities, both for hostiles targeting friends and for friends targeting hostiles, and both for determining if targeting has probably occurred and for determining if targeting could have occurred under what-if conditions.

Geographical Overlap

In the simplest case, the reconstruction process would be concerned with the events affecting a single ship and would use its data base, only. Event reports received by the ship would be used to the extent that they overlap geographically at the time. For example, all contacts close enough to detect emissions from the ship and all weapon-carrying contacts within weapons range would be of concern.

More generally, the reconstruction and post analyses will concern many platforms. Three cases of most interest are where integration is with respect to

- a U.S. battlegroup;
- a fixed geographical area;
- a hostile battlegroup.

In the battlegroup case, all contacts within detection/weapon range of any of the respective units would be included. In all cases, some out-of-area observations (e.g., ports of departure and destination, and damage observed after an attack) would also often be of use. Damage could be indicated by clues such as a repair ship visit, a platform under tow or under partial power for a long period, and the absence of emissions of certain kinds.

Evaluating Decisions

When a decision aid recommends a decision, the human may accept it or reject it. He may consciously agree with the machine recommendation or he may neglect to override it, but these two probably cannot be distinguished in reconstruction. We will assume that the situations where the human takes an action in conflict with the machine recommendation are recorded. The simplest situation, of course, is the case of only two possible decisions. More typically, the decision will be one of a number of possibilities, and there will be "distances" between the possible decisions. For example, a decision that the contact is a friend is distant from the decision that contact is a hostile combatant, while the decision that is a hostile auxiliary is generally close to the decision that a contact is a hostile national.

While a plan should be evaluated in the light of its results, another consideration is: did a bad plan succeed or a good plan fail only because a highly unlikely event occurred? The question of which is the best decision has several interpretations. It can be (1) the best decision based on evidence possessed at that time, (2) the best based on evidence obtainable at that time, (3) the best based on the reconstructed picture or actual situation at that time, or (4) the decision that would have produced the best outcome considering chance events.

Since it is essentially impossible to determine the "best" decision of each kind for all decision opportunities and to analyze performance in its full complexity, we need to reduce the problem to a manageable one. Primary attention should be given to important decision problems where a nontrivial amount of pertinent information was available at the time and where the machine recommendation and human decision differed or were both incorrect (based on the situation at the time, according to an expert analyst after reconstruction).

If they differed, some questions to be considered:

- Which was correct, based on the information in the data base at that time?
- Did the human and the machine base their decisions on the same facts?
- Did one use facts unavailable to or disregarded by the other?

If the machine decision based on partial evidence was incorrect based on full evidence, was there information obtainable from some external source which would have led to the correct decision. If so, what would have been the cost or risk to that site? Could crucial evidence have been obtained with different sensor-allocation or EMCON doctrine?

If the machine decision was incorrect based on the partial evidence, can any of the rules that fired in the process of reaching the decision be improved? Or, was there pertinent evidence not used, which could have been used with the addition of another rule?

SYSTEM OVERVIEW

Basic Representation

The following are the major kinds of frames in the data base. A "postulated event" and "what__if" can occur only in the post-analyses data base, while other frames can occur also in the data-fusion data base.

Physical object. Examples of physical objects are platform, sensor, emitter, and weapon.

Event. Situations and states will also be treated as events. Examples of events are track, sensor__off, communication (of specific information), detection, and attack. Two events used in geometry computations are track__segment and track__segment__pair. There are three types of events: actual, virtual, and postulated.

- Actual event: An event known to have occurred or possibly to have occurred, based on a report or inference. If later it is determined the event did not occur, this could be indicated, for example, by revising the confidence value or by tagging it as false.

- Virtual event: An event which, at the time reported, is predicted or planned. It remains a virtual event even after the predicted/planned time of the event. A similar actual event results if the prediction/plan is fulfilled.

- Postulated event: An event that did not occur; it is postulated after reconstruction for the purpose of "what__if" post analyses.

What__if. In a what__if frame, events that did not occur are postulated, and/or actual events are assumed not to have occurred — they are "negated." Inferencing with the what__if assumption can lead to the creation of other postulated events and can negate (conditional on the what__if) some actual events. (Physical objects could also be postulated, but this capability would not likely be used in post analyses applications.)

System Organization

Figure 1 shows the key pieces of the reconstruction and post analyses system for a multisite application. Figure 2 shows the different kinds of data in a site data base at the end of the operation or exercise. In the simplest case, the site data-fusion systems will be essentially alike — they would use identical rule evaluators, frame representations, and static data, and have most rules in common. Each site should archive the modifications to its data frames, and by using this file of modifications it should be possible, although difficult, to reconstruct a snapshot of the site's active data base at any given instant during the exercise/opreation. archiving snapshots of the active data base at key decision points would probably be more practical than constructing them, even though the record of modifications would stil be needed.

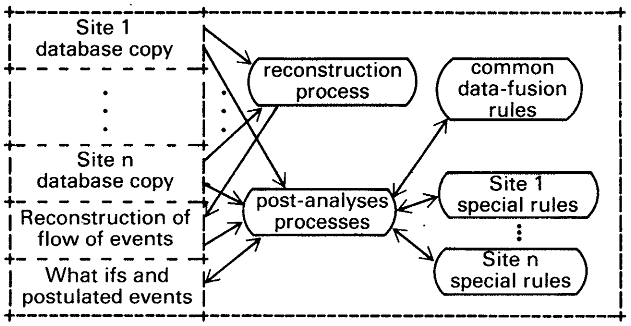


Figure 1. The reconstruction and post-analyses system.

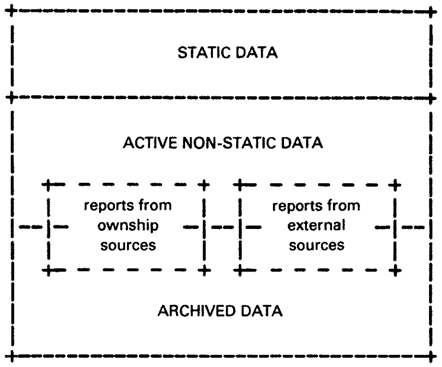


Figure 2. Copy of a site's data base at the end of an exercise or operation.

We will assume that the reconstruction process operates on the combined set of reports (local and external) received by the data-fusion systems at the n sites, and does not use the individual system inferences. (However, certain post-analyses process will use the inferred data when, for example, evaluating the inferencing capability.) The reconstruction of tracks probably will take place platform-by-platform rather than be time-ordered, but the order should not affect the result. Many of the reports will be duplicates; e.g., a report originating from one site and communicated to another. The reports from external sources are used to create "communication" axssertions in the reconstruction data base — records of what was communicated from site-to-site.

Although not shown in figure 1, the what-if capability should extend to postulating different rules, both inference rules and procedural rules (e.g., for exchanging data among sites). Most of the what-ifs, though, will postulate having taken a different action. After a what-if action or situation is postulated, the post-analyses system should determine the probable outcome based on the postulated situation, or the possible outcomes with their respective likelihoods. This process would use deterministic reasoning where possible —much the same reasoning as used in data fusion and planning. For example, if the enemy's active radar would be in detection range of one of our ships given the postulated situation, it could be concluded that the radar would have detected that ship. In addition to simple reasoning, simulation often will be necessary, and many simulations may be needed for statistically reliable results. Also, comparison procedures and measures of performance will be needed to determine which outcome is preferred. Human assistance in this process probably will be needed for a long time, and gradually the human will build his knowledge into the system and automate the process.

Data Organization

To illustrate the organization of the data in the data base after reconstruction, we give examples of typical kinds of data frames. Some of the examples are frames postulated during post analyses. The syntax of the frames would depend largely on the kind of expert system employed. In some systems, for example, each attribute may be expressed as a separate frame or assertion. In practice, much greater detail will be needed, and confidence measures will be associated with some data. Also described are some functions and rules that generate attributes and new frames as needed.

The first examples are of platform frames. Attributes such as maximum speed, nation, and medium would be specified unless derivable by inheritance. Attributes that must frequently be retrieved, e.g., type and ID (IDentity), probably should be specified even when derivable. Emitters, sensors, and weapons can be listed as attributes or can be generated as needed using functions discussed later. In the example of platform_3, it is assumed that these attributes are generated from class name.

```
platform_3
    is_a = platform
    ID = friend
    name = Kinkaid
    class = Spruance
    type = destroyer
    track = track_13

platform_58
    is_a = platform
    ID = hostile
    label = contact_SA18
    type = submarine
    propulsion = diesel
    track = track_44
    poss_alias = (platform_19 platform_30) — Other subsurface
                 — contacts possibly were of this same submarine.

platform_106
    is_a = platform
    ID = hostile
    label = ESMSAT
    type = satellite
    sensor = (sensor_52 sensor_89)
    track = track_6
```

A platform can have one actual track and several virtual and postulated tracks. If it is uncertain whether a track is a continuation of an earlier track, the two tracks are associated with different platform assertions, and one (platform_i) is made a poss_alias of the other (platform_j). The syntax should allow also for giving a confidence for each possibility.

A "plat_group" frame will be needed to represent groups of platforms, e.g., several bombers in formation or a series of aircraft launched from a carrier.

The name or label could be used in place of "platform_i" instead of name or label being an attribute as shown. However, if a contact is later identified, the frame would need to be revised and all references to it modified. Instances of events are probably best numbered (e.g., detection_6) because they have no names. (In some systems, however, numbering can be avoided and uniqueness among events can still be adequately represented.) For consistency and convenience, and since many names will be unrecognized by some users, we continue the policy with emitters and sensors.

Active sensors and most communication systems are represented here both with emitter frames and with sensor or comm_system frames. Depending on the expert system used, other representation schemes could be more effective. For example, we could expand the sensor-frame types into radar, sonar, intercept receiver, etc., and indicate in another way which are also emitters.

Next are examples of emitter frames. The first is a specified emitter, and the remaining are generic emitters. Similar emitters can be lumped into a single generic one, for convenience and efficiency. Also, often the actual emitters of a contact are unknown, and generic ones must be assumed. Similarly, generic sensors and communication systems are assumed.

```
emitter_21
    is_a = emitter
    label = SPS-200
    type = radar
    function = surface_search
    band = SHF — More likely, specific frequencies
    medium = shipborne
    uniqueness = low — A common emitter reveals little
                     — about its platform.

emitter_31
    is_a = emitter
    label = generic_SS_radar_2
    type = radar
    function = surface_search
    band = SHF
    medium = shipborne
    uniqueness = medium
```

```
emitter_49
    is_a = emitter
    label = generic_clear_FH_comms
    type = communications
    band = HF
    medium = shipborne
    uniqueness = low

emitter_82
    is_a = emitter
    label = generic_ESM_jammer_3
    type = jammer
    band = SHF
    medium = airborne
```

Functions (or rule sets) will be needed to map a platform class or model to specific emitters, sensors, weapons, or other equipment, and to map a platform type to a list of generic equipment. Also, a function can be written to map platform type or class to a radar cross-section area. In the example below (sensor_1) of a radar, this area has been more coarsely mapped into large, medium, etc. Alternatively, radar ranges can be given as a list of platform type and range pairs. Daytime visual ranges and radar ranges would be expressed in a similar manner. For signal-intercepting sensors (ESM — Electronic Support Measures), specific ranges against each emitter can be listed. These may be attenuated when affected by weather or certain circumstances. In some cases, sensor detection ranges will have to be computed as needed. The following are sensor frames.

```
sensor_1
    is_a = sensor
    label = SPS-200
    type = radar
    band = SHF
    function = surface_search
    medium = shipborne
    range = ((large 30) (medium 25) (small 18) (very_small 12)
            (periscope/snorkel 2))

sensor_21
    is_a = sensor
    label = generic_SHF_ESM_3
    type = ESM
    band = SHF
    medium = shipborne
    range = ((emitter_3 32) (emitter_4 25) (emitter_5 25) ...)
```

Certain sensors and emitters are assumed to be active except during their sensor_off or emitter_off periods. These periods usually will correspond to down time or EMCON. The following are examples of an emitter_off frame and a sensor_off frame. Time is shown in examples as a date-time-group, but in practice would be stored in a form mappable also to month and year.

```
emitter_off_14
    is_a = emitter_off
    event_type = actual
    platform = platform_3
    emitter = emitter_9
    start_time = 221420
    end_time = 221600

sensor_off_92
    is_a = sensor_off
    event_type = postulated
    platform = platform_13.
    sensor = sensor_5
    start_time = 231900
    end_time = 232100
```

An "emitter_burst" frame (not shown) is applicable to emitters not used on a fairly continuous basis. This frame could be used mainly for covert communications, and another created for brief fire-control or missile-control activity. These frames and also a "jamming" frame would have start and stop times corresponding to the active period rather than to the "off" period.

The creation of an EMCON frame can result in the automatic creation, via rules, of emitter_off frames (and sensor_off frames for those emitters which are sensors). The EMCON type will be associated with different emitters from platform-to-platform, so each combination of platform and EMCON type will be associated with a list of emitters. Below is an example of an EMCON frame. During data fusion in real time, the end_time may not be known at the creation of the frame or may be modified later.

```
EMCON_18
    is_a = EMCON
    event_type = actual
    platform = platform_4
    type = alpha
    start_time = 241100
    end_time = 241140
```

Track data will include all reported positions and bearings, but a simple representation is also needed, one which can be expressed concisely and is amenable to cpa (closest point of approach) and other calculations. Connected track segments are used here. (The one illustrated later is a line segment, but arc segments can also be used.) A reported position overrides an interpolated position in short-range calculations. The following are examples of track frames.

```
track_2
    is_a = track
    event_type = actual
    platform = platform_14
    track_segment = (track_segment_13 track_segment_14 ...)
    position = (position_28 ...)
    bearing = bearing_8 — attribute for non-friends

track_9
    is_a = track
    event_type = postulated — "what_if" oneship had ...
    what_if = what_if_4
    platform = platform_15
    track_segment = (track_segment_46 ...)
```

Note that track_2 has positions and a bearing as attributes in addition to track_segment. The track segments often will be derived from the reported positions. A reported position or bearing sometimes will not be associated, in the message, with an established track. If an association cannot be made or if there are two or more possibilities, a new track and a new platform are created. In the ambiguous case, the two or more possibilities are linked to the new platform via the poss_alias attribute, as discussed earlier. The following are examples of position, bearing, and track_segment frames.

```
position_3
    is_a = position
    track = track_1
    track_confidence = .95
    — Confidence that the report belongs to track_1
    observation_time = 222245
    posit = (9.447 32.441)
    accuracy = 8
    originating_plat = platform_4
    sensor = sensor_1 — optional

bearing_3
    is_a = bearing
    track = track_1
    track_confidence = .8
    observation_time = 222320
    bearing = 240
    position = position_8 — of reporting platform
    accuracy = 5
    emitter = emitter_8
    emitter_confidence = .8
    reporting_plat = platform_4
    sensor = sensor_21 — optional
```

```
track_segment_108
    is_a = track_segment
    event_type = actual
    platform = platform_49
    start_posit = (8.500 139.020)
    end_posit = (12.000 140.320)
    start_time = 241017
    end_time = 241744
    course = 20.0
    speed = 30.0
    accuracy = 22.0
```

Track_segment_pair frames can be generated, when needed, from track_segments, as outlined in the procedural rule.

```
Rule create_track_segment_pair:
    For each segment S1i of plat1 — friendly
    and each segment S2j of plat2, — not known to be friendly
    if S1i and S2j overlap in time,
    find the segments or subsegments for the time interval in
    common and compute the cpa for the interval
    and create a track_segment_pair.
```

The cpa (closest point of approach) usually will be at the start or end time of the segment_pair time interval. The following is an example of a track_segment_pair frame.

```
track_segment_pair_410
    is_a = track_segment_pair
    event_type = actual
    friend_plat = platform_6
    other_plat = platform_49
    f_segment = track_segment_142
    o_segment = track_segment_108
    f_start_posit = (9.750 140.12)
    f_end_posit = (10.680 141.020)
    o_start_posit = (9.600 139.420)
    o_end_posit = (11.380 140.090)
    start_time = 241237
    end_time = 241625
    cpa_range = 42.38
    cpa_time = 241237
    accuracy = 25
```

If it is not known whether detections occurred, and if the cpa computed for a track_segment_pair is less than the maximum detection range over all detectable features, then the detection probability for each detectable feature is estimated (for one platform relative to the other) and detection frames are created if the confidence is greater than some minimum. Continuous detection over adjoining segment pairs can be combined into a single detection frame.

Detection frames are usually inferred, while position frames and bearing frames originate from sensor systems. Most detections will derive from track data. When inferred from tracks, the detected platform is usually a friendly platform during real-time data fusion but is frequently not a friend in post-analyses cases.

A detection may also be inferred with reciprocity rules, in which case the detected platform is a friend. A detection may have source = reported (instead of inferred) when a detection is reported with no amplifying data such as position or bearing. In the latter two cases, the attributes will be somewhat different than in the example below.

```
detection_4
    is_a = detection
    event_type = actual
    source = inferred
    detecting_plat = platform_9
    detected_plat = platform_22
    detectable_feature = plat_by_radar
    detecting_sensor = sensor_22
    track_segment_pair = (track_segment_pair_18 ...)
    start_time = 240920
    end_time = 241010
    uniqueness = low
    accuracy = 14
    confidence = .9
```

These detection assertions are then collected (for each combination of a detecting platform and detected platform), and a "targeted" assertion may be created:

```
targeted_2
    is_a = targeted
    event_type = actual
    source = inferred
    — Source could be "reported"
    targeting_plat = platform_9
    targeted_plat = platform_22
    contributing_detection = (detection_4 detection_5
    detection_6)
    start_time = 240940
    end_time = 241005
    accuracy = 5
    confidence = .8
```

A communication frame may take a variety of forms. The following is an example of the communication of probable detections by a hostile. (Inference rules are needed to determine when hostile communications probably include certain detection information.) The confidence attribute generally will not be needed for friendly communications. The values of the attribute "reported_event" will usually be inferred detections for frames representing communications among hostiles, and will frequently be positions and bearings for those among friends.

```
communication_7
    is_a = communication
    event_type = actual
    communicator = platform_33
    circuit = comms_circuit_44    — or comm_system
    recipient = platform_35
    reported_event = (detection_430 detection_431)
    receipt_time = 221715
    confidence = .55
```

Other event frames will include attack, attack_opportunity, harass, depart, arrive, and a number of frame types relating to supplies and replenishment.

What-if Example

Example. Ownship used EMCON to avoid detection by a platform estimated by dead reckoning to pass within 25 nmi. What if ownship instead avoided detection by changing course, staying well beyond detection range of the platform?

The following frames represent this in the data base. We assume there are mechanisms to treat the negated events as though they are not in the data base during the what_if analyses.

```
what_if_6
    is_a = what_if
    negated = (EMCON_8 track_120)
    postulated = (track_281 track_segment_402
                  track_segment_403)

track_281 <= modification (track_120
    event_type = postulated
    what_if = what_if_6
    track_segment = (track_segment_91 track_segment_92
                     track_segment_402 track_segment_403))

track_segment_402
    is_a = track_segment
    event_type = postulated
    platform = ...

track_segment_403
    is_a = track_segment
    event_type = postulated
    platform = ...
```

In the above example, the decision to use EMCON may have been a rule action, and the user's intent may be to postulate a change to the set of EMCON rules. In general, the user can postulate the modification to or addition of inference or doctrine rules by entering the appropriate what_if data frame. He would negate the data entered by a rule in question and/or postulate the data that the revised or new rule would enter. However, it would be desirable to add mechanisms allowing him to postulate the change to the rule set directly, and have the appropriate what_if data be generated automatically.

The analyses data base would begin with a snapshot of the reconstructed situation at the time the what_if begins to change events. (Events with start and stop times encompassing the specified time may have to be broken into two events — e.g., a track_segment broken into two.) In a multisite application, the reconstructed picture will include knowledge of what was known at each site, plus any misinformation in a site's data base. The actual events that occurred after that time would be chronologically reviewed to determine if they would have been negated by the what_if assumptions. If so, they would be tagged as negated by that what_if frame when entered into the analyses data base. This will require an extensive set of deterministic-reasoning rules (discussed in the section on System Organization). At the appropriate times, such rules can also enter postulated events, linking them to the what_if frames by their what_if attribute. The handling of postulated chance events generated by simulation or other means will be more difficult, and may require a number of repetitions of the entire process, with different chance events leading to other different chance events.

POST ANALYSES OF DATA-EXCHANGE RULES

In the discussion on the organization of a reconstruction and post-analyses system, we suggested that a set of rules could be developed to govern the exchange of data among sites. The initial set of rules probably would reflect current policy, and this set would be refined using what-if and other post-analyses methods over many exercises and operations. Here we will consider the future situation where communication is largely automated. The present kinds of messages could exist, but the tactical narratives would be immediately machine readable, since natural language processing systems would interact with message creators. In addition, links connecting the command control processing units would exchange data with little human intervention, in some generic computer representation chosen as the common language of exchange. The envisioned set of rules would control both the human and automated kinds of communication and would also incorporate EMCON doctrine. The rules would have a major effect on activities, since they would help to determine the degree of completeness of the individual site pictures and thus affect actions which, in turn, affect the situation in the future.

A regional network would connect the individual command control information processing units. A regional processing unit (RPU) could be designated, one which could transmit to, and receive from, all other units. (The RPU is very much like the "cluster head" described in ref. 3.) Rules would determine the duties of the RPU, and at times it may be the central processing system, receiving all the information of mutual interest and distributing that of interest to particular units. Normally, the RPU would be responsible for communications with the RPUs of other battlegroups. In addition to controlling communications among the sites within a battlegroup, including communication with off-board sensors, the rules should control the communications with other battlegroups, even selecting the links and frequencies to be used.

One important function of the RPU would be to detect and resolve the inconsistencies among the data bases at different sites by comparing the summary data from the sites and sending corrections where needed or notifying the sites of the discrepancies. Some of these discrepancies may occur due to inaccurate or incorrect data and some due to inferences based on incomplete data. A site finding inconsistent data in its own data base could request information concerning it from sites in the geographical area of concern. Priorities would be assigned according to the importance of the discrepancy. An extensive set of inferencing and procedural rules would be needed to automate the detection and resolution of inconsistencies.

The communications would fall essentially into two categories: information and order/request. The information type would include tracks and sightings. Rules should govern the track update rate, taking into consideration the contact's range, speed, ID, course changes, etc. However, these could be separate rule sets operating within the individual tactical data systems. A distinction would be made, of course, between exercise data and actual data, with high priorities given to truly hostile tracks during exercises. In general, the state of peace or hostilities would be an important factor in the rules.

CONCLUSIONS

Much of the data needed for good records will at some time be stored in the data base of the automated C^2 information processing systems envisioned for the near future. The "history file" created by a data fusion subsystem will record events of tactical exercises and engagements in a manner useful for event reconstruction and evaluation. Automated techniques can be designed which exploit the historical records to assist in many kinds of post analyses, and thus to improve the data fusion and planning processes and the control strategies. The emphasis in this paper has been on the data representations needed to support the analyses, and, in particular, to support what-if reasoning. Many other aspects of the problem need to be examined, and a major effort is needed to acquire and code the knowledge of the "expert" analyst.

REFERENCES

1. Dillard, R.A., Research Needs for Artificial Intelligence Applications in Support of C^3, NOSC Technical Report 1009, December 1984.

2. Dillard, R.A. Multisite Situation Assessment: Knowledge-Based Interaction, Reconstruction, and Post Analysis, NOSC Technical Document 903, May 1986.

3. Baker, D.J., Wieselthier, J., and Ephremides, A., "The HF Intratask Force Communications Network Design Study," Proc. 4th MIT/ONR Workshop, Vol. III, 7-29, 1981.

DISTRIBUTED TACTICAL C^2 SURVIVABILITY ANALYSIS

F.A. BAUSCH and R.T. BARRETT

E-SYSTEMS - CENTER FOR ADVANCED PLANNING AND ANALYSIS
10530 ROSEHAVEN STREET, SUITE 200, FAIRFAX, VA 22030

SUMMARY

A number of functional distribution options were developed for the future Tactical Air Control System (TACS). Operability and survivability analyses were then conducted to evaluate the strengths and weaknesses of the options. Key threats to the command and control (C^2) system were examined in various scenarios in order to assess the benefits of various survivability enhancements. Dispersal, mobility, and signature reduction significantly aided survivability. A major vulnerability is the communications required to support the distributed system.

APPROACH

Figure 1 illustrates the three task approach used in defining for the U.S. Air Force the operability and survivability particulars of a distributed Tactical Air Control System. The downward flow on the left hand side of the figure shows a sequence of actions that results in a survivability analysis. This in turn is a key element in the upward flow shown on the right hand side of the figure which culminates in a ranking of the most practical distribution options. This paper focuses primarily on the Task 3, survivability assessment process. Work was performed under Contract F19628-85-C-0106 for the Electronic Systems Division, Air Force System Command.

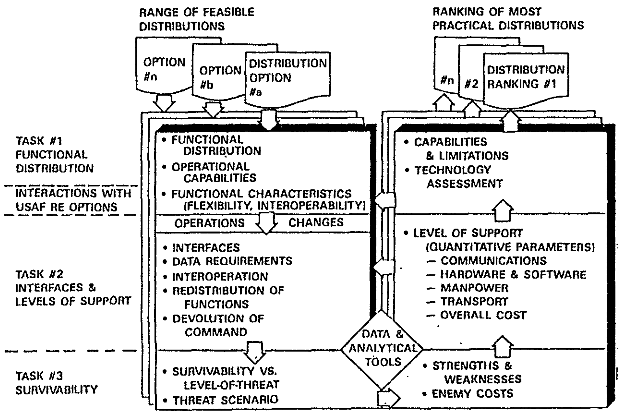

RANGE OF FEASIBLE DISTRIBUTIONS
OPTION #n
OPTION #b
DISTRIBUTION OPTION #a
RANKING OF MOST PRACTICAL DISTRIBUTIONS
#n
#2
DISTRIBUTION RANKING #1
TASK #1 FUNCTIONAL DISTRIBUTION
INTERACTIONS WITH USAF RE OPTIONS
TASK #2 INTERFACES & LEVELS OF SUPPORT
TASK #3 SURVIVABILITY
• FUNCTIONAL DISTRIBUTION
• OPERATIONAL CAPABILITIES
• FUNCTIONAL CHARACTERISTICS (FLEXIBILITY, INTEROPERABILITY)
OPERATIONS
CHANGES
• INTERFACES
• DATA REQUIREMENTS
• INTEROPERATION
• REDISTRIBUTION OF FUNCTIONS
• DEVOLUTION OF COMMAND
• SURVIVABILITY VS. LEVEL-OF-THREAT
• THREAT SCENARIO
DATA & ANALYTICAL TOOLS
• CAPABILITIES & LIMITATIONS
• TECHNOLOGY ASSESSMENT
• LEVEL OF SUPPORT (QUANTITATIVE PARAMETERS)
– COMMUNICATIONS
– HARDWARE & SOFTWARE
– MANPOWER
– TRANSPORT
– OVERALL COST
• STRENGTHS & WEAKNESSES
• ENEMY COSTS

FIGURE 1

PROJECT DESCRIPTION

Figure 2 illustrates the first option developed. In this diagram WOC-Wing Operations Centers - basically control the execution aspects of the air war. OCA/AI - Offensive Counter Air/Air Interdiction - is the C^2 element that direct deep strikes in the enemy's rear. DCA/AS - Defensive Counter Air/Air Surveillance - is the C^2 element that directs the detection, tracking and intercept of enemy aircraft entering friendly airspace. Finally, CAS/TAC Recce - Close Air Support/ Tactical Reconnaissance - are the C^2 elements that for the most part support the U.S. Army engaged in combat on the front lines. As noted above, prior to entering the survivability assessment task, each options' operability was determined to be feasible. The communications, logistics, and security requirements were judged reasonable and functional connectivity was successfully simulated. However, no electronic signatures were generated for the options since the development of a communication architecture was not a part of the project. The most notable thing about option 1 is the large size Command and Battle Staff node, which with its associated communications and support vehicles, (not shown) consists of 30 vans.

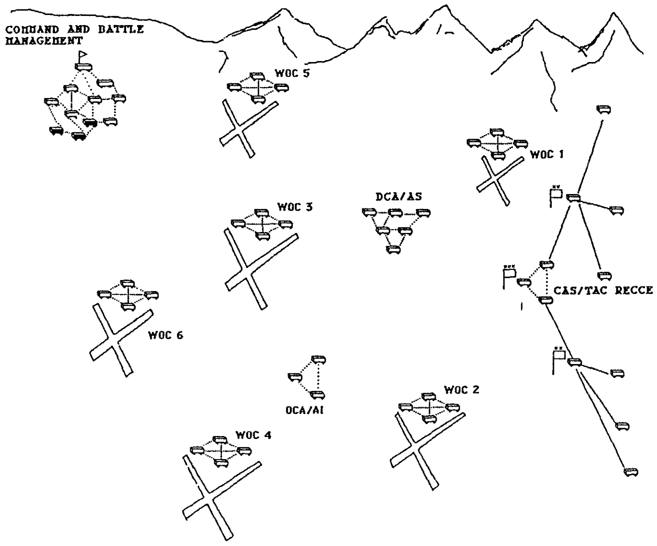


FIGURE 2
OPTION 1

The option described in Figure 3 differs from the previous one in that Battle Management functions now reside with the Wing Operations Centers, thus reducing the size of the command element. Three WOCs remain essentially unchanged from Option 1 while the DCA/AS, OCA/AI and CAS/TAC Recce C^2 Functions are now associated with the other three WOCs.

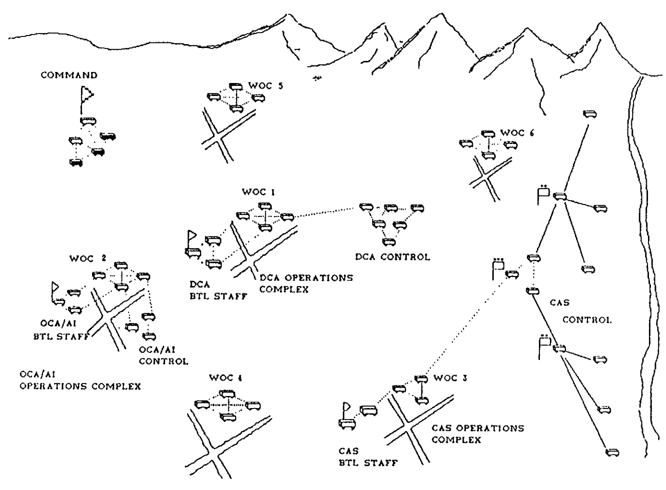


FIGURE 3
OPTION 2

The option illustrated in Figure 4 reforms the Command and Battle Management Staff adding some OCA/AI and DCA/AS C^2 functions but moves it to a remote location either in the continental U.S. or a friendly country not likely to be involved in the conflict. For the first time aircraft are used for C^2 purposes.

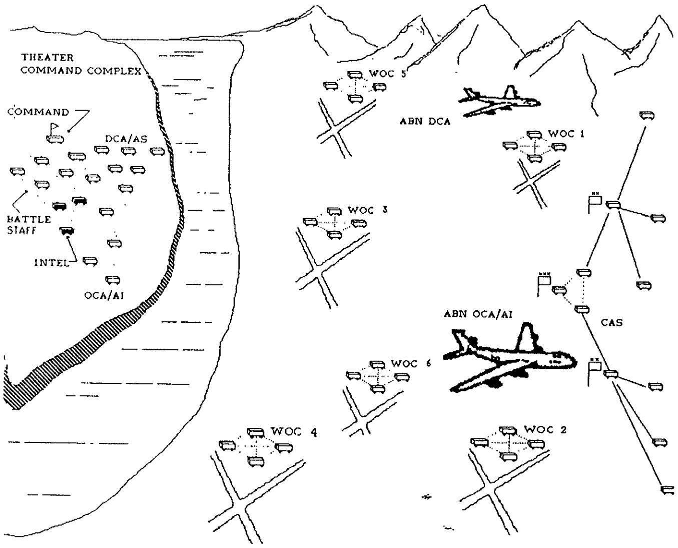


FIGURE 4
OPTION 3

Figure 5 illustrates an option which expands the airborne C^2 elements and introduces fixed, hardened facilities. In general, C^2 functions are replicated in mobile vans and fixed installations. A sanctuary is also retained.

The major items addressed in analyzing each option were scenarios, threats, and survivability enhancements. Scenarios were critical to assessing the vulnerability of the support forces and the expected intensity of the threat. The actual threat weapons considered were primarily Soviet although excursions were required for the various scenarios. The various survivability enhancements were then examined for each option.

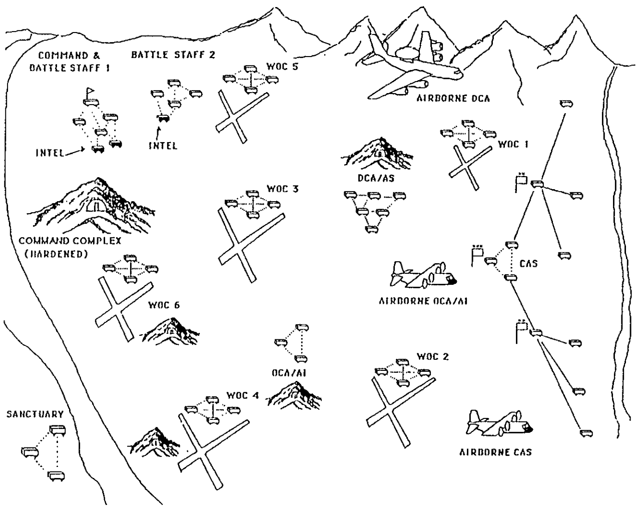


FIGURE 5
OPTION 4

The SWASIA scenario considered a Soviet invasion into Iran with U.S. Air Force units operating from both sides of the Persian Gulf. Only a moderate level of combat intensity was examined, i.e, only conventional weapons and no participation by U.S. allies or other third parties. The European scenario involved a massive Soviet air attack coincident with a major ground thrust against NATO. Three different intensity levels were examined: conventional weapons only; all weapons except nuclear e.g., chemical, ASAT, etc; and nuclear weapons. The Korean scenario did not involve the Soviet Union but focused on a massive North Korean attack directed toward seizing Seoul. The key element in assessing TACS survivability was the large scale attacks in South Korea on U.S. airbases by airborne and shipborne North Korean ranger commandos.

In each of the scenarios, we analyzed the most significant threats in respect to the most feasible means of surviving those threats. Table 1 lists the major Soviet threat categories and the five most productive survivability enhancements.

TABLE 1
THREATS AND SURVIVABILITY ENHANCEMENTS

MAJOR SOVIET THREAT CAPABILITIES	SURVIVABILITY ENHANCEMENTS
- DIRECT ATTACK A/C	- DISPERSAL/REPLICATION (REQUIRES MULTIPLE ATTACK)
- AIR TO SURFACE MISSILES	- FREQUENT MOVES (REQUIRES RAPID RESPONSE TIME)
- SURFACE TO SURFACE MISSILES	- SIGNATURE REDUCTION (LOW CONFIDENCE IN TARGET ID)
- MOBILE GROUND FORCES OMG AIRBORNE RAIDS RANGER/COMMANDOS	- PROTECTIVE COVERING (REQUIRES MASS ATTACK/"O" CEP)
- RADIO ELECTRONIC COMBAT RECONNAISSANCE ELECTRONIC WARFARE	- SANCTUARY (REQUIRES WAR WIDENING)

Figure 6 summarizes the overall structure of our analysis. This combination of scenarios, threats, and survivability enhancements provided adequate variety to fully illuminate the issues but avoided excessive detail inappropriate to the type of analysis required for each of the options.

SURVIVABILITY ENHANCEMENTS SCENARIO/ PRIMARY THREATS	DISPERSE REPLICATE (REQUIRES MULTIPLE ATTACK)	FREQUENT MOVES (REQUIRES RAPID RESPONSE TIME)	SIGNATURE REDUCTION (LOW CONFIDENCE IN TARGET ID)	PROTECTIVE COVERING (REQUIRES MASS ATTACK/ "O" CEP)	REMOTE SANCTUARY REQUIRES WAR WIDENING
SWASIA DIRECT ATTACK AIRCRAFT					
AIR TO SURFACE MISSILES					
KOREA RANGER COMMANDOS			EVALUATE DISTRIBUTION OPTIONS 1 - 4		
SURFACE TO SURFACE MISSILES					
EUROPE SURFACE TO SURFACE MISSILES					
DIRECT ATTACK AIRCRAFT					
MOBILE MANEUVER FORCES					

FIGURE 6
SCENARIOS, THREATS AND SURVIVABILITY

METHODOLOGY

Figure 7 taken from a paper presented during the 8th MIT/ONR workshop precisely describes the methodology used in assessing TACS survivability [1]. Figure 7A shows a sequence of three tasks the Soviets (Red) must accomplish in successfully attacking the TACS, i.e., seeing, deciding, and acting with their attendant probabilities. The U.S. then attempts to thwart one or more of these Red tasks to enhance TACS survivability. Figure 7B illustrates this situation. Figure 7C represents U.S. (Blue) survivability enhancements. They consist of policy, security, operational and development activities which would substantially reduce the Soviets ability to attack the TACS. In the face of this U.S. survivability program, the Soviets will then assess the situation as shown in Figure 7D. Potential Soviet threats, the Red counter-counter measures, (Figure 7E) technically feasible in the period 1990 - 2000 were then examined and the effectiveness of the U.S. survivability countermeasures program evaluated under these conditions. This was the final step in the analysis. Prior to describing the Blue countermeasures, a few observations on the Red Plan i.e., how the Soviets see, decide, and act or, in military parlance, search, target, and attack, are in order.

RED PLAN

As Table 2 indicates, the Soviets have no exotic search systems nor is their collection strategy other than a routine, common sense technique. There are, however, two vital elements that the U.S. must provide for their search process to be successful. One is communications to intercept and analyze and the other is a visual signature for targeting purposes.

TABLE 2
SOVIET RECONNAISSANCE

SENSOR PLATFORMS
- SATELLITE NEAR REAL TIME IMINT & ELINT
- AIRBORNE REAL TIME VIDEO, SAR, IR, AND ELINT
- GROUND BASED SIGINT

COLLECTION STRATEGY
- PREDICT TACS LOCATION AND SIGNATURE
- INTERCEPT AND ANALYZE TACS COMMUNICATIONS
- IDENTIFY NODES AND LOCALIZE VIA DF TECHNIQUES
- CUE SAR/IR SENSORS FOR PRECISE LOCATION
- VERIFY IDENTIFICATION & LOCATION BY IMINT

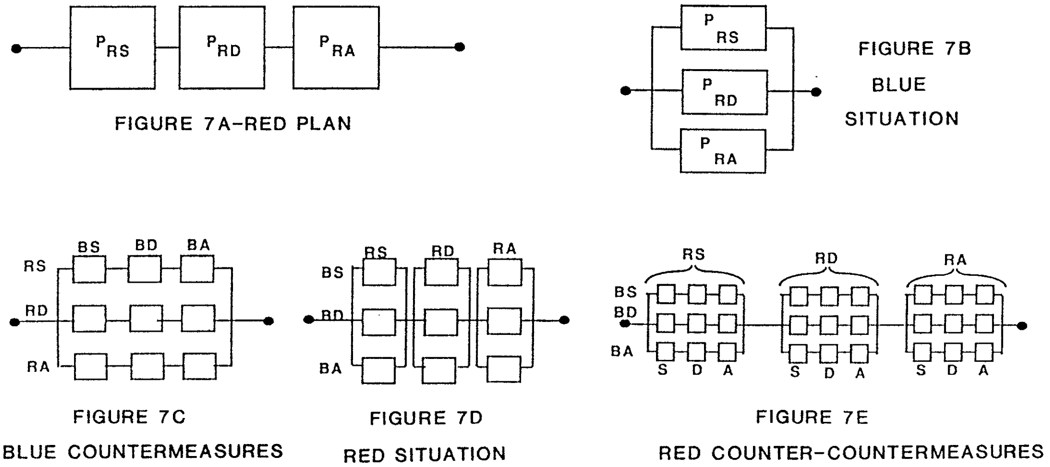


FIGURE 7A-RED PLAN

FIGURE 7B
BLUE
SITUATION

FIGURE 7C
BLUE COUNTERMEASURES

FIGURE 7D
RED SITUATION

FIGURE 7E
RED COUNTER-COUNTERMEASURES

FIGURE 7-SEQUENTIAL RELIABILITY MODEL

Soviet targeting priorities as listed in Table 3 are conditioned by their concern that a conventional war with the U.S. will escalate into a nuclear war. Therefore, they believe it is vital for them to quickly destroy the U.S. nuclear, tactical retaliatory threat. For high priority targets like airfields, U.S. C^2 nodes will likely be lesser priority targets than aircraft and their direct support activities.

TABLE 3
SOVIET TARGETING PRIORITIES

NO. 1 -	NUCLEAR CAPABLE FORCES AND SUPPORTING INFRASTRUCTURE - DELIVERY MEANS - STOCKPILES - TRANSPORT - COMMAND AND CONTROL
NO. 2 -	OTHER TARGETS AS SITUATION DEMANDS, E.G. AIRFIELD - AIRCRAFT/AIRCRAFT SHELTERS - POL STORAGE - AMMO STORAGE - C^3 FACILITIES

During the attack phase, the Soviets face the same kind of tradeoffs usually associated with weapons effectiveness planning. The Red on Blue Joint Munitions Effectiveness Manual [2] was used extensively in ascertaining rational application of the Soviet resources. The six considerations shown in Table 4 encompass main thrusts of Soviet operational art and were used in arriving at realistic Soviet force levels and application schemes.

TABLE 4
SOVIET ATTACK CONSIDERATIONS

RANGE (TARGET LOCATION/AVAILABLE ASSETS)

COMBINED WEAPONS (MAXIMIZE Pk)

WEAPONS SELECTION (AREA/POINT TARGETS)

PRIORITY (COUNTERFORCE VALUE)

TIMING (FIXED/MOBILE TARGETS)

SURPRISE (STRATEGIC/TACTICAL)

BLUE COUNTERMEASURES

Dispersal is the key survivability enhancement, and establishes the base on which to assess the efficacy of all other measures. While dispersal is implicit in the design of all options, there are two circumstances where it may be undesireable and warrant tradeoff. In the case of a threat where the front line of troops is indeterminate and enemy ground force elements are moving freely in the friendly zone, a dispersed TACS will require substantial security assets to protect. Given a fixed level for these protection forces, it may be prudent to consolidate TACS elements together and pool the defensive resources rather than be defeated piecemeal by the opposing forces. The other undesireable circumstance is where the communications required to connect the elements and nodes clearly identifies the specific TACS functions being performed and provides the Soviets locational data adequate for targeting. Thus, it may be more advantageous to consolidate the TACS if such action alleviates this condition. In general, however, dispersal must be considered the basic means of survival and clearly is the enhancement upon which all others are evaluated in terms of their additive or reinforcing value.

Equally valuable as dispersal in enhancing survivability is being able to move swiftly from place to place on the battlefield. Mobility has a major operability drawback - the support and the associated transportability (airlift) requirements. In assessing survivability however, mobility becomes the basis for many other enhancements. Vehicle symmetry allows TACS elements and nodes to mix with many other battlefield vehicles making distinction difficult. Deception and decoys meant to confuse enemy search can work best if the nodes generate locational uncertainty by occasional moves. Uncertainty then opens avenues for errors of perception by the Soviet intelligence structure. The benefits of mobility can be reinforced by using drive-in hardened bunkers and dispersing the bunkers to provide added protection. Drive-in bunkers allow mobile shelters to achieve almost the same degree of attack protection as fixed hardened facilities while retaining those attributes that defeat Soviet search and targeting.

Much like mobility, the only disadvantage to signature reduction is the potential increase in support equipment required to accomplish the enhancement. Even then, the benefits of signature reduction usually outweigh the disadvantages. This is due in large measure to the fact that signature reduction thwarts the enemy's search capability, which must be successful for subsequent targeting and attack. Even simple measures like denying the enemy a knowledge of the TACS nominal signature or down playing the importance of U.S. C^2 capabilities makes his search operations less liable to detect or identify TACS nodes. Signature reduction is the one enhancement that need not be traded off in any circumstances for any other trait. It is always useful and complementary to all other survivability enhancements.

In general, hardening is beneficial in those cases where the enemy's search and targeting capabilities cannot be defeated by any other survivability means or where such hardening does not substantially reduce the benefits other enhancements have in denying enemy search and targeting success. Fixed hardened facilities have utility, for instance, if they could be situated and operated covertly or in cases where there are absolutely no weapons that can be used successfully in an attack. In this latter case, it should be recognized that very hard targets are a spur to inventive weaponeers. The WWII 22,000 LB Grand Slam used to destroy the Bremen sub pens testifies to their success in meeting difficult targeting challenges.

On the other end of the hardening spectrum, Kevlar blankets added to soft mobile C^2 vans provide only limited protection against attack but do not affect the viability of other enhancements. A variation of this approach, i.e., the use of armored vehicles to house TACS elements provides increased protection against attack but increases the support burden and its attendant vulnerability. In only one instance can hardening enhance other survivability means. This is through the use of dispersed hardened shelters which mobile TACS elements would move to and from. This means of playing the "shell game," involves a flexible combination of all the in-theater enhancements considered so far. It preserves mobility and dispersal while providing a reduced signature and hardening at the same time. The vulnerability of the support structure still remains a problem however.

The sanctuary concept of enhancing survivability does not in itself involve tradeoffs with other type enhancements. It is, in effect, a recognition that a combination of all other in-theater features will not provide the requisite protection. There are, however, serious consequences to the failure of such an approach. If the political situation that made the sanctuary possible in the first place no longer stays viable, then some other form of protection is vital. Both dispersal and hardening are ways to achieve this protection and additionally may be disincentives to a war widening, enemy strategy. A sanctuary based on geographic remoteness from the war zone is, like very hard fixed facilities, another incentive for innovative enemy attacks. The carrier based B-25 raids on Tokyo early in WW II show that risks and costs are not the prime considerations where a high value target is concerned. It is equally important then to disperse, and probably harden TACS elements even in a geographically remote sanctuary.

Figure 8 summarizes the survivability enhancements just covered. It should be understood that a prudent military commander, even if given a guarantee that one or another enhancement would always be successful, would still opt for a mix of enhancements than can cover all bets. The blend of these enhancements with flexibility for modification by a commander in the actual combat situation is a vital ingredient in the design of a survivable TACS.

	BENEFITS			
	DEFEAT RECCE	DEFEAT TAR.	DEFEAT ATTACK	OTHER
DISPERSAL	X	X	X	ENHANCES ALL OTHER TRAITS
MOBILITY	X	X		FACILITATES DECOYS DECEPTION
SIGNATURE REDUCTION	X	X		
HARDENING			X	CAN BE USED WITH DISPERSED MOBILE CONFIGURATIONS; DRIVE IN BUNKERS OR SHELL GAME
SANCTUARY			X	

FIGURE 8
BLUE COUNTERMEASURES SUMMARY

RED COUNTER-COUNTERMEASURES

now will the Soviets respond to this design? Through the application of advanced technology, the Soviets will try to keep pace with the dispersed, mobile dynamics which the TACS represents. A new standoff recce system could provide multi-sensor coverage from behind the Soviet forward line of troops well into the U.S. rear. Using a combination of SIGINT, EO/IR and SAR sensors which are cross cued and a real time processing system that can correlate and fuse the resulting data, the Soviets could disseminate in real time the location and identify of TACS elements to various theater strike systems. The increased vulnerability of the TACS to such a threat would require greater dispersion, mobility, and signature reduction than currently required. The key point is that the direction of U.S. survivability measures remain constant; only the pace and magnitude of the effort need be enhanced.

Whereas the multi-sensor, standoff reconnaissance platform combines only search and targeting capabilities, expendable RPVs, which can seek out and destroy TACS elements on their own, constitute a total blend of Soviet search, targeting and attack capabilities. Again the necessary U.S. countermeasure is a more dispersed, more mobile, and less identifiable TACS.

Another of the technology areas that the Soviets can be expected to pursue is a long-range, extremely smart, high performance, air-to-air missile. Capable of speeds over Mach 3 and a range of 100+ miles, the missile would use both IR and millimeter wave radar to attack TACS C^2 aircraft. If the missile were provided with even a modest maneuver capability, U.S. countermeasures would be severely limited.

In the area of improved means for attack, the Soviets could develop conventional weapons that use high kinetic energy to impact hard targets. Released from space or powered downwards by rockets from the upper atmosphere, these weapons, if made from dense materials (depleted uranium for example) can easily penetrate reinforced concrete, protective earth covered bunkers. The principle is the same used for the current runway weapons except the guidance is better and impact speeds greater. Brilliant cluster munitions would be capable of seeking out and destroying small targets spread out over greater areas. Again, it appears that a flexible blend of dispersal, mobility, signature reduction and hardening will continue to provide survivability for the TACS against expected new Soviet advanced technology threats.

CONCLUSIONS

Through judicious application of the nine steps shown in Table 5, a functionally distributed TACS can be configured to survive against the Soviet threats examined. The communications needed to tie the TACS together, however is a potential major vulnerability. These communications, which could be extremely susceptible to Soviet intercept and exploitation, require further detailed examination.

TABLE 5

SURVIVABILITY STEPS

DEFEAT RECONNAISSANCE

ELUDE DETECTION
CLOUD IDENTITY
DENY LOCALIZATION

DEFEAT TARGETING

FOSTER UNIMPORTANCE
CREATE UNCERTAINTY
RAISE THRESHOLD

DEFEAT ATTACK

LIMIT DAMAGE
FACILITATE RECONSTITUTION
ELIMINATE SERENDIPITY

REFERENCES

[1] Anthony, Robert W. 1975. Probabilistic Logic Models of Military Conflict and C^3I Development. Proceedings of the 8th MIT/ONR Workshop on C^3 Systems Laboratory for Information and Decision Systems, MIT.

[2] Joint Technical Coordinating Group for Munitions Effectiveness. Effectiveness Estimates for Soviet Warsaw Pact Nonnuclear Munitions, May 1985.

CATASTROPHE THEORY AS A COMMAND AND CONTROL METHODOLOGY

Alexander E. R. Woodcock[1,2]

Synectics Corporation
111 East Chestnut Street
Rome, New York 13440

"... **The catastrophes are the surprises, all else is mere repetition ...**"

... Ian Stewart

Summary

Models based on catastrophe theory are used to illustrate the impact of command and control capabilities, firepower, and force strength on military force survivability. The validity of this approach has been established by the use of sophisticated new statistical procedures to analyse the output from elaborate simulation models of the combat environment. A framework based on catastrophe theory that can generate all the commonly used Lanchester-type of combat attrition equations (such as modern warfare, ancient combat, and area fire) as well as equations representing new types of combat process (such as smart weapons fire) has also been developed.

1. Catastrophe Theory - A Proven Framework

Invented by Thom in the 1960s (Thom (1, 2), Brocker and Lander (3), Poston and Stewart (4), Zeeman, (5), Zeeman and Trotman, (6), and Woodcock and Poston, (7)), catastrophe theory has been used in many applications in the mathematical, physical, life, and social sciences (see Arnold (8), Beaumont (9), Berry (10), Cobb (11), Gilmore (12), Hilton (13), Janich (14), Lu (15), Stewart (16), Stewart and Woodcock (17, 18), Wilson (19), Woodcock (20, 21), and Woodcock and Davis (22), for example). Catastrophe theory has also been used in a series of military applications by Woodcock (23), Woodcock and Dockery (24, 25, 26, 27), Dockery and Chiatti (28), Dockery and Woodcock (29, 30), Isnard and Zeeman (31), and Holt, Job, and Marcus (32).

Thom (1, 2) called sudden changes in behavior "catastrophes" and developed a theory (subsequently called "catastrophe theory" by Zeeman) as a new method for analysing and classifying this behavior. Catastrophe theory can be used to analyse the behavior of those systems which exhibit at least some of the properties of hysteresis, bimodality, and divergence. The theory is particularly useful under those circumstances where gradually changing forces can cause either gradual or sudden changes in behavior in the same system under different conditions. Catastrophe theory also provides a rigorous mathematical framework to support the "top-down" analysis of the behavior of complicated systems such as military systems. In this case, the theory can illustrate the impact of force strength, firepower, and command and control capabilities on military force survivability.

Elementary catastrophe theory is based on a theorem due to Thom which provides a classification of the stationary states of those systems with up to four key inputs or independent variables (called control or conflicting factors) and two outputs or dependent variables (called behavior variables) that consist of cooperating elements whose actions seek to minimize some form of potential energy-like function or property associated with the system. The behavior of such systems can be described with the aid of specialized geometric figures (known as catastrophe manifolds) that express the causal relationships between the input and output variables of the system. In applications in which the elementary catastrophes are used, an attempt is made to devise the simplest possible model (that is a model which uses as few control factors and behavior variables as possible) that can capture the essence of system behavior.

The elementary catastrophes have a complexity ranging from that of the two-dimensional fold catastrophe (with one control factor and one behavior variable) to the six-dimensional parabolic umbilic catastrophe (with four control factors and two behavior variables). Mathematical details of these catastrophes are summarized in Exhibit 1, where a, b, c, d, t, u, v, and w are control factors; and x and y are behavior variables. The popular names presented in Exhibit 1 are descriptive of the geometry of their catastrophe manifolds as shown in Woodcock and Poston (7), for example.

Exhibit 1

The Mathematical Form of the Elementary Catastrophes

Popular Name of Catastrophe	Control Factors or Input Variables	Behavior or Output Variables	Potential Function
Fold	1	1	$x^3/3 + ax$
Cusp	2	1	$x^4/4 + ax^2/2 + bx$
Swallowtail	3	1	$x^5/5 + ax^3/3 + bx^2/2$
Butterfly	4	1	$x^6/6 + ax^4/4 + bx^3/3 + cx^2/2 + dx$
Hyperbolic Umbilic	3	2	$x^3 + y^3 + wxy + ux + vy$
Elliptic Umbilic	3	2	$x^3 - 3xy^2 + w(x^2 + y^2) + ux + vy$
Parabolic Umbilic	4	2	$x^2y + y^4 + ty^2 + wx^2 + ux + vy$

The elementary catastrophe potential functions consist of the sum of two components called the germ ($g_{CC}(x)$) and unfolding ($u_{CC}(x)$) of the function. The cusp catastrophe potential function ($V_{CC}(x)$) is:

$$V_{CC}(x) = g_{CC}(x) + u_{CC}(x) = x^4/4 + ax^2/2 + bx \qquad (1)$$

where x is the behavior variable and a and b are control factors. The stationary states of this potential function, which are given by equation (2), can be represented by a three-dimensional (x, a, b) curved surface known as the cusp catastrophe manifold (see Exhibit 2, which presents a two factor model of military behavior based on the cusp catastrophe).

[1] Supported by Contract MDA903-85-C-0262 to Synectics Corporation.

[2] The author gratefully acknowledges the collaboration and continuing interest in this work of Dr. J. T. Dockery, The Organization of The Joint Chiefs of Staff, The Pentagon, Washington D.C. 20301.

$$dV_{CC}(x)/dx = x^3 + ax + b = 0 \qquad (2)$$

Thom's theorem claims that the stationary state behavior of all systems (including physical, chemical, biological, and societal systems) with up to four control factors, two behavior variables, and an associated potential-like function, can be described with the aid of one of the elementary catastrophes. Subsequent mathematical analysis by Zeeman and Trotman (6), for example, has proved Thom's theorem. The use of catastrophe theory to model a particular system, such as that associated with military combat, will require the identification of a suitable set of control factors and behavior variables.

While elaborate simulations can provide considerable insight into the nature of the combat process, the complexity and volume of the data that they produce often serves to reduce their utility as a guide for the military commander. By contrast, the catastrophe models provide a method for presenting information in a new framework which can aid comprehension and "turn old facts into new knowledge" (Thompson (33)).

2. Catastrophe Theory-Based Models of Military Behavior

Two- and four-factor models of military behavior, which are based on the cusp and butterfly catastrophes, respectively, have been developed by Woodcock and Dockery (24, 25). The development of these models was motivated, in part, by the need to provide models with a relatively few control factors that resemble the properties or "axes" around which a military commander may organize his perceptions. Following this work, Dockery and Chiatti (28) have used a statistical computer program developed by Cobb (34) to fit simulated combat data to the surface of the cusp catastrophe manifold.

Catastrophe theory-based models can provide a useful type of interface between the military commander on the one hand and elaborate computer-based combat simulations which provide large amounts of data on the other. These models can provide a series of diagrams, which Woodcock and Dockery have called "problem-solving landscapes," that can aid the commander in tracking events during combat and supporting such activities as decision-making, situation assessment, and impact analysis on an essentially real-time basis.

2.1 A Cusp Catastrophe-Based Model

The first catastrophe theory-based model of the combat process developed by Woodcock and Dockery describes the impact of opposing Red and Blue Forces on the survivability of the Blue Forces. This model involves the impact of two conflicting factor influences on system behavior and, as a consequence of Thom's theorem, is based on the cusp catastrophe. The concept of using conflicting factors (which is due to Zeeman (31)) in place of control factors was employed by Woodcock and Dockery (24, 25, 26, 27) in both the two-factor and four-factor models in order to capture the inherently conflictual nature of the military combat process. Analysis has shown that conflicting factors provide a very good basis for developing an intuitive understanding of military behavior involving conflict as described in this paper.

In the two-factor model of military behavior (shown in Exhibit 2), Woodcock and Dockery (24, 25, 26) have identified two conflicting factors and the behavior variable associated with military situations as follows:

1. Factor 1 represents influences which impose order on a defending (or attacking) (Blue) force and will be referred to as the Blue Force Strength.

2. Factor 2 represents influences which impose order on an attacking (or defending) (Red) force and will be referred to as the Red Force Strength.

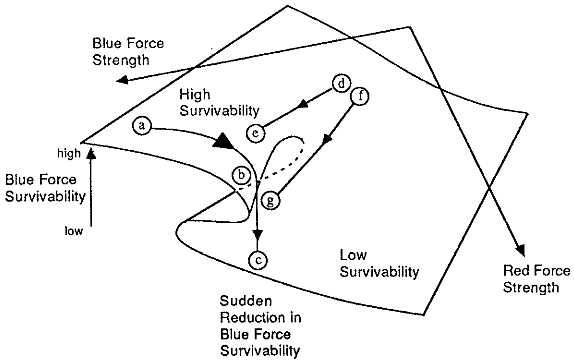
3. The Behavior Variable represents some manifestation of an order-disorder transition in a military combat situation such as the Survivability of the Blue Force.

Two areas of the cusp catastrophe manifold, which represent two qualitatively different types of system behavior, can be identified in Exhibit 2. One area of the manifold, arbitrarily located on its upper surface, represents a condition of high Blue Force survivability or force integrity. The lower layer of the surface (below the pleat) represents a condition of low Blue Force survivability or force destruction. Conditions of high Blue and low Red Force strengths are represented by a point (called the state point) at position (a) (Exhibit 2) on the catastrophe manifold surface. As the combat proceeds, a relatively large reduction in the strength of the Blue Force (caused by combat attrition) and increase in that of the Red Force (caused by reinforcement) can lead to a rapid decrease in Blue Force survivability (path (a-b-c), Exhibit 2).

Exhibit 2

A Military Analysis and Problem-Solving Landscape based on the Cusp Catastrophe

- Force balance determines force survivability

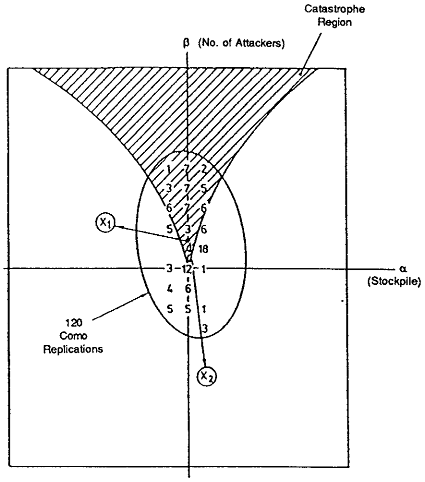


The build-up of opposing forces from low levels is illustrated in Exhibit 2. Such a build-up can lead to a condition of high Blue Force survivability (illustrated by path (d-e)). By contrast, a build-up which slightly favors the Red Forces can lead to a condition in which the Blue Force has a low survivability even before the combat has begun (illustrated by path (f-g)). While conditions characterized by points (e) and (g) in this exhibit may have closely similar values of Blue and Red Force strength, they support markedly different Blue Force survivability conditions. This behavior can represent ambiguous conditions in which a Blue Force commander perceives that he is in a strong military position compared with an opponent but his military position is actually very weak. This difference in Blue Force strength can be caused by a sequence of apparently insignificant events occurring during the initial build-up of forces before actual combat has taken place.

This two-factor model has illustrated the impact of Blue and Red Force strength on the survivability of the Blue Force. Together with the fitting of simulated data with a program developed by Cobb (34) by Dockery and Chiatti (28), one result of which is illustrated in Exhibit 3, this model provides an anchor for further investigations. Woodcock and Dockery (24, 25) have demonstrated that it is possible to describe even more elaborate patterns of military behavior with a four-factor catastrophe model in which the catastrophe factors or influences are associated with processes which impose short, inte-mediate, and long range order on the combat environment.

Exhibit 3

Application of Catastrophe Theory to Military Analysis



Control plane plot of 120 replications of an event driven simulation. The independent vaiables are stockpile (X_1) and number of attackers (X_2). Values of X_1, X_2 used were total over several time periods.

From: Dockery, John and Stefano Chiatti (28).

Such a line of reasoning takes quite literally the concept that lower dimensional manifolds are embedded in those of higher dimensions. The four factor model refers to work based on the butterfly catastrophe with four factors and one behavior variable. The swallowtail catastrophe (with three factors and one behavior variable) was not used as a model of military behavior since it possesses conditions under which no stationary state can occur.

2.2 A Butterfly Catastrophe-Based Model

The second model describes the impact of Red and Blue Force strength, firepower, and command and control capabilities on the survivability of the Blue Forces. This model is based on the butterfly catastrophe (Exhibt 4) with the following identifications being made:

1. Factor 1 represents influences which are assumed to impose short range order on a defending (or attacking) (Blue) force which will be called the Blue Force Strength.
2. Factor 2 represents influences which are assumed to impose short range order on an attacking (or defending) (Red) force, which will be called the Red Force Strength.
3. Factor 3 represents the relative levels of firepower of the opposing forces, reflects influences which are assumed to impart intermediate range order on these forces, and will be called the Firepower Balance.
4. Factor 4 represents the influence of those factors, described as the Command and Control Capability available to the Blue forces, that are assumed to impose long range order on the Blue Forces.
5. The Behavior Variable represents the Survivability of the Blue Force.

In this four-factor model the conflicting factor axes represent the inherent strengths of the opposing Red and Blue Forces in terms of their levels of training or group cohesiveness, but without consideration of the relative levels of firepower or command and control capabilities or other longer range assets such as helicopter forces. The crucial thing is that these longer range assets represent influences external to the forces' inherent strength.

2.2.1 The Effect of Firepower and Command and Control Capabilities on Force Survivability The relative firepower and command and control capabilities available to the two forces are represented by scales with indicator arrows in Exhibit 4. The position of the indicator arrow represents the existing value of the particular factor and a change in such a value, represented by a movement of this arrow, causes a corresponding change in the shape of the manifold. We track these changes by monitoring the changes that they produce in the shape of the "footprint" of the folded region of the manifold.

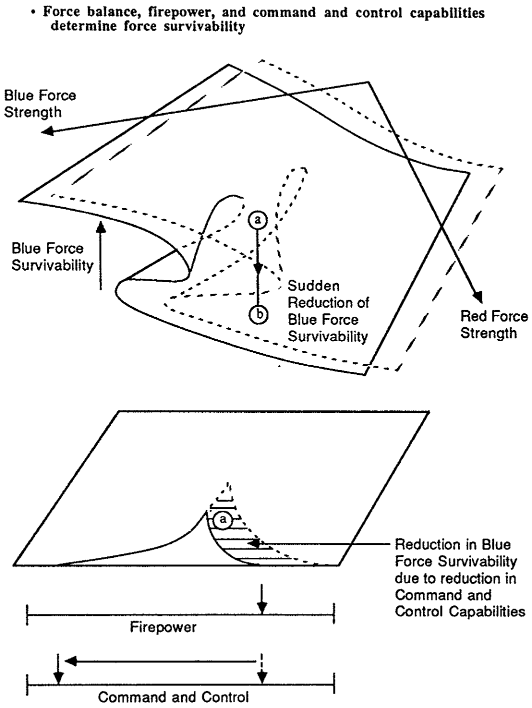
Conditions in which the Blue Force has a significant advantage in firepower and command and control capabilities compared to the Red Force are illustrated in the model by a distortion of the manifold surface to produce a relatively large region that is identified with conditions of high Blue Force survivability (Exhibit 4). These additional influences can thus off-set the effect of a relatively low intrinsic Blue Force strength.

A reduction in the Blue Force command and control capability advantage during combat will lead to a reduction in Blue Force survivability. Under these circumstances, it would perhaps appear to the Blue Force Commander that the "ground" was falling away from under his feet as the folded region of the surface moves in response to changes in the level of command and control capabilities. Here the model illustrates how a drastic reduction in command and control capabilities (caused by the destruction of a key command center, for example) can cause a significant decrease in the survivability of these forces. Such an event can be represented by the movement of the state point from the upper (or high survivability) to the lower (or low survivability) region of the manifold (path (a-b), Exhibit 4) in response to a decrease in the level of Blue Force command and control capabilities. The hatched area drawn on the plane in this Exhibit represents those sets of factor values for which a reduction in Blue Force survivability can occur as the result of such a reduction in the command and control capability of this Force.

The next section of the paper shows how catastrophe theory can provide a framework for generating many of the well-known Lanchester-type combat attrition models.

Exhibit 4

A Military Analysis and Problem-Solving Landscape based on the Butterfly Catastrophe


• Force balance, firepower, and command and control capabilities determine force survivability
Blue Force Strength
Blue Force Survivability
a
b
Sudden Reduction of Blue Force Survivability
Red Force Strength
a
Reduction in Blue Force Survivability due to reduction in Command and Control Capabilities
Firepower
Command and Control

3. Lanchester Equations as Models of the Combat Process

Lanchester-type combat models ((35), also Taylor (36), for example) consist of a series of time-dependent deterministic differential equations which represent different combat attrition processes. Initially motivated by the need to provide a mathematical justification for the principle of "force concentration" (which can be stated that, in war time, the best strategy is to keep one's forces concentrated in a particular location) Lanchester's work has subsequently provided the basis for the modern mathematical analysis of the combat process.

The initial development and application of Lanchester-type equations has also been driven by the need for a deterministic differential equation that could be integrated. Numerical solutions of complex, multiterm, Lanchester equations have been obtained, particularly by Bonder and Honig (37) for example) in a series of models known as VECTOR II. The restriction to deterministic cases has been overcome by the recent development by Cobb and Harrison (38) of a software package which can be used to integrate stochastic and highly complicated versions of the Lanchester-type combat equations.

3.1 Specific Combat Models

The following is a description of some of the most well known and more widely used deterministic Lanchester-type combat models.

1. Lanchester ancient combat conditions consist of a series of one-on-one duels with personal weapons, such as swords or hand guns, and can be described by the following equations:

$$dx/dt = - a \qquad dy/dt = - b \tag{3}$$

Where the coefficients (a) and (b) represent the combat effectiveness of the two opposing forces (x) and (y).

2. Lanchester modern warfare conditions are those in which the number of casualties are considered to be directly proportional to the number of combatants, and can be represented mathematically as:

$$dx/dt = - ay \qquad dy/dt = - bx \tag{4}$$

where the coefficients (a) and (b) are known as the Lanchester modern warfare attrition coefficients. It is these equations that most often come to mind when people speak about "Lanchester equations."

3. Lanchester area fire combat conditions are those in which each side fires in a uniform manner into the general area occupied by the opposing side, but not at specific targets, and are described by the equations:

$$dx/dt = - axy \qquad dy/dt = - bxy \qquad (5)$$

where the coefficients (a) and (b) are known as the Lanchester area fire coefficients.

When groups of individuals engaged in firing weapons are represented by the symbol F (firers) and those individuals who are the targets of these firers are represented by the symbol T (targets), Lanchester modern warfare conditions can be classified as an (F | F) attrition process while area fire is an (FT | FT) attrition process, for example.

4. An (F | FT) attrition process occurs when the x forces attack "well-dug-in" y forces, and can be described by the following equations:

$$dx/dt = - ay \qquad dy/dt = - bxy \qquad (6)$$

5. An (F + T | F + T) Lanchester-type attrition process is one in which the combat between two forces (labeled x and y) causes losses as the result of enemy action, and by one or more "self-inflicting" processes such as desertion or sickness, or the impact of "friendly" fire. Such a process can be described by the following equations:

$$dx/dt = - ay - mx \qquad dy/dt = - bx - ny \qquad (7)$$

where (a) and (b) are attrition coefficients representing the impact of adversarial fire and (m) and (n) are coefficients representing self-inflicted attrition.

6. A (T | T)-type attrition process involving the use of camouflage and concealment are described by the following equations:

$$dx/dt = - ax \qquad dy/dt = - by \qquad (8)$$

where (a) and (b) are Lanchester attrition coefficients.

4. A Framework for Generating Lanchester-type Combat Equations based on Catastrophe Theory

A general framework based on catastrophe theory that generates the Lanchester-type combat equations described in equations (3) to (8) has been developed by Woodcock and Dockery (26). This framework serves to unify the Lanchester approach to modeling combat dynamics. The Lanchester-type combat equations are time-dependent differential equations and the various terms of these equations have been identified, as described below, with terms in sets of equations obtained by partially differentiating the catastrophe functions with respect to the behavior variables (x) and (y).

An important motivation for this analysis was the realization that, for potential functions of the form V(x,y), the following relationships exist:

$$dx/dt = - \partial V(x,y)/\partial x \qquad dy/dt = - \partial V(x,y)/\partial y \qquad (9)$$

(see Poston and Stewart (4), and Guckenheimer and Holmes (39), for example). Thus, if the properties (x) and (y) in equations (9) are identified as functions of the strength of two military forces, then the left-hand-sides of these equations are equivalent to the left-hand-sides of the Lanchester-type of combat equations presented in Section 3. The right-hand-sides of equations (9) can be derived by the differentiation of a catastrophe type of potential function with respect to its (x) and (y) behavior variables, as described below.

Since there is no *a priori* method for drawing distinctions between the opposing (x) and (y) forces in a combat situation, a catastrophe function used as a generator of attrition coefficient relationships for such forces should be symmetrical in its treatment of the behavior variables describing these forces. Under these circumstances, the fold, cusp, swallowtail, and butterfly catastrophes, which describe system behavior in terms of a single (x) behavior (or output) variable, will not be suitable generators of such attrition coefficient relationships. By contrast, the umbilic catastrophes, which describe system behavior in terms of two (x) and (y) variables are potentially suitable generators of attrition coefficient relationships since they provide a method for considering the impact of changes in system controls on two separate output parameters.

Of the three umbilic catastrophes (Exhibit 1), only the hyperbolic umbilic catastrophe has a germ ($x^3 + y^3$) that is symmetrical in x and y, and so is a suitable candidate to act as a generator of Lanchester-type equations. Woodcock and Dockery (26) have shown that the hyperbolic umbilic catastrophe can be used as a generator of relationships representing ancient combat and modern warfare and that the use of the double cusp catastrophe is required to generate the remainder of the relationships presented in Section 3.

4.1 The Double Cusp Catastrophe as a Generator of Lanchester-type Attrition Relationships

The double cusp catastrophe has two behavior variables (x) and (y) and eight control factors (labeled a, b, c, d, e, f, g, and h) and is not a member of the list of elementary catastrophes defined by Thom (1, 2). The potential function associated with the double cusp catastrophe ($V_{DC}(x,y)$), which contains both germ ($g_{DC}(x,y)$) and unfolding ($u_{DC}(x,y)$) expressions, is the following equation:

$$V_{DC}(x,y) = g_{DC}(x,y) + u_{DC}(x,y) = x^4 + y^4 + ax^2y^2 + bx^2y + cxy^2 + dx^2 + ey^2 + fxy + gx + hy \qquad (10)$$

Partial differentials of the double cusp equation (10) with respect to the (x) and (y) behavior variables ($\partial V_{DC}(x,y)/\partial x$ and $\partial V_{DC}(x,y)/\partial y$, respectively) generate expressions which describe the impacts of x-related and y-related changes, respectively, on the behavior of the system. Stationary states (which can be represented geometrically by the double cusp catastrophe manifold) occur when these partial derivatives are set equal to zero. These partial differentials have the form:

$$\partial V_{DC}(x,y)/\partial x = 4x^3 + 2axy^2 + 2bxy + cy^2 + 2dx + fy + g \qquad (11)$$

$$\partial V_{DC}(x,y)/\partial y = 4y^3 + 2ax^2y + bx^2 + 2cxy + 2ey + fx + h \qquad (12)$$

These equations consist of two different sets of components, those derived from the germ of the catastrophe and those derived from its unfolding. In the context of military modeling, this separation into germ-derived and unfolding-derived types can be considered to reflect the separation of military activity into strategic and tactical domains, respectively. In the military sense, the strategic environment provides a backcloth against which tactical activities can take place. In the catastrophe sense, the germ of the catastrophe provides the backcloth against which the unfolding terms can exert their influence. Thus, models of tactical behavior based on catastrophe functions such as the double cusp should include only terms generated from the unfolding terms of these functions.

Following this line of reasoning, Woodcock and Dockery (26) have written the following multiple element combat equations involving tactical activities which are based on the double cusp catastrophe:

$$dx/dt = - 2axy^2 - 2bxy - cy^2 - 2dx - fy - g \qquad (13)$$

$$dy/dt = - 2ax^2y - bx^2 - 2cxy - 2ey - fx - h \qquad (14)$$

Inspection of these equations reveals that the x force can manipulate the (d) and (g) coefficients without influencing the y force and that the y force can manipulate the (e) and (h) coefficients without influencing the x force (Exhibit 5). However, both forces share access to processes represented by the (a), (b), (c), and (f) coefficients. Thus, if one force establishes the value of one or more of these terms by defining the nature of the combat environment, for example, then this will directly restrict the potential actions of the other force.

Woodcock and Dockery (26) have shown that it is possible to make the following identifications between Lanchester-type combat relationships on the one hand, and the coefficients of the double cusp catastrophe-based multiple element combat equations on the other (Exhibit 6):

1. The (g) and (h) terms are Lanchester ancient combat relationships.

2. The (fy) and (fx) terms are restricted modern warfare relationships in which access to the (f) coefficient is shared between the (x) and (y) forces. (However, Woodcock and Dockery (26) have used a simple transformation to link the separate terms (fy and g) and (fx and h) to

produce attrition coefficient relationships ((g'y) and (h'x), respectively) which have the standard modern warfare format of equations (4)).

3. The (2bxy) and (2cxy) terms are Lanchester area fire relationships.

4. The (fy) and (2bxy) terms are (F) and (FT)-type attrition relationships, respectively.

5. The (fy+2dx) and (fx+2ey) terms are (F+T)-type attrition relationships.

6. The (2dx) and (2ey) terms are (T | T)-type attrition relationships.

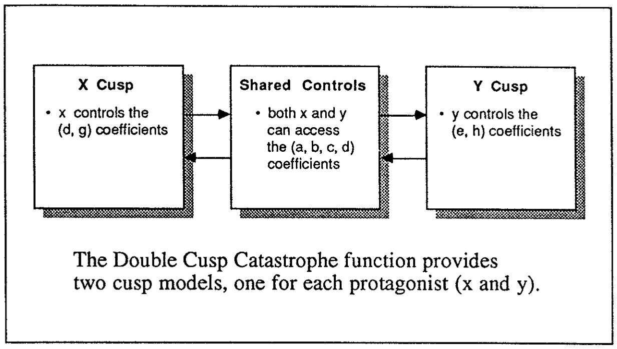
Two additional relationships, which do not resemble those typical of the classic type of Lanchester attrition processes can be identified as a result of this catastrophe theory-based analysis. These relationships are:

7. The (cy^2) and (bx^2) terms can be identified as (F^2)-type attrition relationships, and may be considered to result from the impact of so-called "smart-weapons" fire.

8. The ($2axy^2$) and ($2ayx^2$) terms can be identified as (F^2T)-type attrition relationships, and can be considered to be a result of the impact of combined weapons fire.

Exhibit 5

Independent and Shared Controls

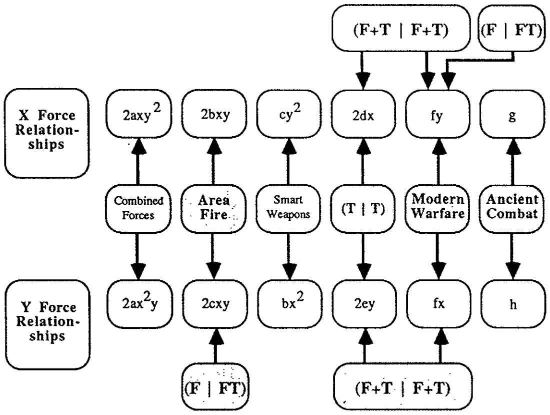

X Cusp
• x controls the (d, g) coefficients
Shared Controls
• both x and y can access the (a, b, c, d) coefficients
Y Cusp
• y controls the (e, h) coefficients

The Double Cusp Catastrophe function provides two cusp models, one for each protagonist (x and y).

This analysis shows that the double cusp catastrophe provides a mathematical structure that can serve as a framework for generating significant numbers of Lanchester-type combat equations on the basis of mathematically rigorous rules. These equations can include contributions to an overall picture of combat involving ancient combat, modern warfare, area fire, (F | FT) attrition, (F + T | F + T) attrition, and (T | T) attrition. Also included in this picture are contributions that can be interpreted as reflecting the impact of "smart" weapons and combined forces fire on the attrition process (Exhibit 6).

Exhibit 6

Double Cusp Generation of Lanchester Attrition Coefficient Relationships



A review of Exhibit 6 reveals, however, that the double cusp-based combat equations have properties which are more elaborate than those of the simple Lanchester equations (see equations (3) to (8)). Thus, while the double cusp function can generate the area fire terms (2bxy) and (2cxy) (so that the coefficients (b) and (c) in the multiple element combat equations (13) and (14) must have non-zero values), these area fire terms will also be associated with the so-called "smart" weapons terms (cy^2) and (bx^2) for the (x) and (y) force relationships, respectively. Under these circumstances, it is possible to write the following combat equations:

$$dx/dt = - 2bxy - cy^2 = - y (2bx + cy) \qquad (15)$$

$$dy/dt = - bx^2 - 2cxy = - x (bx + 2cy) \qquad (16)$$

These combat equations appear to describe a hybrid form of modern warfare attrition process (see equations (4)) involving both area fire and smart weapons fire in which the coefficients of the attrition process of the (x) and (y) forces in equations (4) are replaced by the terms (2bx + cy) and (bx + 2cy) in equations (15) and (16), respectively. Equations (15) and (16) are obviously more elaborate than the "standard" Lanchester modern warfare equations and an analysis reveals that the nature of the attrition that they describe is dependent on the sign and relative values of the (b) and (c) coefficients.

Thus, while the initial motivation for the use of the catastrophe functions was simply the need to produce a generator of attrition coefficient relationships of the Lanchester type, this work has led to the possible identification of hybrid combat environments in which force attrition may be caused by more than one process. While the Lanchester equations describe different types of simple combat process where only one type of combat can take place at any given time, the catastrophe analysis suggests that particular types of combat attrition process may be interdependent so that they might take place simultaneously. Thus, when used in this way, the catastrophe theory-based analysis may provide new clues to the nature of the elaborate multi-force interactions that can take place on the modern battlefield. A further analysis of this possibility is the subject of on-going research.

The relatively recent development of so-called "smart" weapons has provided the opportunity to define a new type of functional relationship which can serve as the basis of a new set of attrition process equations. Under conditions in which smart weapons are being used, it is proposed that the firing of a weapon will depend upon the cooperative interaction between two members of the force that is engaged in firing the weapon. One of these members could employ sensors for target detection and control the trajectory of the weapon while the other member would actually load and fire the weapon, for example.

Under such circumstances, combat involving "smart" weapons could be described by the following equations:

$$dx/dt = - cy^2 \qquad dy/dt = - bx^2 \qquad (17)$$

where (b) and (c) are smart weapons fire attrition coefficients.

Manipulation and integration of these equations produces the following cubic law combat equation:

$$b\{x_0^3 - x(t)^3\} = a\{y_0^3 - y(t)^3\} \qquad (18)$$

where x_0 and y_0 are the initial force strengths at time (t = 0) and x(t) and y(t) are the force strengths at time (t). It is believed by the author that this is the first time that a cubic-law combat equation of this type has been identified.

This paper has presented several uses of catastrophe theory as illustrations of the development of a new approach to modeling military combat and the analysis of the impact of command and control capabilities on military behavior. This work is part of a larger endeavor, the results of which may be made available in due course in other publications.

5. Bibliography

(1) Thom, R. (1969). "Topological models in biology." *Topology*, **8**: 313-335.

(2) Thom, R. (1975). *Structural Stability and Morphogenesis.* Reading Mass.: W.A. Benjamin.

(3) Brocker, Th. and L. Lander. (1975). *Differential Germs and Catastrophes.* London Mathematical Society Lecture Notes, **17.** Cambridge: Cambridge University Press.

(4) Poston, Tim. and Ian Stewart. (1978). *Catastrophe Theory and Its Applications.* London: Pitman.

(5) Zeeman, E.C. (1977). *Catastrophe Theory, Selected Papers 1972-1977.* Reading Mass.: Addison Wesley.

(6) Zeeman, E.C. and D. Trotman. (1976). "The classification of elementary catastrophes ≤ 5." In: *Structural Stability, The Theory of Catastrophes, and Applications.* (Hilton, P. (ed.)). *Lecture Notes in Mathematics.* **525:** 263-327. Berlin and New York: Springer Verlag.

(7) Woodcock, A.E.R. and T. Poston. (1974) *A Geometrical Study of the Elementary Catastrophes. Lecture Notes in Mathematics,* **373,** Berlin and New York: Springer Verlag.

(8) Arnold, V.I. (1984). *Catastrophe Theory.* (Tr. by R.K. Thomas). Berlin and New York: Springer Verlag.

(9) Beaumont, J.R. (1982). "Towards a conceptualization of evolution in environmental systems." *Int. J. Man-Machine Studies,* **16:** 113-145.

(10) Berry, M.V. (1976). "Waves and Thom's theorem." *Adv. Phys.* **25:** 1-26.

(11) Cobb, L. (1978). "Stochastic catastrophe models and multimodal distributions." *Behavioral Science,* **23:** 360-374.

(12) Gilmore, R. (1981). *Catastrophe Theory for Scientists and Engineers.* New York: Wiley Interscience.

(13) Hilton, P. (ed). (1978). *Structural Stability, The Theory of Catastrophes, and Applications. Lecture Notes in Mathematics,* **525.** Berlin and New York: Springer Verlag.

(14) Janich, K. (1974). "Caustics and catastrophes." *Math. Ann.* **209:** 161-180.

(15) Lu, Y. -C. (1980). *Singularity Theory and an Introduction to Catastrophe Theory,* Berlin and New York: Springer Verlag.

(16) Stewart, I. (1981). "Applications of catastrophe theory in the physical sciences." *Physica,* **2D:** 245-305.

(17) Stewart, I.N. and A.E.R. Woodcock. (1981). "On Zeeman's equations for the nerve impulse." *Bulletin Mathematical Biology.* **43:** 279-325.

(18) Stewart, Ian and Alexander Woodcock. (1984). "Bifurcation and hysteresis varieties for the thermalchainbranching model II: positive modal parameter." *Math. Proc. Camb. Phil. Soc.* **96:** 331-349.

(19) Wilson, A. (1981). *Catastrophe Theory and Bifurcation Applications to Urban and Regional Systems.* London: Croom-Helm.

(20) Woodcock, A.E.R. (1979). "Catastrophe theory and cellular determination, transdetermination, and differentiation." *Bull. Math. Biol.* **41:** 101-117.

(21) Woodcock, A.E.R. (1978). "On the geometry of space- and time-equivalent catastrophes." *Bull. Math. Biol.* **40:** 1-25.

(22) Woodcock, Alexander and Monte Davis. (1978). *Catastrophe Theory.* New York: E.P. Dutton.

(23) Woodcock, A.E.R. (1986). *An Investigation of Catastrophe Theory as a Command and Control Device.* Rome, New York: Synectics Corporation.

(24) Woodcock, A.E.R. and J.T. Dockery (1984 a). *Application of Catastrophe Theory to the Analysis of Military Behavior.* The Hague, The Netherlands: SHAPE Technical Centre. Consultants Report. **STC: CR-56.**

(25) Woodcock, A.E.R. and J.T. Dockery. (1984 b). "Artificial intelligence and catastrophe theory." In: *The Use of Artificial Intelligence in the Analysis of Command and Control.* By Dockery, J.T. and J. van den Driessche. The Hague, The Netherlands: SHAPE Technical Centre. Technical Report. **STC: TM-749.**

(26) Woodcock, A.E.R. and J.T. Dockery. (1986 a). "Models of combat I: catastrophe theory and the Lanchester equations." In: (23).

(27) Woodcock, A.E.R. and J.T. Dockery. (1986 b). "Models of combat IV: population-dynamics models of military behavior." In: (23).

(28) Dockery, John and Stefano Chiatti. (1986). "Application of catastrophe theory to the problems of military analysis." *European Journal of Operations Research.* **24:** 46-53.

(29) Dockery, J.T. and A.E.R. Woodcock. (1986 a). "Models of combat II: catastrophe theory and chaotic behavior." (Pre-print)

(30) Dockery, J.T. and A.E.R. Woodcock. (1986 b). "Models of combat III: combat rheology." (In Preparation).

(31) Isnard, C.A. and E.C. Zeeman. (1976). "Some models from catastrophe theory in the social sciences." In: *The Use of Models in the Social Sciences.* Collins, L. (ed). pp. 44-100. London: Tavistock Publications.

(32) Holt, R.T., B. Job, and L. Marcus. (1978). "Catastrophe theory and the study of war." *Journal of Conflict Resolution,* **22:** 171-208.

(33) Thompson, D'A. W. (1917). *On Growth and Form.* Cambridge: Cambridge University Press.

(34) Cobb, L. (1983). *A Maximum Likelihood Computer Program to fit a Statistical Cusp Hypothesis.* The Hague, The Netherlands: SHAPE Technical Centre.

(35) Lanchester, F.W. (1914). "Aircraft in warfare: the dawn of the fourth arm- No. V., the principles of concentration." *Engineering* **98:** 422-423.

(36) Taylor, J.G. (1983). *Lanchester Models of Warfare, Volumes I and II.* Alexandria, Virginia: Operations Research Society of America.

(37) Bonder, S. and J.G. Honig. (1971). "An analytical model of ground combat: design and application." In: *Proceedings of the Tenth Annual U.S. Army Operations Research Symposium.* Durham, North Carolina.

(38) Cobb, L. and G. Harrison. (1985). *A Computer Program to Solve Stochastic Lanchester Equations.* Washington, D.C.: The Organization of the Joint Chiefs of Staff, The Pentagon.

(39) Guckenheimer, J. and P. Holmes. (1983). *Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields.* New York: Springer Verlag.

A DISTRIBUTED INFORMATION SYSTEM DESIGN FOR C3 SYSTEMS

William Perrizo

Computer Science; North Dakota State University

ABSTRACT

Battle management information systems of the future must involve small elements which are physically dispersed and functionally distributed. The underlying database must respond quickly to updates and queries, and must be able to sustain losses of elements without loss of logical data. A distributed information system design for Future Tactical Battle Management Systems is described which would meet these requirements. This report of the design project involves the following areas: replication and distribution, concurrency control and query processing.

INTRODUCTION

Tactical command, control and communication in battle management of the future must involve modular elements which are physically dispersed and functionally distributed and the command and control functions must be moved from the present largely manual operation to a system with automated decision support. [12],[13],[7] Preliminary survivability studies support these conclusions. [5],[11] The distributed battle management information system design described in this paper is intended to meet these requirements. The project is described in separate segments: data modeling, data replication and distribution, initialization and recovery, concurrency control, commitment, and query processing.

DATA MODELING

Data modeling is the process of representing data, data relationships, and data manipulation operations in a systematic way. Structuring database records by considering data item usage can yield substantial efficiencies in the database system. Physical data modeling techniques such as mathematical clustering, iterative grouping, and hierarchical aggregation can be used to determine optimal record, segment, file, and data set structuring as well as efficient access paths. [6]

REPLICATION AND DISTRIBUTION METHODS

Efficient automatic methods for replicating and distributing data are needed to support C3 functions which are physically dispersed and functionally distributed. In this paper a backup chain for a site is an ordering of all other sites in the system, where the first site in the chain is the primary data backup site, the second is the secondary backup site, etc. A replication configuration is a chain of backup sites for each site in the system. [10] When it is deemed unnecessary to have a lengthy backup chains, the full chain can be truncated to an appropriate length. Several sites can be designated as co-primary backups (co-secondary backups, etc.). To allow for this generality, we consider the chain of backups for site, a, in a system with sites, N={a,b,...} to be a sequence of subsets of N, N(1,a) (primary), N(2,a) (secondary),...

Replication configurations can be represented using product functions. Given sets N and M, N*M will represent the subset of the Cartesian product with the diagonal removed. For example, if N = {a,b} and M = {a,c}, then N*M = { (a,c) (b,a) (b,c) }. A replication configuration for a system with sites N={a,b, ...} can be represented as a function, f, from N*N to the positive real numbers, R+, in the following way. A function f:N*N to R+ represents a replication configuration in which for each site, a, N(1,a), N(2,a), N(3,a), ... is the data backup chain for a, where N(1,a) = {b in N ¦ f(a,b)= min {f({a}*N)} }, N(2,a) = {b in N ¦ f(a,b) = min {f({a}*(N-N(1,a))) }, etc.

A function, f, canonically represents a replication configuration if f assigns the value 1 to all primary backups, the number 2 to all secondary backups, etc. It is easily shown that these canonical functions characterize all replication configurations in the sense that every replication configuration is represented by exactly one canonical function and each canonical function represents a exactly one replication configuration.

Some of the configurations which have been considered are:

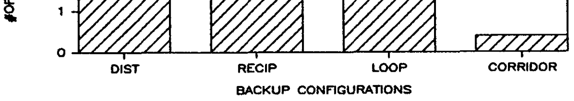
The PHYSICAL DISTANCE configuration (the function is f(a,b)=distance (a,b)), is robust under site moves and the backup loads are evenly distributed among the sites (the expected load is 1 and the maximum is 5). The main disadvantage of this method is its' vulnerability to corridor or area site losses.

The RECIPROCAL OF DISTANCE configuration (f(a,b) = 1/distance) is immunity to area-based site losses and robustness under element movement. The disadvantages include increased loads on the communication resources and vulnerability to area-based site losses.

The CLOSED LOOP configuration (N = {a(1), a(2), ...}, f(a(i), a(j)) = (j-i) mod(n)) is tuneable via the enumeration choice and distributes backup loads evenly (all sites backup one other site). The disadvantages include vulnerability to area-based site losses.

LOSS-PATTERN based configurations (linear corridors, radial propagation, curvilinear propagation, etc.) are designed to provide maximum protection from logical data loss assuming a particular probable pattern of site loss. For example, in the linear corridors method, logical data protection from multiple-site losses which are likely to occur along straight line paths (corridors) is provided.

Experiments conducted to study the question of logical data survivability (at least one surviving copy) under multiple site failures showed the following results:



INITIALIZATION AND RECOVERY

Sites will undergo initial connection to the distributed database followed by one or more subsequent separations from it. For both initialization and recovery, algorithms are needed to bring the resident database up to date while avoiding unnecessary data transmissions and rewrites. Local fault recovery will also be important. Most present recovery schemes involve either transaction logging and data archiving or database duplexing (keeping two identical copies of each local database).

CONCURRENCY CONTROL

Concurrency control in future battle management database systems must provide protection from lost updates and inconsistencies while allowing significantly higher throughput than is required in most other systems.

Most database systems use some form of locking for concurrency control. We assume each user transaction is handled by a transaction manager (TM) and locks are requested by transaction managers from a lock manager (LM). A TM owning a lock, releases it when finished. We assume two phase locking is used for synchronizing this activity. [3] Two phase locking (2PL) requires that all locks be acquired before any are released.

A common application in battle management would involve the generation of composite or summary information using many items of raw data. Such a transaction would require locks on a set of lockable units but in no particular order (set of requests rather than a sequence). Much time can be lost if a lock is requested for a unit which is not available, forcing the transaction to become blocked (execution suspended) waiting for that lock while other needed locks are available. Since much of the data is time-sensitive in nature, these delays would be intolerable. Some method which would allow transactions to discover which data units are locked without risking blocking delays is needed. Several such methods are proposed below.

The standard lock manager (herein called LM0) accepts lock requests from transaction managers (TMs), queues requests to locked units, responds to the request only when lock becomes available to the requesting TM. Thus, with LM0, TMs are blocked (from further processing) while waiting for LM0 to respond and the lock comes with the response.

LM1 is a new lock management technique which has two lock request entry points: one blocking and one non-blocking. The blocking request is as in LM0. The non-blocking request returns True or False, depending on whether the lock is presently available.

LM2 goes one step further. If the lock is not available, the transaction is entered into its queue. The following is a precise description of LM2.

```
LM2(Lock_ID, Level, Trans_ID): Boolean;
   If Lock_ID locked by Trans_ID or
   available then Begin;
       Record lock for Trans_ID;
       LM2 := True
   Else Begin;
       If trans_ID not in Lock_ID queue
       then begin;
           place Trans_ID in Lock_ID queue;
           LM2 := False
```

LM3 goes yet another step and can be given a list of requests and instructed to unblock the transaction and return the lockable unit's ID when any of the lockable units become available.

A transaction manager (TM) is an entity that interprets high-level queries by issuing lower-level requests, including negotiations with the lock manager. The standard transaction manager (TM0) below show the usual way a transaction manager deals with a set request:

```
TM0:
  For i := 0 to n do begin;
     LM0(D(i), Level, Trans_ID);
     Process D(i)
```

It is possible to propose many TMs that use LM1 and LM2. One approach would be to make no blocking requests at all, but to make non-blocking requests for each unprocessed unit in turn until all have been processed. This would involve busy waiting: it never blocks but sometimes will find no locks available; the wait involves successive unsuccessful calls on the lock manager. A busy deadlock is also possible: T1 holds a lock on D1 and requests one on D2; T2 holds a lock on D2 and requests one on D1; each pesters the lock manager. There is no guarantee of successful completion with unlimited non-blocking requests. There must be an upper limit on the number of non-blocking requests made between blocking requests.

TM1 makes blocking requests only when an entire pass through the lockable units with non-blocking requests finds none available.

TM2 below makes one pass through the list of lockable units (LUs) using non-blocking requests and then arbitrarily selects an unprocessed LU to wait for. These two processes are alternated until no LUs remain unprocessed. It differs from TM1 in that it never makes multiple consecutive non-blocking passes through the list before making a blocking request.

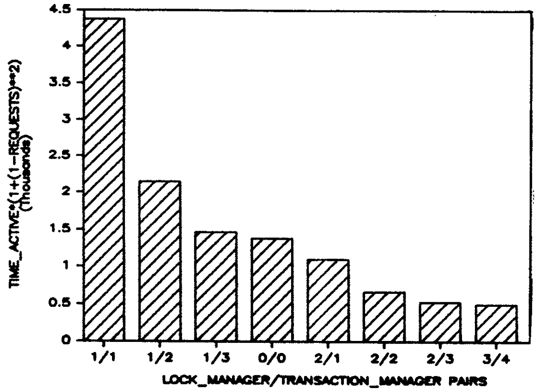
Cutting down on the number of non-blocking requests relative to the blocking requests seems to hurt performance. TM3 is proposed to improve that ratio:

```
TM3:
 Process LU-list using non-blocking requests
 Process LU-list using blocking requests
```

LM3 would be used with a simple transaction manager, TM4:

```
TM4:
 Process LU-list using non-blocking requests
 While LUs remain, wait for next LU.
```

The graph below contains average results of several runs of a concurrency control simulation program for several transaction types (different numbers of lockable units and an activity density levels).



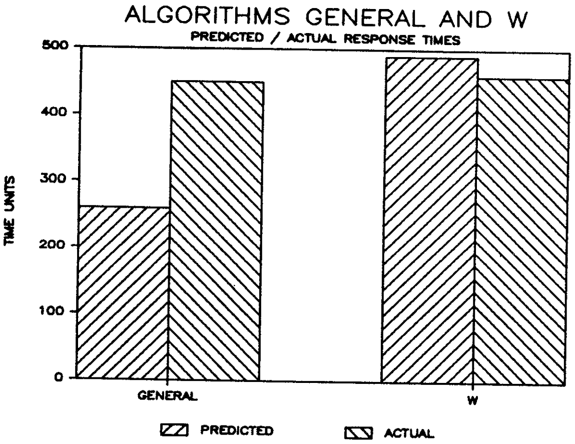
TM4/LM3 achieved the best performance in the simulation, but might prove difficult to implement. TM3/LM2 performed almost as well as TM4/LM3 and might constitute a good practical choice. The system might be designed to use different transaction managers in different circumstances. TM0 might be used to reduce the CPU time consumed in lock requests when there is little database activity, and a different manager used to improve response time when activity is heavier.

QUERY PROCESSING

Distributed query processing algorithms translate a user query into a strategy (schedule of transmission and processing activities) for answering the query. The choice of query processing algorithm is an important design decision. A major difference among algorithms is the method of estimating intermediate relation sizes. Algorithm-General [1] is a static (strategy determined at compile) query processing algorithm which produces optimal strategies assuming the size of a relation changes according to selectivity theory (discussed below). Several other algorithms make the same assumptions. [4],[8] Selectivity theory assumes data is uniformly distributed within each attribute and the distributions of data in different attributes are independent. The point of view in these algorithms is distinctly average-case. The algorithms in [2],[14] involve successive decompositions. A survey of query processing algorithms can be found in. [15]

Algorithm-W is a static query processing algorithm which produces optimal strategies under worst-case data distributions assumptions. [9] Algorithm-W performs well as a general purpose distributed query processing algorithm. It is simple (linear complexity in the number of relations). It is robust (no distribution is favored). It also provides least upper bound response times for the strategies generated (actual response time will be no larger than this figure and will equal it for one database state). This fact makes Algorithm-W useful in time-critical environments such as battle management.

A simulation study was done to compare predicted and actual response times for Algorithm-General which takes an average case approach and Algorithm-W which assumes data distributions are such that little data reduction is possible. The simulation shows that the response time predictions produced by average case algorithms can be very inaccurate (prediction much lower than actual response time). Sample data for the study was randomly generated. A sufficient number of data sets were created and run as input to the programs to allow for valid statistical analysis.



A new query processing algorithm is being developed which, like General and W, employs preliminary semijoins to eliminate most or all data records which are unnecessary to answer the query. This new algorithm also provides a tuning parameter so that the particular semijoin strategy employed can be varied between average and worst-case situations from query to query and over time as data distributions change.

The central idea of the new algorithm is to construct a pattern of preliminary data volume reducing semijoins prior to transmitting required relations to the querying site. This is done by constructing a schedule of the least costly semijoins which can be initiated upon completion of a previously initiated semijoin and to test delivery of each relation to the result site for optimality.

When estimates are to be used for timing in a real time environment, the frequent discrepancy between estimated and actual times for average-case algorithms is intolerable. On the other hand, if the estimates are not needed, the average-case algorithms appear to be good choices since they produces strategies which show better actual response times. The new algorithm allows for either average-case strategies or worst-case depending on the the query by varying a tuning parameter.

COMMITMENT

The fundamental function of a database management system is to carry out transactions. Transactions must be atomic units of work in the sense that they are the smallest independent units of activity making up database applications. Transactions are all-or-nothing propositions. Their effect on the system should be as if they executed either in tota: or else not at all. Transactions involving more than one resource manager (such as high-command information dissemination) require <u>two phase commitment</u> procedures. In the first phase each resource manager must report a <u>ready to commit</u> message to the transaction coordinator. If all resource managers report <u>yes</u> then in the second phase, the coordinator issues a <u>commit</u> command; otherwise the coordinator issues a <u>rollback</u> command. Due to the stringent time constraints on much of the battle management data, commitment procedures will need to be very efficient.

<u>BIBLIOGRAPHY</u>

(1) P. Apers, A. Hevner & S. B. Yao; "Optimization algorithms for distributed queries"; IEEE-TSE V9:1, 1/1983.

(2) P. A. Bernstein and D. W. Chiu; "Using semijoins to solve relational queries"; Journal of ACM, V. 28, No. 1; Jan. 1981.

(3) Eswaran, J. N. et al; "The notion of consistency and predicate locks in a database system"; CACM; 19:11; 11/1976.

(4) Hevner, A.; Data Allocation & Retrieval in Dist. DBMS; <u>Advances in Database Management</u> , V2; Heyden: 1983.

(5) Kasputys, J.; TACS-2000 survivability evaluator; Mitre Corp. MTR8790; 11/1982.

(6) March, S.; "Techniques for structuring database records"; ACM Surveys; 3/1983.

(7) Morris, Maj. J. K.; "TACS 2000 concepts & technology"; AF ESD TR-81-140; 1981.

(8) Perrizo, W.; "A method for processing dist. queries"; IEEE-TSE 10:4; 7/1984.

(9) Perrizo, W.; "A Distributed Query Processing Algorithm Yielding a Least Upper Bound Response Time Strategy"; IEEE Conf. on Supercomputing, 12/1985.

(10) Perrizo, W.; "Data Distribution Methods for Replicated Systems"; IEEE Phoenix Conf. on Computers & Comm.; 3/1986.

(11) Perrizo, W. & D. Varvel; "Future Tactical Air Control System Database Design"; 1984 USAF-SCEEE Report; 8/1984.

(12) 21st Century Tactical Command & Control Study; HQ AFSC/XRK; Aug. 1985.

(13) Wech, O.; "Sound Track for TACS-2000 Motion Picture"; ESD/XR; 1982.

(14) E. Wong & K. Yousefi; "Decomposition - a strategy for query processing"; ACM-TODS; V 1:3; 9/1976.

(15) C.T. Yu & C.C. Chang; "Distributed query processing"; ACM Comp. Surveys 12:4 12/84.

A FAST CONVERGING REAL-TIME ADAPTIVE NOISE CANCELLER

Mohamed El-Sharkawy Maurice Aburdene Arvind Betrabet

Electrical Engineering Department
Bucknell University
Lewisburg, Pennsylvania 17837

Abstract

This paper introduces a real-time hyperstable adaptive noise canceller which uses a new complete hyperstable adaptive recursive algorithm. The proposed algorithm produces a fast converging rate without the need for a priori information about the unknown transmission channel. Rapid convergence is achieved by using a double stage adaptive algorithm to estimate the unknown signal to satisfy the strict positive real condition. Simulation and real-time results are presented.

Introduction

Adaptive noise cancellers have applications in radar, speech processing and sonar signal processing. Most of the early work reported on adaptive noise cancellers concentrated on using the nonrecursive algorithm. This is due to the simplicity in the realization of the noise canceller and the convergence properties of the algorithm associated with it, namely the least mean square algorithm (LMS) [1]. However, adaptive recursive noise cancellers are used when the desired canceller can be more economically modeled with poles and zeroes than with the all zeros form of the nonrecursive canceller. Maintaining stability during the adaptive phase becomes an important consideration due to the presence of poles in the recursive structure. A new generation of adaptive noise cancellers alleviate this problem by using either the simple hyprecursive filter algorithm (SHARF) [2] [3], or the modified hyperstable adaptive recursive filter algorithm (MHARF) [4]. The SHARF algorithm, requires the design of a smoothing filter which depends on a significant a priori information regarding the filter. By making the smoothing filter time-varying, MHARF has overcome this major difficulty although it constrains the roots of the smoothing filter to remain within the unit circle. This leads to stability problems similar to those for the SHARF algorithm. Furthermore, the convergence of most noise cancellers depend on proper choice of a convergence factor. Given a priori knowledge of the input statistics, the convergence of the LMS algorithm can be guaranteed for a limited range of the convergence factor in nonrecursive noise cancellers. No similar results exist for recursive noise cancellers. Although a limited region of convergence can be found for algorithms like SHARF, the selection of this region requires a bound on the initial parameters and the output error which are not commonly known. The proposed algorithm satisfies the strict positive real condition without the need for a priori information about the unknown transmission channel and it has a fast convergence rate.

The Algorithm

A general configuration of this double stage adaptive algorithm appears in Figure 1.

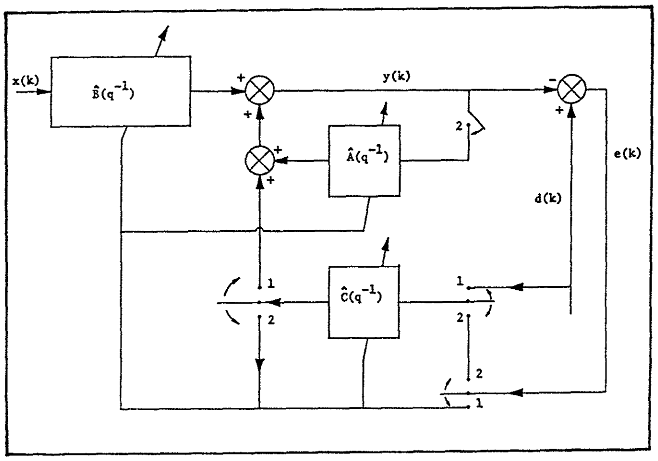


Figure 1.
Complete Hyperstable Adaptive Recursive Filter

(1) In the first stage (path one) the algorithm is used to estimate the parameters of the smoothing filter. The output e_1 is then used by the adaptation algorithm to adjust the parameters of the smoothing filter to have e_1 converge to zero as time increases. In this stage, the algorithm behaves as a recursive-like algorithm. It is based on splitting a recursive filter into two stable nonrecursive filters to estimate the parameters of the smoothing filter.

(2) In the second stage (path two), the algorithm behaves as a recursive filter. The parameters of the smoothing filter obtained in the first stage satisfy the strict positive real condition. The output error e_2 (k) and the augmented error $v_2(k)$ are guaranteed to converge to zero as time increases thus insuring the global convergence of the recursive filter. Simulation results have shown that the rate of convergence of the proposed algorithm is much faster than the SHARF algorithm.

The desired response d(k) is assumed to be stable and is given by the following nth order recursive moving average model (Figure 1)

$$d(k) = \sum_{i=1}^{n} a_i \; d(k - i) + \sum_{j=0}^{m=n-1} b_j \; x(k - j) \qquad (1)$$

where x(k) is the measurable input, a_i and b_j are unknown parameters.

In the first stage, the parameters of the smoothing filter c_i are estimated according to

$$\hat{c}_i(k) = \hat{c}_i(k - 1) + \sigma_i \; d(k - i) \; e_1(k) \qquad i=1,\ldots,n \qquad (2)$$

where σ_i are positive convergence parameters; $e_1(k)$ represents the error between the desired response d(k) and the predicted output $y_1(k)$. These can be written as

$$y_1(k) = \sum_{i=1}^{n} \hat{c}_i(k) \; d(k - i) + \sum_{j=0}^{m} \hat{b}_j(k) \; x(k - j) \qquad (3)$$

and

$$e_1(k) = d(k) - y_1(k)$$

$$= \sum_{i=1}^{n} \tilde{c}_i(k) \; d\,(k - i) + \sum_{j=0}^{m} \tilde{b}_j(k) \; x(k - j) \quad (4)$$

where

$$\tilde{c}_i(k) = a_i - \hat{c}_i(k)$$

$$\tilde{b}_j(k) = b_j - \hat{b}_j(k)$$

$\hat{b}_j(k)$ is an estimate of the unknown parameter b_j which can be updated according to

$$\hat{b}_j(k) = \hat{b}_j(k - 1) + \gamma_j \; x(k - j) \; e_1(k) \qquad (5)$$

$$j = 0,\ldots,m$$

where γ_j are positive convergence parameters.

Equation (4) can be written as

$$e_1(k) = \delta_1(k)^T \psi_1(k) \qquad (6)$$

where

$$\delta_1(k)^T = [\tilde{c}_1(k),\ldots, \tilde{c}_n(k), \tilde{b}_0(k),\ldots, \tilde{b}_m(k)]$$

and

$$\psi_1(k)^T = [d(k - 1),\ldots, d(k - n), x(k),\ldots, x(k - m)]$$

from (2) and (5)

$$\delta_1(k) = \delta_1(k - 1) - \Gamma \; \psi_1(k) \; e_1(k) \qquad (7)$$

where

Theorem:

If the parameter vector $\delta_1(k)$ is updated according to (7), then the error $e_1(k)$ converges to zero as time increases.

Proof:

The following proof is a special case of the general case discussed in [5], [6] where the feedforward path is unity (Figure 2). According to the hyperstability theorem [5], the system of Figure 2 will be hyperstable, that is, $\lim_{k\to\infty} e_1(k) = 0$, if the following inequality is satisfied for all $k_1 \geqslant 0$

$$\sum_{k=1}^{k_1} e_1^2(k) \leqslant \rho_0^2 \qquad (8)$$

where ρ_0^2 is a non-negative constant depending on the the initial conditions.

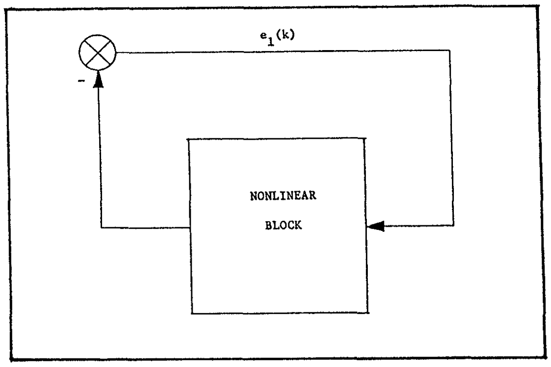


Figure 2.
Equivalent Free Feedback System

Combining (6) and (7), the error $e_1(k)$ can be rewritten as

$$e_1(k) = \delta_1(k)^T \; \psi_1(k)$$

$$= [\delta_1(k - 1) - \Gamma \; \psi_1(k) \; e_1(k)]^T \; \psi_1(k)$$

$$= [\delta_1(k - 1) - \frac{1}{2} \Gamma\psi_1(k) \; e_1(k)]^T \; \psi_1(k)$$

$$- \frac{1}{2} \psi_1(k)^T \Gamma\psi_1(k) \; e_1(k)$$

$$= u_1(k) + u_2(k) \qquad (9)$$

where

$$u_1(k) = [\delta_1(k - 1) - \frac{1}{2} \Gamma \; \psi_1(k) \; e_1(k)]^T \; \psi_1(k)$$

$$u_2(k) = - \frac{1}{2} \; \psi_1(k)^T \; \Gamma\psi_1(k) \; e_1(k)$$

Therefore, in order for the condition (8) to be satisfied, it is sufficient to verify the following inequalities

$$\sum_{k=1}^{k_1} u_1(k)e_1(k) \leqslant \rho_0^2 \qquad (10)$$

$$\sum_{k=1}^{k_1} u_2(k)\ e_1(k) \leqslant 0 \qquad (11)$$

To verify the first inequality, the values of $[\delta_1(k) + \delta_1(k-1)]$ and $\psi_1(k)\ e_1(k)$ are first calculated using equation (7) as follows

$$\delta_1(k) + \delta_1(k-1) = 2[\delta_1(k-1) - \frac{1}{2} \Gamma\ \psi_1(k)\ e_1(k)] \qquad (12)$$

$$\psi_1(k)\ e_1(k) = -\ \Gamma^{-1}[\delta_1(k) - \delta_1(k-1)] \qquad (13)$$

Thus,

$$\sum_{k=1}^{k_1} u_1(k)\ e_1(k)$$

$$= \sum_{k=1}^{k_1} [\ \delta_1(k-1) - \frac{1}{2} \Gamma\ \psi_1(k)\ e_1(k)]^T\ \psi_1(k)\ e_1\ (k)$$

$$= - \frac{1}{2} \sum_{k=1}^{k_1} [\delta_1(k) + \delta_1(k-1)]^T\ \Gamma^{-1}[\delta_1(k) - \delta_1(k-1)]$$

$$= - \frac{1}{2} \sum_{k=1}^{k_1} [\delta_1(k)^T\ \Gamma^{-1}\ \delta_1(k) - \delta_1(k-1)^T\ \Gamma^{-1}\ \delta_1(k-1)]$$

$$= - \frac{1}{2} [\delta_1(k_1)^T\ \Gamma^{-1}\ \delta_1(k_1) - \delta_1(0)^T\ \Gamma^{-1}\ \delta_1(0)]$$

$$\leqslant \frac{1}{2}\ \delta_1(0)^T\ \Gamma^{-1}\ \delta_1(0) = \rho_0^2 \qquad (14)$$

Finally, the second inequality can be verified as follows

$$\sum_{k=1}^{k_1} u_2(k)\ e_1(k) = \qquad (15)$$

$$- \frac{1}{2} \sum_{k=1}^{k_1} \psi_1(k)^T\ \Gamma\psi_1(k)\ e_1{}^2(k) \leqslant 0$$

By assuming that the information vector $\psi_1(k)$ is sufficiently rich, the global convergence of the smoothing filter parameters can be guaranteed.

In the second stage, the parameters of the recursive filter can be estimated using the smoothing filter obtained in the first stage. The input of the smoothing filter is the error between the desired response d(k) and the predicted output $y_2(k)$. $y_2(k)$ can be written as

$$y_2(k) = \sum_{i=1}^{n} \hat{a}_i(k)\ y_2(k-i) + \sum_{j=0}^{m} \hat{b}_j(k)\ x(k-j) \qquad (16)$$

Then, $e_2(k) = d(k) - y_2(k)$

$$= \sum_{i=1}^{n} a_i\ e_2(k-i) + \sum_{i=1}^{n} \tilde{a}_i(k)\ y_2(k-i)$$

$$+ \sum_{j=0}^{m} \tilde{b}_j(k)\ x(k-j) \qquad (17)$$

where

$$\tilde{a}_i(k) = a_i - \hat{a}_i(k)$$

$$\tilde{b}_j(k) = b_j - \hat{b}_j(k)$$

The output of the smoothing filter is the augmented error v(k) which can be written as a moving average of the output error $e_2(k)$;

$$v(k) = e_2(k) - \sum_{i=1}^{n} \hat{c}_i\ e_2(k-i)$$

$$= \hat{C}(q^{-1})\ e_2(k)$$

where $\hat{C}(q^{-1}) = 1-\hat{c}_1\ q^{-1} \ldots - \hat{c}_n q^{-n}$ (18)

Combining equations (17) and (18), the augmented error v(k) can be rewritten as

$$v(k) = \delta_2(k)^T\ \psi_2(k) \qquad (19)$$

where

$$\delta_2(k)^T = [\tilde{a}_1(k), \ldots, \tilde{a}_n(K), \tilde{b}_0(k), \ldots, \tilde{b}m(k)]$$

$$\psi_2(k)^T = [y_2(k), \ldots, y_2(k-m) \ldots, x(k), \ldots, x(k-m)]$$

The augmented error v(k) is then used to update the parameters $\hat{a}_i$ and $\hat{b}_j$ according to

$$\hat{a}_i(k) = \hat{a}_i(k-1) + \mu_i \, y_2(k-i) \, v(k) \quad i=1,\ldots,n \qquad (20)$$

$$\hat{b}_j(k) = \hat{b}_j(k-1) + \rho_j \, x(k-j) \, v(k) \quad j=0,\ldots,m \qquad (21)$$

where μ_i and ρ_j are positive convergence parameters.

Equations (20) and (21) can be written in a vector form as follows

$$\delta_2(k) = \delta_2(k-1) - \Gamma \, \psi_2(k) \, v(k) \qquad (22)$$

where $\Gamma = \text{diag}\,[\mu_1, \ldots, \mu_n, \rho_0, \ldots, \rho_m]$

By updating the parameters of the recursive filter as in equation (22), the augmented error v(k) and the output error $e_2(k)$ can be guaranteed to converge to zero as time increases. The proof is similar to the previous one where v(k) is used instead of $e_1(k)$ and $y_2(k-i)$, instead of d(k-i), are contained in the information vector $\psi_2(k)$. Then, the free feedback system is hyperstable and v(k) converges to zero as time increases. Since $C(q^{-1})$ in equation (18) has all of its roots within the unit circle, the convergence of v(k) to zero implies that the output error $e_2(k)$ converges to zero as time increases.

Adaptive Noise Cancelling

Figure 3 presents a model for adaptive noise cancelling. It is desired to estimate the signal component s(k), measurable in the presence of an additive uncorrelated noise process $n_0(k)$.

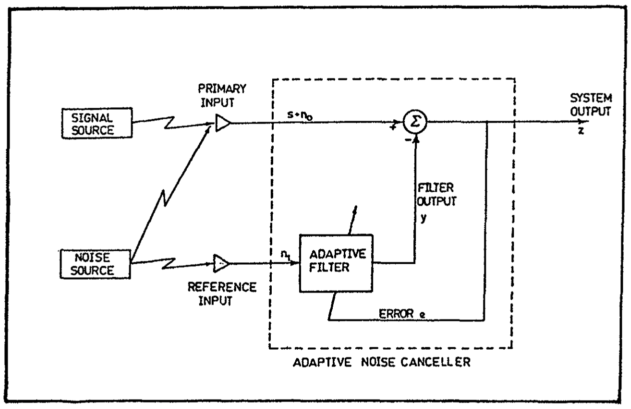


Figure 3. The Adaptive Noise Cancelling Concept

This observed process is called the primary input, d(k), where

$$d(k) = s(k) + n_0(k) \qquad (23)$$

Also, a second sensor is able to provide a reference measurement of a related noise process $n_1(k)$. The relationship between n_0 and n_1 can be described by the following recursive structure.

$$n_0(k) = B(q^{-1}) \, n(k) \qquad (24)$$

and $$n_1(k) = A(q^{-1}) \, n(k) \qquad (25)$$

so $$A(q^{-1}) \, n_0(k) = B(q^{-1}) \, n_1(k) \qquad (26)$$

where $$A(q^{-1}) = 1 - a_1 q^{-1} - \ldots - a_n q^{-n} = 1 - A^*(q^{-1})$$

$$B(q^{-1}) = b_0 + b_1 q^{-1} + \ldots + b_n q^{-n}$$

Where $A(q^{-1})$ and $B(q^{-1})$ are the transfer functions of the transmission channels. Using (23), equation (26) can be written as

$$d(k) = A^*(q^{-1}) \, d(k) + B(q^{-1}) \, n_1(k) - A^*(q^{-1}) \, S(k) + s(k) \qquad (27)$$

This is similar to the proposed algorithm discussed in the previous section. Clearly, if the estimated parameters converge to the true parameters ($\hat{a}_i = a_i$, $\hat{b}_j = b_j$), then e(k) = s(k) as desired, i.e., the output of the adaptive noise canceller is the uncorrupted signal.

Results

To demostrate the convergence speed of the proposed algorithm, we applied it to the following second order example given in [2] and compared the results with SHARF.

$$G(q) = \frac{b_0 + b_1 q^{-1} + b_2 q^{-2}}{1 + a_1 q^{-1} + a_2 q^{-2}}$$

where $a_1 = 1.5588 \quad a_2 = -0.81$

$b_0 = 1.0 \quad b_1 = b_2 \; 0.0$

For small convergence factors which is necessary for SHARF's stability, the present algorithm had a better convergence rate. By increasing the convergence factor to over 0.035, SHARF will diverge whereas this algorithm converges with a higher rate. The output errors of the first and second stages e_1 and e_2 are shown in Figure 4. The convergence parameters were set equal to one. It should be noted that the convergence parameters are not limited to a specific region which dramatically improves the time required by the two stages of this algorithm to converge (error $<10^{-6}$) with respect to SHARF(Figure 5).

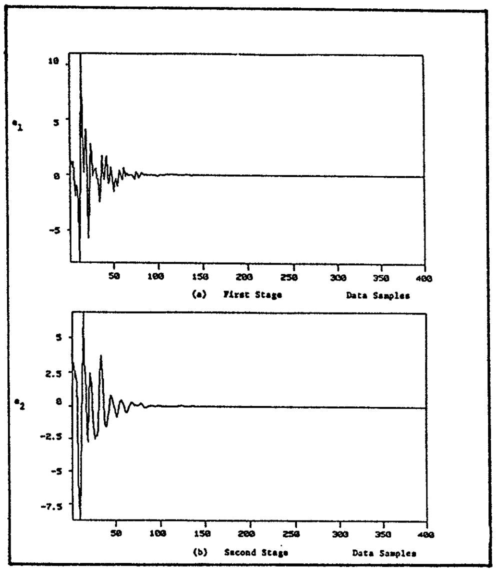


Figure 4. Output and Augmented Errors of a Second Order Complete Hyperstable Recursive Filter.

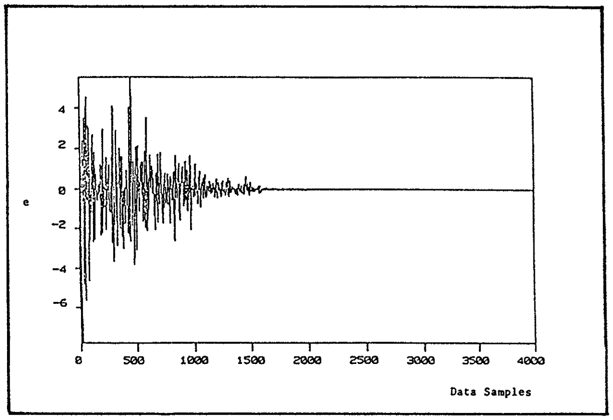


Figure 5. SHARF Algorithm

Similar results were obtained when applied to adaptive noise cancelling problem. This was done using the HP86B computer [7] with an IEEE-488 bus Connecticut Micro Computer Inc. A/D converter. The random noise was generated by a General Radio Company type 1381 random noise generator. In this case, the signal component was a periodic waveform masked by a strong noise component. Figure 6 shows the signal, the corrupted signal and the estimated signal. Note that the noise completely obscures the signal component in the input signal. In the output, however, the input signal can be easily detected.

Conclusions

In this paper, a fast converging adaptive noise canceller was introduced. The noise canceller uses a new complete hyperstable algorithm to guarantee the global convergence of the adaption process by using a double stage adaptive filter. The first stage satisfies the strict passivity condition using a recursive like structure which in turn guarantees the global convergence of the recursive filter in the second stage. In gene the new algorithm compares favorably with the SHARF algorithm. Real time results of the noise canceller were also presented.

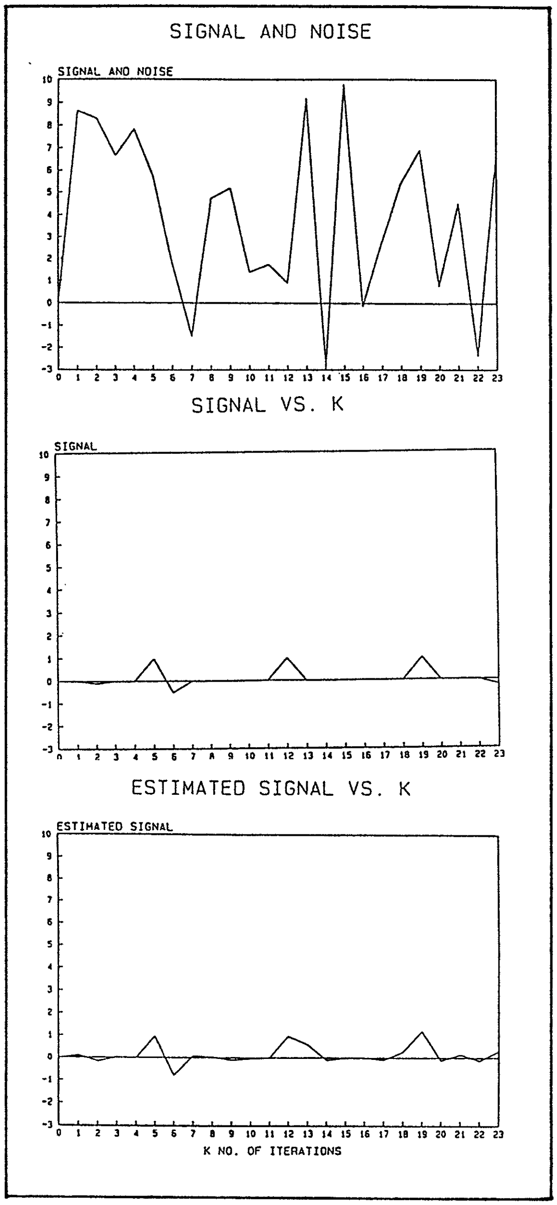


Figure 6.
Signal and Noise

References

[1] B. Widrow et al., "Stationary and Nonstationary Learning Characteristics of the LMS daptive Filters," Proc. IEEE, Vol. 64, pp. 1151-11 August 1976.

[2] M. G. Larimore, et al., "SHARF: An Algorithm for Adapting IIR Digital Filters", IEEE Trans. ASSP, Vol. 28, pp. 428-440, August 1980.

[3] C. R. Johnson, Jr., et al., "SHARF Convergence Properties", IEEE Trans. ASSP, Vol. 29, pp.659-670, June 1981.

[4] D. Parikh, et al., "Convergence Study of the Modified HARF Algorithm", Fourteenth Asilomar Conf. On Circuits, Systems, and Computers, pp. 355-360, Nov. 1980.

[5] V. M. Popov, Hyperstability of Control System, Berlin Springer-Verlag, 1973.

[6] B. O. Anderson, "A Simplified Viewpoint of Hyperstability," IEEE Trans. Automat. Contr., Vol AC-13, pp. 292-294, June 1968.

[7] Hewlett-Packard, Introduction to the HP-86B, March 1983.

Acknowledgement

Partial support for this work was provided by the National Science Foundation's College Science Instrumentation Program through Grant No. 8551136 and Bethlehem Steel's Grants-in-Aid Program.

MCES - A TOOL FOR INTEGRATING TECHNOLOGY INTO NAVY BATTLE GROUPS

Dennis Mensh
Richard Fox

Naval Postgraduate School
Monterey, California 93943-5000

ABSTRACT

The U. S. Navy daily continues to deal with Battle Force Command and Control (C^2, or C^3 or C^3I) issues. These issues arise from improving the performance of existing C^2 elements including technologies that will enable existing as well as future Battle Groups to improve mission performance. These initiatives have generally been tied to the acquisition cycle, viewed broadly to include operation test and evaluation (OT&E) and interoperation issues. This paper presents the results of an application of the Modular Command and Control Structure (MCES) as a tool for integrating electro-optic (EO) technology into Navy Battle Force C^2 systems.

The MCES identifies appropriate measures of performance (MOPS), measures of effectiveness (MOEs), and measures of force effectiveness (MOFEs) for use in evaluating EO technology for inclusion into C^2 systems.

These measures are firmly tied to a C^2 Process Model which is also described. The MCES makes explicit the functionality of the C^2 system being evaluated; consequently, it indicates points of integration for new technologies such as EO. Improved Battle Force performance, resulting from integrated EO technology, is a function of how and where the technology is integrated into the C^2 system. This paper presents the results of the work done to date.

INTRODUCTION

The advent of RADAR stealth technology will cause combat systems that rely totally on RADAR sensing devices to have great difficulty in handling future threats. This situation creates unique opportunities for the synergistic application of other sensor technologies in the combat system. In general, Table (1) shows the advantages and disadvantages of three types of sensor technologies. In particular, the value added to the combat system through the addition of an EO element is a function of how the element is integrated into the combat system. Specifically, the application of the MCES, as a tool, provides some insight into the effect of integrating EO technology into the C^2 structure of the combat system.

PROBLEM STATEMENT

U. S. Naval Battle Forces encountering stealth threats that rely totally on RADAR sensing devices for target detection, will have difficulty in detecting future stealth threats. The stealth threats are characterized by a radar cross-section of .001 m^2. Also, the sea skimming anti-ship cruise missile (ASCM) threats that hide in the sea clutter and are part of the radar multipath problem will be difficult to detect using current RADAR technology.

Figure 1 shows the four different paths that RADAR energy may take in going from the fire control radar antenna to the target and back while tracking sea skimming targets. These four paths will tend to occur simultaneously. The multipath effect will cause the angle of arrival of the strongest return signal at the antenna to move within the angle defined by the direct paths from the antenna to the target and from the antenna to the reflected image of the target below the surface. The antenna will adjust its position so that it points in the direction at which it receives the strongest return signal. The result is that the antenna begins to oscillate wildly as it attempts to track the centroid of the return RADAR energy. The centroid will migrate back and forth between the target and the target's reflected image. The oscillations experienced by the fire control radar can be so severe as to cause the radar to break target track.

RADAR sea clutter is another naturally occurring phenomenon which can affect RADAR system performance. RADAR sea clutter is caused by RADAR energy which reflects off the sea surface and returns to the radar antenna. The strength of this return energy is dependent on factors such as: sea state; wind speed; the length of time and the distance (fetch) over which the wind has been blowing; direction of the waves relative to that of the radar beam; whether the sea is building up or is decreasing; and the presence of contaminants in the water, such as oil. The strength of the return is also dependent on radar system parameters such as frequency, polarization and to the grazing angle relative to the sea surface of the energy path.

RADAR sea clutter will cause the radar system to record a high level of noise from the area under observation. The noise level may be high enough to mask the 2D radar return signal from a target present in the region where the noise originates.

The sea skimming and stealth threat adversely impacts the Battle Force. The impact is felt by the individual shipboard combat systems that are currently ill equipped to handle these threats. Electro-Optic Technology which is immune to both the RADAR stealth threat, and the sea clutter and multipath problems of sea-skimming threats, provides a possible solution. In addition this technology, as presented in Table (1), provides high resolution and elevation data to the combat system. Combat systems receiving RADAR data (range and bearing data (2D RADAR) and elevation data (3D RADAR)) will now be able to do sensor correlation providing synergistic target track data. It appears that the correlated sensor data can improve combat system target capability in adverse weather conditions, in the multipath and sea clutter environments.

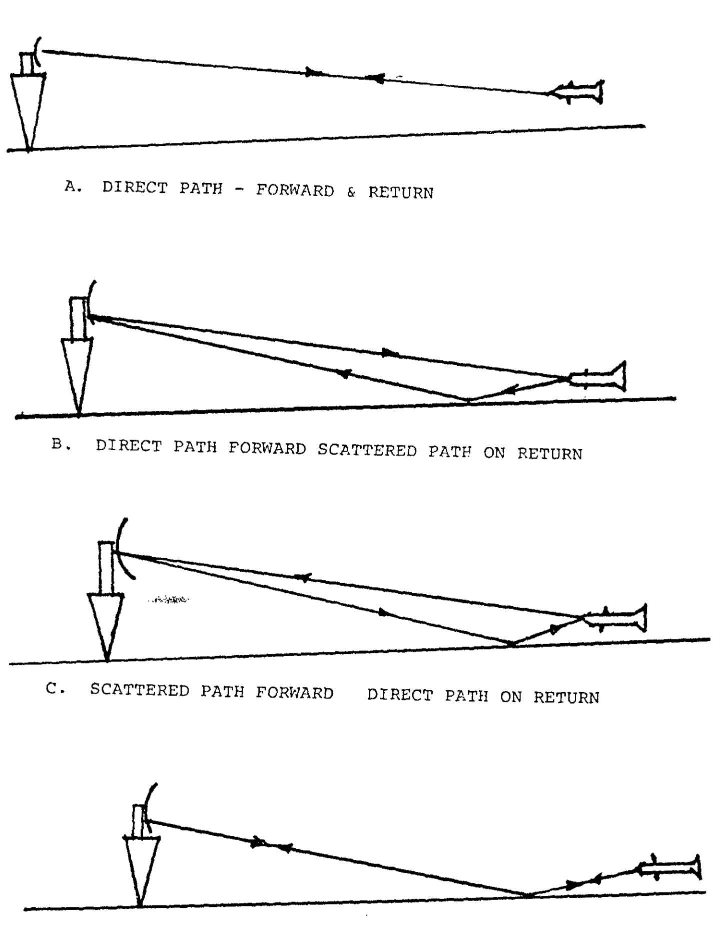
MCES

The C^2 system which these two groups have in common is composed of the following components:

(A) Physical entities such as sensors (detectors), computers ....

SENSOR TYPE:	RADAR	ESM	IR
ADVANTAGES:	ALL WEATHER	PASSIVE/COVERT	PASSIVE/COVERT
	PRECISE POSITION	TARGET ID	HIGH RESOLUTION Az
	LONG RANGE	CHARACTERIZE TARGET	HIGH RESOLUTION El
DISADVANTAGES:	MULTIPATH	LOW RESOLUTION Az	NOT ALL WEATHER
	RF SEA CLUTTER	NO TARGET RANGE	NO TARGET RANGE
	JAMMING	NO TARGET El	NO TARGET SPEED
	STEALTH	EMI	
	EMI		
	ARM		

TABLE 1 SENSOR SYSTEM DATA CHARACTERISTICS



A. DIRECT PATH - FORWARD & RETURN

B. DIRECT PATH FORWARD SCATTERED PATH ON RETURN

C. SCATTERED PATH FORWARD DIRECT PATH ON RETURN

D. SCATTERED PATH FORWARD AND ON RETURN

FIGURE 1 FIRE CONTROL RADAR MULTIPATH

(B) A structure that incorporates concepts of operation, data integration and information flow.

(C) A C2 process that describes what the C^2 system is doing in terms of the functions performed by the C^2 process. These functions include: (]) sense, detect, assess, (2) generate, select plan and direct.

Any methodology that effectively integrates technology into a combat system should meet the following requirement: It should consist of a set of logical steps that define and bound the integration problem. It should also provide insights and valid estimates of numerical measures of EO element performance, combat system effectiveness, and Battle Force effectiveness. The Modular Command and Control Evaluation Structure (MCES) is a tool that meets these requirements. Figure 2 presents a flow chart of the MCES. The detail of the application of the MCES to the EO integration problem is described in the following section.

APPLICATION OF MCES

The first MCES module, applications objective, initializes the analysis process by deriving a clear statement of the problem. Considering the decision makers in the Engineering and Operational communities that have to deal with the operational requirement of defeating the stealth and sea skimming threat, a problem statement might be: Improve the effectiveness of the C^2 system which supports the Battle Force by integrating EO sensors into shipboard combat systems.

After working through this module it became obvious that the EO integration problem transforms itself into architectural issues; where an architecture is the assignment of functions to the organizational structure. The Composite Warfare Coordinator, CWC, organizational structure, as employed by the Battle Force, is shown in Figure 3.

The C^2 system bounding module defines the C^2 system statics. This module bounds the EO integration problem in terms of:

* a subsystem of interest - EO sensor element
* the C^2 boundry of interest - CWC structure
* the force boundary - Blue and Orange forces, etc.
* the environment - Natural command authority; shore based command centers, etc.

Figure 4 presents a detailed subset of the CWC structure. This figure shows an example of the functions of detect, engage and control that are assigned to the CWC structure and performed by the AAWC and ASUWC. The sense function is currently performed using an air/surface surveillance radar. The effectiveness of the C^2 process function can be increased with the addition of an EO sensor element.

It is evident from this analysis that the EO technology can be integrated

* inside the C^2 system boundary in support of the CWC structure
* inside the Force boundary as a sensing element on an individual ship supporting the mission of Blue force.

The C^2 system dynamics are defined functionally by the C^2 process module. It maps these functions (1) into Battle Force missions as derived from operational situations (OPSITS) or scenarios, and (2) into combat system functions that achieve the derived mission objectives. Figure 5 displays the results of the functional mapping.

The EO sensor senses the environment and obtains contact data consisting of a contact signature, an azimuth and an elevation. This data is processed through the combat system functions and at the same time, integrated into the C^2 process functions. It is important to note that these combat system functions represent actual observable events. These events can be quantified using data collected from fleet exercise experiments or computer simulation.

Once the problem is bounded and the C^2 process defined, measures of performance, effectiveness, and force effectiveness can be derived. Figure 6 defines these measures in terms of system boundaries. Based on these definitions, the following measures were defined for the EO/CS system integration problem:

* MOP - Probability of detection given a contact
* MOE - System reaction time from initial target detection to engage
* MOFE- Number of targets killed

Data quantifying these three measures were obtained from computer experiments using the Ship Combat System Simulation.

An anti-Air Warfare Combat System configuration is shown in Figure 7. This system performs detect, engage and control functions using

* Air surveillance elements -
  Radar
  Infrared Search Target Designator (EO)
  Designator (EO)
* Command and Control System elements supporting the CWC structure through the
  AAWC
  Engagement Controller (EC)
  C^2 System Computer (TR. Module)
  Tracking Module
* Engaging Elements
  Missile System
  Rapid Fire Gun System

The elements in Figure 7 are considered nodes in a network connected by links. Information flows through the links causing the nodes to take action.

Two architectures are also shown in the figure. The EO system integrated into the C^2 system; EO system integrated into the fire control system. Attention is focused on the C^2 system.

To complete the example, computer experiments were performed where the detect, engage, and control functions were exercized in an OPSIT containing four sea-skimming, anti-ship cruise missles in sea clutter en vironment (sea state 3). Target radar cross-section in square meters was varied from 10.0, 1.0, 0.1, 0.001. Figure (8) presents a combat system configuration that uses a 2D surveillance radar for target detection. Once detected and firm track established by the control system, the targets are engaged by the missile system.

Preliminary results indicate for radar cross-sections 10.0, 1.0, 0.1 and 0.01, all four targets were detected. No 0.001 m^2 targets were detected. One square meter target was destroyed i.e., the value of the MOFE was one target killed. Three targets hit the ship. As the radar cross-section became less than equal to one square meter, some or all of the combat system functions shown in Figure (5) were not performed.

Figure (9) presents a time line graph for the one target killed. The horizontal axis displays time to

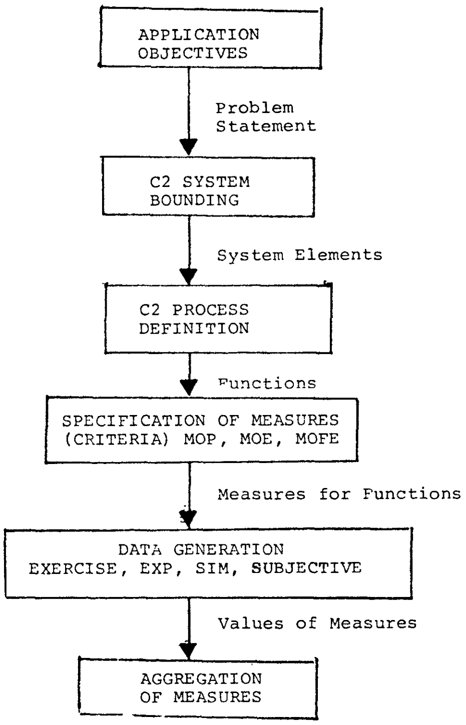


FIGURE 2 MCES MODULES

OTC

AAWC ASUWC ASWC STWC

OTC	=	OFFICER IN TACTICAL COMMAND
AAWC	=	ANTI-AIR WARFARE COORDINATOR
ASUWC	=	ANTI-SURFACE WARFARE COORDINATOR
STWC	=	STRIKE WARFARE COORDINATOR

FIGURE 3 CWC STRUCTURE

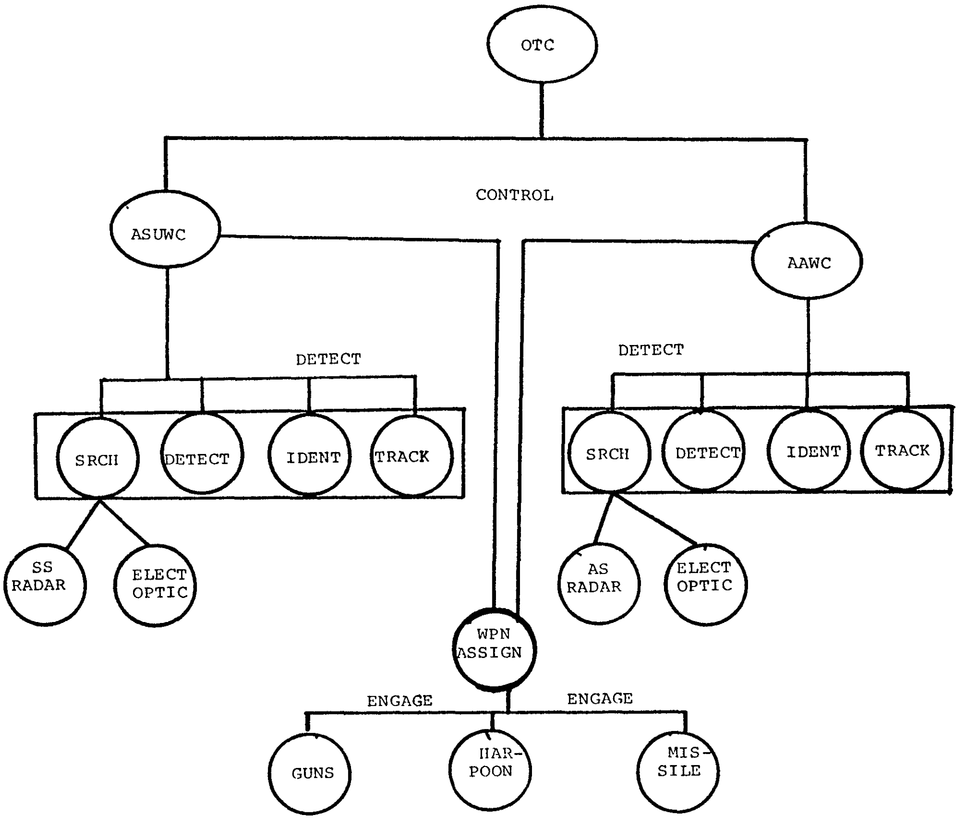
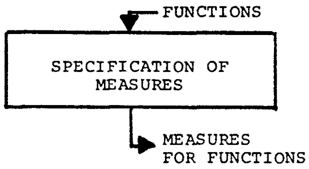


FIGURE 4 EXPANDED CWC STRUCTURE

C2 PROCESS MODEL	COMBAT SYSTEM FUNCTIONS
FUNCTIONS	MEASUREABLE EVENT (OBSERVABLES)
SENSE (DETECT)	INITIAL DETECTION FIRM TRACK
ASSESS	ENGAGEABILITY FIRE CONTROL SOLUTION
GENERATE	DESIGNATE WEAPON ORDER
SELECT	ACQUIRE ORDER ILLUMINATOR LOCK ON
PLAN	WHEN TO LAUNCH SHOOTING DOCTRINE
DIRECT	LAUNCH

FIGURE 5 FUNCTIONAL MAPPING



DEFINITIONS

MEASURE OF PERFORMANCE (MOP)

Measures/Specific Inside the Boundary of the C2 System:

MOP: These are also closely related to inherent parameters (physical and structural) but measure attributes of system behavior (gain throughput, error rate, signal-to-noise ratio).

MEASURE OF EFFECTIVENESS (MOE)

Measures/Specified Outside the Boundary of the C2 System

MOE: Measure of how the C2 system performs its functions within an operational environment (probability of detection, reaction time, number of targets nominated, susceptibility of deception).

MEASURES OF FORCE EFFECTIVENESS (MOFE)

Measures/Specified Outside the Boundary of the Force

MOFE: Measure of how a C2 system and the force (sensors, weapons, C2 system) of which it is a part performs missions

FIGURE 6 SPECIFICATION OF MEASURES

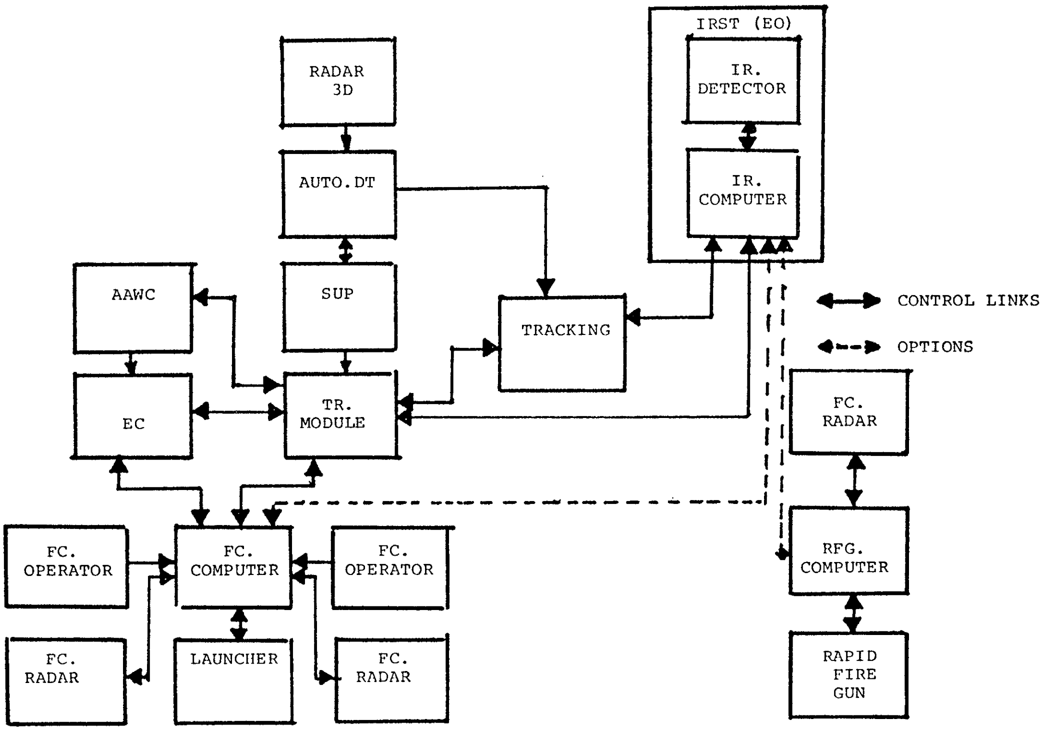


FIGURE 7 ANTIAIR WARFARE COMBAT SYSTEM CONFIGURATION

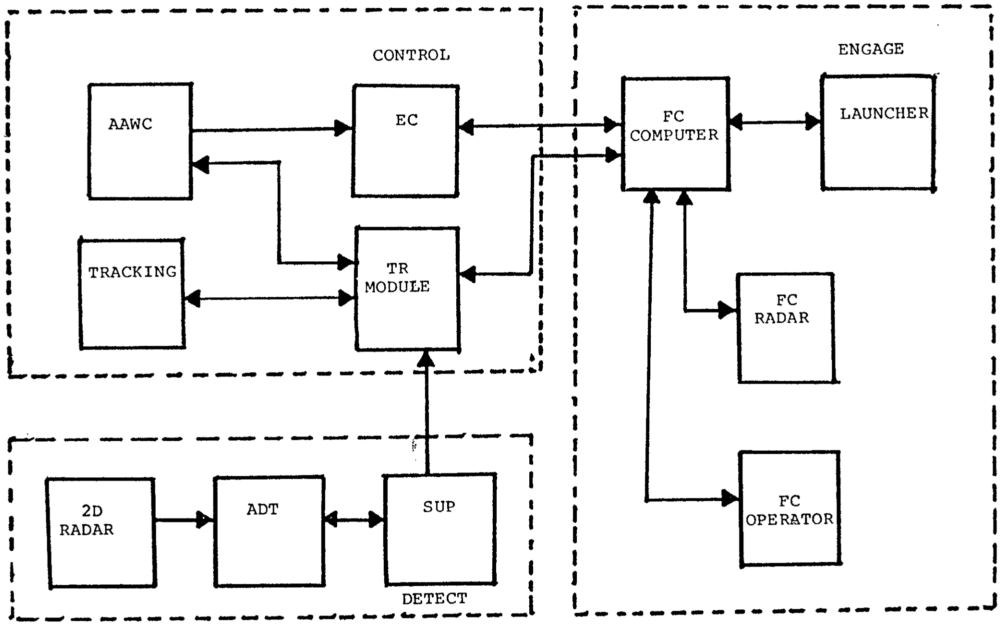


FIGURE 8 COMBAT SYSTEM CONFIGURATIÒN

target impact. The vertical axis displays the combat system functions (observable events) mapped to the C^2 process functions. This data indicates that initial target detection occurred in time for all combat system functions to be performed including missile launch occurring close to ship impact. It appears from the time line data that the MOE (System reaction time from initial detection to engage) needs to be significantly improved (shifted to the right) so that more targets can be engaged without the ship being hit by sea skimming missiles. Specifically, target detections and declarations must occur early enough so that the MOFE significantly increases.

Figure (10) presents a similar combat system configuration that uses a near current generation EO sensor for target surveillance and detection. This configuration was subjected to the same OPSITS. The results of this computer experiment indicated a dramatic improvement.

The data showed that all four contacts were detected and declared as targets far from target impact. The MOFE improved from 1 to 3 targets killed. This improvement is attributed to the EO sensor element. Early target detections improved the response time of the control and engage functions. It appears that for sea skimming threats an EO sensor element improves combat system response time and increases Battle Force effectiveness.

SUMMARY

This paper has described a tool, the Modular Command and Control Evaluation Structure (MCES), that provides a methodology for integrating new technology, i.e. Electro-Optics, the shipboard combat systems. When the engineering and operational community use the MCES together as a tool, the integration of EO technology into the combat system is enhanced. As a direct consequence, the Battle Force is more effective. The MCES bounds the integration effort, defines MOPS and MOFES, and develops alternative configurations. These configurations can be tested. The results presented here show that the configuration that employs EO technology as part of the C^2 system will improve the performance of the Battle Force.

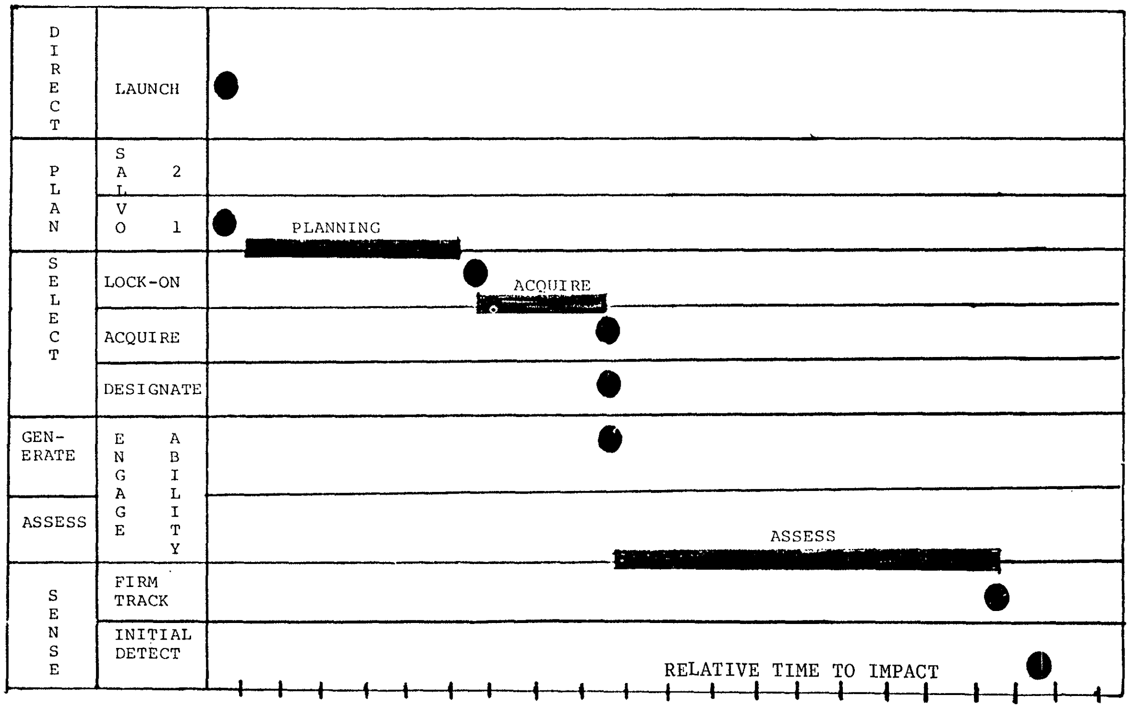


FIGURE 9 MEASURE OF EFFECTIVENESS

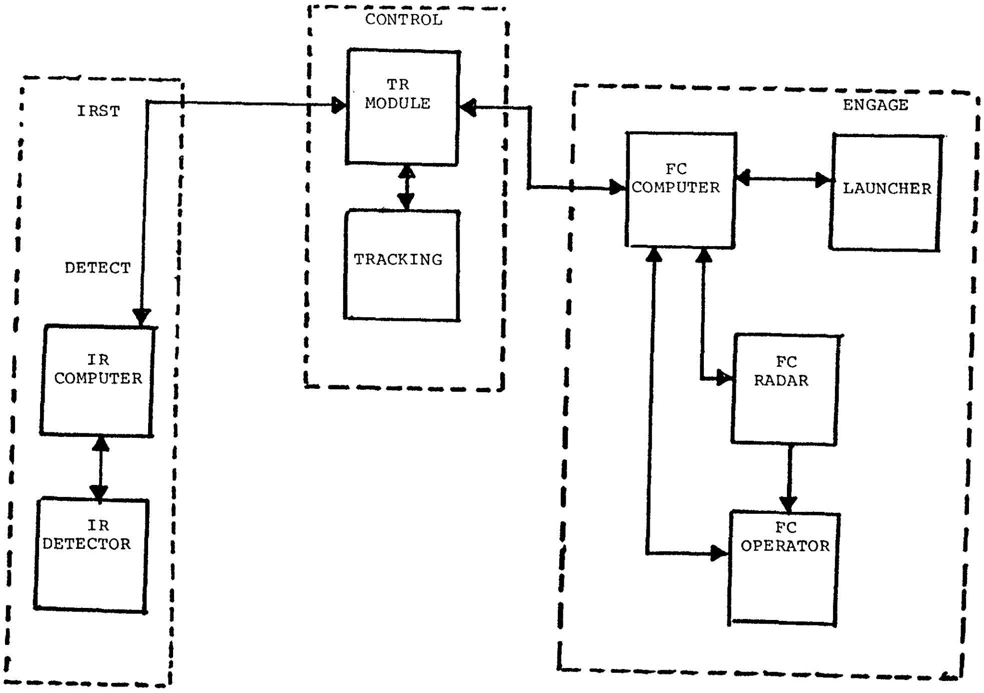


FIGURE 10 COMBAT SYSTEM IRST CONFIGURATION

NONLINEAR NONEQUILIBRIUM STATISTICAL MECHANICS APPROACH TO C^{3}I SYSTEMS

Lester Ingber

National Research Council–NPS Senior Research Associate
and Physics Department and C^3 Group – Code 61IL
Naval Postgraduate School, Monterey, CA 93943

It is proposed to incorporate "intuition" into large complex multivariate nonlinear C^{3}I systems requiring stochastic or probabilistic treatment, i.e., to seek regions of variable–space where more local analytic resources can be optimally allocated. These mathematical techniques have been utilized for a variety of other systems, ranging from neuroscience, to nuclear physics, to financial markets. The experiences gained by detailing each of these systems offers specific insights by which to approach C^{3}I systems.

I. INTRODUCTION

Even without having agreement on just what is C^{3}I, there is widespread criticism that we do not spend enough on C^{3}I relative to what we spend on specific weapons systems. [1] The Eastport Study Group [2] has made this issue its primary concern with regard to the SDI program. There is also an everpresent problem of weighing the political and military aspects between hierarchical and distributed design of C^{3}I, the former being politically desirable and appropriate for deterministic or modestly stochastic operations, and the latter being more appropriate for severely stochastic systems. [3] Future battle management, e.g., as being investigated by the SDI program, certainly must consider distributive adaptive C^{3}I for severely stochastic systems. [2]

In this paper I will outline an interdisciplinary approach that is attempting to piece together a specific coherent C^{3}I model that may yield insights into C^{3}I systems most appropriate for severely stochastic combat operations.

Section II motivates the necessity of formulating "order parameters" relevant to specific situations, which views the physiology (function) of C^{3}I systems as complementary to their anatomy (structure), by outlining a theory of personal combat. [4–7] There can be no pretense that personal combat is equivalent to international combat, but there are some similarities that deserve mention.

Section III outlines a theory I have formulated of mesoscopic and macroscopic brain function, derived from microscopic synaptic chemical–electrical interactions. [8–15] Since many investigators now find it useful to use brain function as a metaphor for other processes they perceive to be present in their own disciplines, it is relevant to discuss the actual brain and the processes by which it performs "Biological Intelligence" (BI). The mathematical formalism used turns out to describe a parallel processing of mesoscopic information in a distributed adaptive system that we know exists, and that we know to be robust under many changes in its internal and external environments. Indeed, in many circumstances, especially those requiring pattern recognition under uncertainty, [16–18] BI is still superior to AI which typically requires a deterministic and hierarchical spine on which to grow tree– and loop–like structures.

These technical methods are quite general, and I also have applied them to nuclear physics – detailing Riemannian contributions to the binding energy of nucleons interacting via exchanges of mesons, [19–22] and to financial markets – defining an approach to explain various phenomena such as leptokurtosis, the biasing of price data. [23] These systems are all quite different in their natures, but they do share a common approach by these methods of nonlinear nonequilibrium statistical mechanics. The nuclear physics system illustrates how patterns of information can be represented by eigenfunctions of the probability distribution. The markets system illustrates how the mesoscopic scale can be formulated phenomenologically, without the luxury of deriving it from a microscopic system as was done for the neocortical system.

Section IV outlines how the mathematics used for BI can be used to develop a distributive adaptive system capable of processing more general types of information relevant to C^{3}I.

Section V discusses work in progress in which we are attempting to use BI to fit data from combat training simulations. We are using these simulations because data is available to fit our theory, and because we can then test our theory by seeing if the dynamic probability distribution we develop can be used successfully in future simulations to enhance the chances of victory. Perhaps these methods will be useful for SDI BM/C^{3}I systems as well.

Section VI discusses how these tools might be used as decision–making aids in a C^{3}I system in real time combat situations.

I emphasize that these methods have not been previously applied to C^3 systems, although they have been tested in other systems. Very approximately, this approach can be considered a nonlinear stochastic generalization of Lanchester theory. [24]

II. ANATOMY VS. PHYSIOLOGY OF C^{3}I

A typical C^{3}I organization–chart–e.g., Sense, Process, Decide, Act, Analysis, Environment, etc.–might well be useful for allocating resources to build a system for combat, or even be useful for developing training methods to keep each component fit and ready for battle. However, especially in this simple example, it is easy to intuit that this outline is not directly useable in actual combat, since in a real–time situation, only a small subset of these parameters, possibly even an entirely new subset aggregated from those given, is of immediate concern to a commander.

Establishing the function (physiology) of C^{3}I systems seems today to be at least as much an art as a science. However, this function is extremely important, as it defines the actual variables, or order parameters, that a commander requires in combat operations.

For example, in the context that there is much to learn for C^{3}I systems from the function of the human brain, there is a counterpart to the three levels of processing in processes of attention. I refer to three levels of attention required in personal combat between highly skilled opponents. As a first approximation, time resources are roughly equivalent to distance between opponents.

At a "far" distance, i.e., beyond the distance at which either side can touch the other within a single movement, there is so much uncertainty as to future possibilities, that the only realistic techniques called upon are strategic feints and "themes" of sparring, often categorized by five elements (earth, air, fire, water, void), to cause and break rhythms in the opponent. [7] This is akin to a gross macroscopic perception of the engagement.

At a "medium" distance, i.e., just within the distance at which either side can strike each other with a single movement, skills required are strategic and tactical feint–defense–attack combinations composed of arhythmic spurts of several techniques, somewhat similar to the "middle" game of chess, with the dimension of time thrown in. [4–7] This is akin to a mesoscopic perception of the engagement, wherein the order parameters are the individual combinatoric phrases rather than their individual techniques.

At a "close" distance, i.e., within the distance at which either side can reach or lunge with elbows and knees, one must function within critical reaction times of very few tenths of a second. At this microscopic perception it is more sheer power and chance than strategy or even tactics, that determines the outcome, as only simple repeated firings of techniques are realistic.

Some other interesting analogies between C^{3}I systems and personal combat can be drawn. In order to be effective within tenths of a seconds against strong opponents, one must train to have distributed control at many stages of the C^{3}I–karate organization. Visual and auditory senses must be trained to receive information in parallel with somatic senses actively seeking information. Imaginary scenarios and forecasts must be made in parallel with decisions being made in real time. The trained body must coordinate itself to perform techniques, just using quite general constraints imposed by these decisions: there are many techniques that might accomplish similar goals, but the choice of technique does not seem to made by one central command center.

Perhaps the most important analogy to stress emphasize, and that I will also stress in my outline of my work in neocortex, is that the concept of "scaling" should be applied to C^{3}I systems to determine the relevant order parameters describing levels of distributed command and control.

III. BIOLOGICAL INTELLIGENCE (BI)

1. Introduction—Theory vs. Model

BI demonstrates the physiology of neocortex. Proper treatment of nonlinearities demonstrates how multiple hypotheses are generated and processed by STM. Similarly, it should be expected that useful decision aids to commanders will require robust C^3I nonlinear models of previous combat operations.

Modern technology has made it possible to detail actual properties of many physical and biological nonlinear nonequilibrium systems, i.e., in contrast to performing otherwise important investigations of (quasi−)linearized approximate models. Typically, the price paid for this detail is that a set of complementary approaches, sometimes mutually exclusive, must be used for particular aspects. [25] C^3I and neocortex present similar challenges.

A series of publications has detailed a statistical mechanics approach to macroscopic regions of neocortex, derived from statistical aggregates of microscopic neurons, i.e., a statistical mechanics of neocortical interactions (SMNI). [8−14] As found necessary for other nonlinear nonequilibrium systems, a mesoscopic scale is sought to develop a Gaussian−Markovian statistics for further macroscopic development. [26,27] This mesoscopic scale is found in the observed physiology of columnar interactions. Long−term−memory (LTM) properties and the duration and capacity of short−term−memory (STM), i.e., the "7±2 rule," have been derived from multiple minima of a nonlinear Lagrangian (time−dependent and space−dependent "cost function"); the alpha frequency and velocity of propagation of columnar information−processing, consistent with observed movements of attention across the visual field, have been derived in linearized ranges within these minima.

Coarse−graining is an important general method of treating nonlinear nonequilibrium statistical systems, e.g., in order to develop Gaussian−Markovian probability distributions. Also, less resources are required to process the coarser variables, which is efficient if that is all that is required for macroscopic function. The theory capable of treating these systems require mathematical tools only developed in the late 1970's, [28−39] including quite general nonlinear nonequilibrium structures into previously linear treatments of Gaussian−Markovian systems. [40]

This theory is geared to explain macroscopic neocortical activity, retaining as much correct description of underlying microscopic synaptic activity as can be carried by modern mathematical physics, which turns out to be sufficient for several important circumstances. Only after this process is completed, are approximate numerical and algebraic methods applied to solve the resulting mathematics. It is at *this* stage that modelling is most useful. The 1980's already have demonstrated that many systems require the use of several complementary algebraic and numerical algorithms to detail several scales of interaction. [25] Neocortex is not unique in requiring several approaches, nor is it unique in requiring it own unique algorithms.

For example, without sufficient mathematical or physical justification, many models assume (quasi−)linear deterministic rate equations—analogous to conserved quadratic "Hamiltonians"—to postulate "average" neurons, thereby neglecting statistical and stochastic background interactions, nonlinearities induced by interactions among neurons, and spatial−temporal statistics of large ensembles of these interacting neurons. In fact, these nonlinearities and statistics are essential mechanisms of STM, [11,13] and possibly of alpha rhythm observed in electroencephalographic (EEG) and magnetoencephalographic (MEG) [41] activity. [12] These results are not obtained by "fitting" theoretical parameters mocking neuronal mechanisms to empirical data. Rather, these results are obtained by taking reasonable synaptic parameters, developing the statistical mechanics of neocortical interactions, and then discovering that indeed they are consistent with the empirical macroscopic data. Other models which have offered plausible brain mechanisms can be processed by this theory, extending their ranges of validity. [8,9]

2. Description of Theory

Microscopic Neurons

When describing the activity of large ensembles of neocortical neurons, each one typically having many thousands of synaptic interactions it is a reasonable assumption that simple algebraic summation of excitatory (E) depolarizations and inhibitory (I) hyperpolarizations at the base of the inner axonal membrane determine the firing depolarization response of a neuron within its absolute and relative refractory periods. [42]

This is straightforwardly mathematically summarized. Within $\tau_j \sim 5-10$ msec, the conditional probability that neuron j fires, given its previous interactions with k neurons, is

$$p_{\sigma_j} \simeq \Gamma\ \Psi$$

$$\simeq \frac{\exp(-\sigma_j F_j)}{\exp F_j + \exp(-F_j)} ,$$

$$F_j - \frac{V_j - \sum_k a_{jk}^{*} v_{jk}}{\left[\pi \sum_{k'} a_{jk'}^{*} (v_{jk'}{}^2 + \phi_{jk'}{}^2)\right]^{1/2}} ,$$

$$a_{jk} = \frac{1}{2} A_{jk} (\sigma_k + 1) + B_{jk} . \tag{1}$$

This is true for Γ Poisson, and for Ψ Poisson or Gaussian. V_j is the axonal depolarization threshold. v_{jk} is the induced synaptic polarization of E or I type at the axon, and ϕ_{jk} is its variance. The efficacy a_{jk}, related to the inverse conductivity across synaptic gaps, is composed of a contribution A_{jk} from the connectivity between neurons which is activated if the impinging k−neuron fires, and a contribution B_{jk} from spontaneous background noise.

Mesoscopic Domains

As is found for most nonequilibrium systems, a mesoscopic scale is required to formulate the statistical mechanics of the microscopic system, from which the macroscopic scale can be developed. [26] Neocortex is particularly interesting in this context in that a clear scale for the mesoscopic system exists, both anatomically (structurally) and physiologically (functionally). "Minicolumns" of about $N \simeq 100$ neurons (about 200 in visual cortex) comprise modular units vertically oriented relative to the warped and convoluted neocortical surface throughout most, if not all, regions of neocortex. [43−47] Clusters of about 100 neurons have been deduced to be reasonable from other considerations as well. [48] The overwhelming majority of neuronal interactions are short−ranged, diverging out via efferent minicolumnar fibers to within ~1 mm, which is the extent of a "macrocolumn" comprising $\sim 10^3$ minicolumns of $N^* \simeq 10^5$ neurons. Macrocolumns also exhibit rather specific information−processing features. This theory has retained the divergence:convergence of minicolumn:macrocolumn efferent:afferent interactions by considering domains of minicolumns as having similar synaptic interactions within the extent of a macrocolumn. This dynamically macrocolumnar−averaged minicolumn is designated in this theory as a "mesocolumn."

This being the empirical situation, it is interesting that $N \simeq 10^2$ is just the right order of magnitude to permit a formal analysis using methods of mathematical physics just developed for statistical systems in the late 1970's. [34,37] N is small enough to permit nearest−neighbor (NN) interactions to be formulated, such that interactions between mesocolumns are small enough to be considered gradient perturbations on otherwise independent mesocolumnar firing states. This is consistent with rather continuous spatial gradient interactions observed among columns, [49] and with the basic hypothesis that nonrandom differentiation of properties among broadly tuned individual neurons coexists with functional columnar averages representing superpositions of patterned information. [50] This is a definite mathematical convenience, else a macrocolumn of $\sim 10^3$ minicolumns would have to be described by a system of minicolumns with up to sixteenth order next−nearest neighbors. Also, N is large enough to permit the derived binomial distribution of afferent minicolumnar firing states to be well approximated by a Gaussian distribution, a luxury not afforded to an "average" neuron even in this otherwise similar physical context. Finally, mesocolumnar interactions are observed to take place via one to several relays of neuronal interactions, so that their time scales are similarly $\tau \simeq 5-10$ msec.

After statistically shaping the microscopic system, the parameters of the mesoscopic system are minicolumnar−averaged synaptic parameters, i.e., reflecting the statistics of millions of synapses with regard to their chemical and electrical properties. Explicit laminar circuitry, and more complicated synaptic interactions, e.g., dependent on all combinations of presynaptic and postsynaptic firings, can be included without loss of detailed analysis. [10]

The mathematical development of mesocolumns establishes a mesoscopic Lagrangian $\underline{L}$, which may be considered as a "cost function." The Einstein summation convention is used for compactness, whereby any index appearing more than once among factors in any term is assumed to be summed over, unless otherwise indicated by vertical bars, e.g., $|G|$.

$$P = \prod_G P^G [M^G (r;t+\tau) | M^{\bar{G}} (r';t)]$$

$$= \sum_{\sigma_j} \delta\left(\sum_{jE} \sigma_j - M^E (r;t+\tau) \right) \delta\left(\sum_{jI} \sigma_j - M^I (r;t+\tau) \right) \prod_j^N p_{\sigma_j}$$

$$\simeq \prod_G (2\pi\tau g^{GG})^{-1/2} \exp(-N\tau \underline{L}^G) \, ,$$

$$P \simeq (2\pi\tau)^{-1/2} g^{1/2} \exp(-N\tau \underline{L}) \, ,$$
$$\underline{L} = (2N)^{-1} (\dot{M}^G - g^G) g_{GG'} (\dot{M}^{G'} - g^{G'}) + M^G J_G / (2N\tau) - \underline{V}' \, ,$$
$$\underline{V}' = \sum_G \underline{V}''^G_{G'} (\rho \nabla M^{G'})^2 \, ,$$
$$g^G = -\tau^{-1} (M^G + N^G \tanh F^G) \, ,$$
$$g^{GG'} = (g_{GG'})^{-1} = \delta_G^{G'} \tau^{-1} N^G \mathrm{sech}^2 F^G$$
$$g = \det(g_{GG'}) \, ,$$
$$F^G = \frac{(V^G - a_{G'}^{|G|} v_{G'}^{|G|} N^{G'} - \frac{1}{2} A_{G'}^{|G|} v_{G'}^{|G|} M^{G'})}{(\pi [(v_{G'}^{|G|})^2 + (\phi_{G'}^{|G|})^2] (a_{G'}^{|G|} N^{G'} + \frac{1}{2} A_{G'}^{|G|} M^{G'}))^{1/2}} \, ,$$
$$a_{G'}^G = \frac{1}{2} A_{G'}^G + B_{G'}^G \, , \tag{2}$$

where $A_{G'}^G$ and $B_{G'}^G$ are minicolumnar–averaged inter–neuronal synaptic efficacies. $v_{G'}^G$ and $\phi_{G'}^G$ are averaged means and variances of contributions to neuronal electric polarizations. and NN interactions $\underline{V}'$ are detailed in other SMNI papers.

Macroscopic Regions

Inclusion of all the above microscopic and mesoscopic features of neocortex permits a true nonphenomenological Gaussian–Markovian formal development for macroscopic regions encompassing $\sim 5 \times 10^5$ minicolumns of spatial extent $\sim 5 \times 10^9$ μm^2, albeit one that is still highly nonlinear and nonequilibrium. The development of mesocolumnar domains presents conditional probability distributions for mesocolumnar firings with spatially coupled NN interactions. The macroscopic spatial folding of these mesoscopic domains and their macroscopic temporal folding of tens to hundreds of τ, with a resolution of at least τ / N. [11] yields a true path–integral formulation, in terms of a Lagrangian possessing a *bona fide* variational principle for most–probable firing states. At this point in formal development, no continuous–time approximation has yet neen made this is done. with clear justification. for some applications discussed in the next section. This is relevant, e.g., to the possibility of chaotic behavior in neocortex, [10] which. neglecting NN interactions. is essentially a time–discretized. two–dimensional (M^G). dissipative. stochastic system. Much of this algebra is greatly facilitated by. but does not require. the use of Riemannian geometry to develop the nonlinear means. variances. and "potential" contributions to the Lagrangian. [37]

The mathematical macroscopic development proceeds by "folding" the mesoscopic probability distribution over and over, in time θ,

$$\dot{M}^G = [M^G(t+\theta) - M^G(t)]/\theta \, , \, \theta < \tau \, , \tag{3}$$

and in a space $\sim \Lambda \sim 5 \times 10^5$ macrocolumns $\sim 5 \times 10^9$ μm^2. For momentary simplicity. consider the folding of just one variable M at just one spatial point over many time epochs: Labelling u intermediate time epochs by s, i.e., $t_s = t_0 + s\Delta t$. in the limits $\lim_{u \to \infty}$ and $\lim_{\Delta t \to 0}$. and assuming $M_{t_0} = M(t_0)$ and $M_t = M(t \equiv t_{u+1})$ are fixed,

$$P[M_t | M_{t_0}] = \int \cdots \int dM_{t-\Delta t} dM_{t-2\Delta t} \cdots dM_{t_0+\Delta t}$$
$$\times P[M_t | M_{t-\Delta t}] P[M_{t-\Delta t} | M_{t-2\Delta t}] \times \cdots P[M_{t_0+\Delta t} | M_{t_0}] \, ,$$
$$P[M_t | M_{t_0}] = \int \cdots \int \underline{D}M \exp(-\sum_{s=0}^{u} \Delta t \underline{L}_s) \, ,$$
$$\underline{D}M = (2\pi \hat{g}_0^2 \Delta t)^{-1/2} \prod_{s=1}^{u} (2\pi \hat{g}_s^2 \Delta t)^{-1/2} dM_s \, ,$$
$$\int dM_s \to \sum_{\alpha=1}^{N} \Delta M_{\alpha s} \, , M_0 = M_{t_0} \, , M_{u+1} = M_t \, , \tag{4}$$

where α labels the range of N values of M. Extension to multiple variables, e.g.. $G = E$ and I. and to many cells, e.g.. a region of mesocolumns. is discussed in Section IV.2 below.

Mesocolumns were derived in a "prepoint" discretization, e.g.,

$$\dot{M}_s^G = [M^G(t+\theta) - M(t)]/\theta \, ,$$
$$g_s^G = g^G[M^G(t), t] \, . \tag{5}$$

There are a number of non–trivial technical points which must be considered when dealing with multivariate nonlinear systems. Very fortunate for this theory, the necessary mathematical techniques for handling such systems were developed by physicists is the late 1970's, and this neuroscience problem is the first physical system that used these methods.

To capture a flavor of some of the mathematical technicalities, consider that there exists a transformation to the midpoint discretization, in which the standard rules of differential calculus hold for the *same* distribution in terms of a transformed $\underline{L}$, defined as a Feynman Lagrangian $\underline{L}_F$.

$$M^G(\bar{t}_s) = \frac{1}{2}(M_{s+1}^G + M_s^G) \, , \, \dot{M}^G(\bar{t}_s) = (M_{s+1}^G - M_s^G)/\theta \, . \tag{6}$$

i.e.. expanding all prepoint–discretized functions about the midpoint ($t + \theta/2$ above) introduces many additional terms, which are recognized as having the same structure of a Riemannian geometry induced on the M^G variables. These will be specified in more detail in Section IV.2.

Using the midpoint discretization. the variational principle offers insight, but the prepoint discretization does not contain explicit Riemannian terms. The nonlinear variances considerably complicate the algebra required. Riemannian geometry facilitates, but is not necessary, to derive these results. The Riemannian geometry is a reflection that the probability distribution is invariant under general nonlinear transformations of these variables. In other words, the same information content can be expressed in a variety of ways. For example, sensory cortex may transmit information to motor cortex, although they have somewhat different neuronal structures or neuronal languages. Information can be transmitted between "different–looking" regions, e.g., between motor cortex and sensory cortex:

$$I = \int \underline{D}\tilde{M} \, \tilde{P} \ln(\tilde{P}/\bar{P}) \, . \tag{7}$$

3. Applications

Several papers have described in detail how this theory can be used to advantage. [8–14] These applications provide a conceptual framework for treating other similar systems, e.g., those of C^{3}I.

(A) Intuitive view of statistical analyses. Three–dimensional views over $E - I$ of the stationary Lagrangian offers an intuitive "potential" description of neocortical interactions, detailing local minima and maxima. [9,10] Such pairwise presentation of variables offers an intuitive and accurate estimate of relative probabilities and variances associated with multiple minima.

(B) Inclusion of global circuitry. The path–integral formalism permits straightforward extension of this development to include constraints on short–ranged mesocolumnar interactions induced by long–ranged fibers of greater spatial extent than macrocolumnar distances, e.g., long–ranged excitatory fibers from ipsilateral association, contralateral commissural, and thalamocortical processes. [9,10] Such constraints may be viewed as global commands issued to mesoscopic domains, which must use their own internal algorithms on their microscopic units to meet these constraints.

(C) Processing of patterned information. Firing states linearized about stationary firing states, give rise to simple eigenfunction expansions of the macroscopic probability distribution. [8,9] These eigenfunctions are to be identified with the algebraic vector spaces utilized to great advantage by other investigators, [51,52] but not derived by them from realistic synaptic interactions respecting the nonlinear statistical nature of this dynamic system. This identification will permit detailed numerical calculation of associative learning, retrieval and storage of memories, etc. For example, the accuracy of retrieval of a specific pattern is directly proportional to the overlap of a STM "search"–eigenfunction with a long–term memory (LTM) stored eigenfunction. These eigenfunctions may encompass various degrees of neural mass, [50] ranging from minicolumns, to aggregates of mesocolumns coupled by NN interactions, to regions coupled by long–ranged fibers.

More specifically, learning and retrieval mechanisms can be developed by first determining expansion coefficients of eigenfunction expansions of the differential Fokker–Planck distributions, e.g., considering stationary states as Hermite polynomials in neighborhoods of minima. Although this is a reasonably large computer calculation, similar calculations of greater computational difficulty have been performed many years ago, e.g., when calculating quantum states of Schrödinger wave–functions of nucleon–nucleon scattering and of nuclear matter using realistic forces—i.e., quite nonlinear nucleon–nucleon forces derived from meson–exchange forces. [19] The Fokker–Planck equation is quite similar to the Schrödinger equation, and this analogy recently has been used to great advantage, to apply the modern methods used here for neocortex to determine Riemannian contributions to nuclear forces. [20–22] These methods can be very useful for classical systems as well.

(D) Phase transitions and Catastrophes. Higher–order polynomial expansions about stationary states yield Ginsburg–Landau expressions, from

which first–order and second–order phase transitions can be exhibited, if they exist. [10,53] The polynomial expansions, with coefficients derived from empirical synaptic parameters, are a starting point from which to apply methods of Catastrophe Theory, e.g., as discussed by Alex Woodcock at this conference. Such investigations can offer insights into mechanisms that severely alter the global context of a system.

(E) Coding of long–term–memory. A precise scenario of neocortical information processing is detailed, from coding of long–ranged firings from stimuli external to a macrocolumn by short–ranged mesocolumnar firings, to STM storage via hysteresis, and to LTM storage via plastic deformation. [10] In contrast to the appearance of multiple minima in the interior of M^G–space, which are candidates for multiple STM under conditions of sensitive adjustment of synaptic interactions, (see sub–Section G below), [11] typically one or at most a few minima appear at the corners of M^G–space, corresponding to all G–neurons collectively firing or not firing. [10] When these corner minima are present, they are typically much deeper than those found for the interior minima, corresponding to longer–lived states with properties of hysteresis rather than simple jumps. These corner minima are therefore candidates for LTM phenomena. Similar properties of corner minima in simpler models of neocortex have been shown to satisfy properties desirable for multistable perception [54] and for collective computational properties. [55] LTM illustrates the adaptive capabilities of neocortex, a featur very useful for other distributed systems.

(F) Wave–propagation dispersion relations and alpha frequency. Only after the multiple minima are established, then it may be useful to perform linear expansions about specific minima specified by the Euler–Lagrange variational equations. This permits the development of stability analyses and dispersion relations in frequency–wavenumber space. [9,10,12] This calculation requires the inclusion of global constraints, discussed in (B) above.

More specifically, the variational principle permits derivation of the Euler–Lagrange equations. These equations are then linearized about a given local minima to investigate oscillatory behavior. Here, long ranged constraints in the form of Lagrange multipliers J_G were used to efficiently search for minima, corresponding to roots of the Euler–Lagrange equations.

$$0=\hat{\delta}\underline{L}_F=\underline{L}_{F,G,t}-\hat{\delta}_G\underline{L}_F$$
$$\simeq-\underline{f}_{|G|}\ddot{\underline{M}}^{|G|}+\underline{f}^1_G\dot{\underline{M}}^G-\underline{g}_{|G|}\nabla^2\underline{M}^{|G|}+\underline{b}_{|G|}\underline{M}^{|G|}+\underline{b}\,\underline{M}^G\ ,$$
$$[\cdots]_{,G\cdot t}=[\cdots]_{,\dot{G}G'}\dot{M}^{G'}+[\cdots]_{,\dot{G}\dot{G}'}\ddot{M}^{G'}\ ,\ G^{\neg}\neq G\ ,$$
$$\underline{M}^G=M^G-\ll\bar{M}^G\gg\ ,$$
$$\underline{M}^G=\mathrm{Re}\,\underline{M}^G_{\mathrm{osc}}\exp[-i(\underline{\xi}\cdot r-\omega t)]\ ,\ \xi=|\underline{\xi}|\ ,$$
$$\underline{M}^G_{\mathrm{osc}}(r,t)=\int d^2\underline{\xi}\,d\omega\,\hat{\underline{M}}^G_{\mathrm{osc}}(\underline{\xi},\omega)\exp[i(\underline{\xi}\cdot r-\omega t)]\ ,$$
$$\omega\tau=\pm\{-1.86+2.38(\xi\rho)^2;-1.25i+1.51i(\xi\rho)^2\}\ . \tag{8}$$

It is calculated that

$$\omega\sim10^2\ \mathrm{sec}^{-1}\ , \tag{9}$$

which is equivalent to

$$\nu=\omega/(2\pi)=16\ \mathrm{cps\ (Hz)}\ , \tag{10}$$

as observed for the alpha frequency.

The propagation velocity v is calculated from

$$v=d\omega/d\xi\simeq1\ \mathrm{cm/sec}\ ,\ \xi\sim30\rho\ , \tag{11}$$

which tests the NN interactions. Thus, within 10^{-1} sec, short–ranged interactions over several minicolumns of 10^{-1} cm may simultaneously interact with long–ranged interactions over tens of cm, since the long–ranged interactions are speeded by myelinated fibers affording velocities of 600–900 cm/sec. [56] In other words, interaction among different neocortical modalities, e.g., visual, auditory, etc., may simultaneously interact within the same time scales, as observed.

This propagation velocity is consistent with the observed movement of attention [57] and with the observed movement of hallucinations across the visual field, [58] of ~1/2 mm/sec, about 5 times as slow as v. (I.e., the observed movement is ~ 8 msec/°, and a macrocolumn ~ mm processes 180° of visual field.) Therefore, NN interactions may play some part, i.e., within several interations of interactions, in disengaging and orienting selective attention.

(G) Short–term–memory capacity. The most detailed and dramatic application of this theory has been to predict a stochastic mechanism underlying the phenomena of human STM capacity, [11,13] transpiring on the order of tenths of a second to seconds, limited to the retention of 7±2 items. [59] This is true even for apparently exceptional memory performers who, while they may be capable of more efficient encoding and retrieval of STM, and while they may be more efficient in "chunking" larger patterns of information into single items, nevertheless also are limited to a STM capacity of 7±2 items. [60] This "rule" is verified for acoustical STM, but for visual or semantic STM, which typically require longer times for rehearsal in an hypothesized articulatory loop of individual items, STM capacity may be limited to as few as two or three chunks. [61] This STM capacity–limited chunking phenomena also has been noted with items requiring varying depths and breadths of processing. [5–7,16,17] Another interesting phenomena of STM capacity explained by this theory is the primacy vs. recency effect in STM serial processing, wherein first–learned items are recalled most error–free, with last–learned items still more error–free than those in the middle. [62]

STM is the mechanism by which neocortex holds multiple hypotheses for further processing. Multiple minima of Lagrangians modeling similar systems can be similarly analyzed. Contour plots of the stationary Lagrangian, $\underline{\bar{L}}$, for typical synaptic parameters balanced between predominately inhibitory and predominately excitatory firing states, are examined at many scales when the background synaptic noise is only modestly shifted to cause both efferent and afferent mesocolumnar firing states to have a common most–probable firing, centered at [11]

$$M^G=M^{*G}=0\ . \tag{12}$$

Within the range of synaptic parameters considered, for values of $\tau\underline{\bar{L}}\sim10^{-2}$, this "centering" mechanism causes the appearance of from 5 to 10–11 extrema for values of $\tau\underline{\bar{L}}$ on the order of $\sim10^{-2}$. In the absence of external constraints and this centering mechanism, no stable minima are found in the interior of M^G space I.e., the system either shuts down, with no firings, or it becomes epileptic, with maximal firings at the upper limits of excitatory or of excitatory and inhibitory firings. The appearance of these extrema due to the centering mechanism is clearly dependent on the nonlinearities present in the derived Lagrangian, stressing competition and cooperation among excitatory and inhibitory interactions at columnar as well as at neuronal scales.

These number of minima are determined when the resolution of the contours is commensurate with the resolution of columnar firings, i.e., on the order of five to ten neuronal firing per columnar mesh point. Most important contributions to the probability distribution P come from ranges of the time–slice θ and the "action" $N\underline{\bar{L}}$, such that $\theta N\underline{\bar{L}}\leqslant1$. By considering the contributions to the first and second moments of ΔM^G for small time slices θ, conditions on the time and variable meshes can be derived. [63,64]

$$<M^G(t+\theta)-M^G(t)>\simeq g^G(t)\theta\ ,$$
$$<[M^G(t+\theta)-M^G(t)]^2>\simeq g^{GG}(t)\theta\ . \tag{13}$$

The time slice is determined by $\theta\leqslant(N\underline{\bar{L}})^{-1}$ throughout the ranges of M^G giving the most important contributions to the probability distribution P. The variable mesh, a function of M^G, is optimally chosen such that ΔM^G is measured by the covariance $g^{GG'}$ (diagonal in neocortex due to independence of E and I chemical interactions), or $\Delta M^G\sim(g^{GG}\theta)^{1/2}$ in the notation of the SMNI papers. For $N\sim10^2$ and $\underline{\bar{L}}\sim10^{-2}/\tau$, it is reasonable to pick $\theta\sim\tau$. Then it is calculated that that optimal meshes are $\Delta M^E\sim7$ and $\Delta M^I\sim4$, essentially the resolutions used in the coarse contour plots.

Since the extrema appear to lie fairly well along a line in the two–dimensional M^G–space, and since coefficients of slowly varying dM^G/dt terms in the nonstationary $\underline{L}$ are noted to be small perturbations on $\underline{\bar{L}}$, [10] a solution to the stationary probability distribution was hypothesized to be proportional to $\exp(-\Phi/D)$, where $\Phi=CN^2\underline{\bar{L}}$, the diffusion $D=N/\tau$, and C a constant.

$$P_{\mathrm{stat}}\simeq N_{\mathrm{stat}}g^{1/2}\exp(-\Phi/D)\ ,$$
$$\Phi=CN^2\underline{\bar{L}}\sim CN^2\int dM^G L_{,G}\ ,$$
$$D=N/\tau\ . \tag{14}$$

Along the line of the extrema, for $C\simeq1$, this Φ is determined to be an accurate solution to the full two–dimensional Fokker–Planck equation, [13] and a weak–noise high–barrier regime defined by $\Delta\Phi/D>1$, where $\Delta\Phi$ is the difference in Φ from minima to maxima, can be assumed for further analyses. [65]

$$0=\frac{\partial P}{\partial t}=\frac{1}{2}(g^{GG'}P)_{,GG'}-(g^GP)_{,G}+N\underline{V}P\ . \tag{15}$$

This is extremely useful, as a linear stability analysis,

$$\delta\dot{M}^G\simeq-N^2\underline{\bar{L}}_{,GG'}\delta M^{G'}\ . \tag{16}$$

shows that stability with respect to mesocolumnar fluctuations induced by several neurons changing their firings is determined by the second derivatives of $-\Phi$; [66] here this just measures the parabolic curvature of $\underline{\bar{L}}$ at the extrema.

Thus, all the extrema of the stationary Lagrangian are determined to be stable minima of the time–dependent dynamic system. Note however, that it is unlikely that a true potential exists over all M^G–space. [67]

This stationary solution is also useful for calculating the time of first passage, t_{vp}, to fluctuate out of a valley in one minima over a peak to another minima.

$$t_{vp} \simeq \pi N^{-2} \left(| \underline{\bar{L}}_{,GG'}(\ll \bar{M} \gg_p) | \, \underline{\bar{L}}_{,GG'}(\ll \bar{M} \gg_v) \right)^{-1/2}$$
$$\times \exp\{ CN\tau [\underline{\bar{L}}(\ll \bar{M} \gg_p) - \underline{\bar{L}}(\ll \bar{M} \gg_v)] \} \ . \tag{17}$$

It turns out that the values of $\tau \underline{\bar{L}} \sim 10^{-2}$ for which the minima exist are just right to give t_{vp} on the order of tenths a second for about 9 of the minima when the maximum of 10–11 are present. The other minima give t_{vp} on the order of many seconds, which is large enough to cause hysteresis to dominate single jumps between other minima. [11] Thus, 7±2 is the capacity of STM, for memories or new patterns which can be accessed in any order during tenths of a second, all as observed empirically. [60] (When the number of neurons/minicolumn is taken to be ~220, modeling visual neocortex, [11] then the minima become deeper and sharper, consistent with sharper depth of processing, but several minima become isolated from the main group. This effect might be responsible for the lowering of STM capacity for visual processing, mentioned above.)

This is a very sensitive calculation. If N were a factor of 10 larger, or if $\tau \underline{\bar{L}} \sim 0.1$ at the minima, then t_{vp} is on the order of hours instead of tenths of seconds, becoming unrealistic for STM durations. Oppositely, if t_{vp} were much smaller, i.e., less than $\sim 5\tau$, this would be inconsistent with empirical time scales necessary for formation of any memory trace. [68] In this context, it is noted that the threshold factor of the probability distribution scales as $(N^* N)^{1/2}$, demanding that both the macrocolumnar divergence and minicolumnar convergence of mesocolumnar firings be tested by these calculations.

Yin– Yang Processing of Information

This theory demonstrates that, relatively independent of local information–processing at the sub–microscopic synaptic and microscopic neuronal scales, there is statistical global processing of patterns of information at the mesoscopic and macroscopic scales.

This picture represents neocortex as a pattern–processing computer. The underlying mathematical theory, i.e., the path–integral approach, specifies a parallel–processing algorithm which statistically finds those parameter–regions of firing which contribute most to the overall probability distribution: This is a kind of "intuitive" algorithm, globally searching a large multivariate data base to find parameter–regions deserving more detailed local information–processing. The derived probability distribution can be thought of as a filter, or processor, of incoming patterns of information. This filter is adaptive, as it can be modified as it interacts with previously stored patterns of information, changing the mesoscopic synaptic parameters.

IV. APPLICATIONS OF BI TO C^{3}I

1. A Generic System

(A) Target Variables—Recognition In order to make the mathematics more transparent, consider a grid defined within a given time epoch, where the grid is to be conceived as a generalized "radar" screen, representing data being accumulated by multiple sensors. Each cell has information pertaining to relocatable targets that may be moving between cells. Each ">" represents a minimal set of targets, e.g., clusters of targets, which have a number of associated variables, e.g., coordinate position, velocity, acceleration, numbers of targets within these categories, etc. The information collected within each time epoch serves to define changes in these variables between neighboring epochs, both within each cell and between neighboring cells.

Thus, large sets of problems are defined by requiring algorithms to recognize and parametrize changing patterns of these target variables.

(B) Decision–Making Variables—Response It must also be assumed, if objective responses to targets are required, that decision–making variables be defined and functionally parametrized. These variables may include properties of actions to be taken, consistently scaled to match target variables.

Thus, larger sets of problems are defined by requiring algorithms to parameterize and to optimally allocate decision–making variables according to the perceived changing patterns of target variables defined in (A). It is also reasonable to expect that any algorithm for response, i.e., in contradistinction to mere recognition, somehow consistently fold in the parameters of both (A) and (B).

(C) Response– Time and Computational Constraints These problems are further exasperated by the real nature of physical systems. Not much time may be available to optimally solve the problems defined in (A) and (B).

Thus, larger sets of problems are defined by requiring algorithms to respond to problems in (A) and (B), but so constrained that they may not be able to always predict the absolutely best response. It may be necessary to settle for a "good" response.

(D) Fitting and Predicting Error, Noise and Risk Given the absence of perfect humans and of perfect machines, it is clear that any algorithm addressing the problems in (A), (B) and (C) require some degree of parametrization and modeling. There exist some errors in attempting to match any algorithm to a given genuine complex physical system. In order to minimize these errors to within required tolerances, these errors must be quantified.

By design of the targets or by design of the sensors, there also exists some degree of background noise tending to thwart a completely deterministic description of the target variables. This noise must be quantified, at least in order to assess a measure of credibility given to the identification of changing patterns of target variables.

The size and complexity of real physical systems, and the response–time and computational constraints described in (C), dictate that without always being able to make a best single decision, there exist elements of risk in any response algorithm. This risk must be quantified, at least in order to assess the chances to be taken by alternative responses. The "expected gain" of any response is the sum of products of each possible response multiplied by its associated risk, assuming independence among responses; otherwise, cross–correlations must be assessed and folded into this analysis.

Thus, larger sets of problems are defined by requiring algorithms to consistently include fits of variances (error, noise, risk) of all parameters in (A), (B) and (C). Only if variances are consistently fitted, can the mean values (signals), approximately corresponding to the otherwise deterministic parameters in the hypothetical absence of these variances, be extracted. Only if past events include these "2nd moment" fits, i.e., only by fitting *bona fide* probability distributions, can the future be optimally predicted, albeit only with some (quantifiable) degree of statistical (un)certainty.

2. Method of Solution

(A) One Variable, One Cell There are three equivalent representations of this stochastic system.

For momentary simplicity, again consider the above "radar" grid, but now consider only one parameter, $M(t)$, in just one cell, representing just one of the variables discussed in Section (1A) or Section (1B). The problem of determining the change of M within time Δt is

$$M(t+\Delta t) - M(t) = \Delta t \, f[M(t)] \ , \tag{18}$$

where $f[M]$ is some function to be fit, which describes how M is changing. For small enough Δt, and assuming continuity of M, this is often written as

$$\dot{M} = \frac{dM}{dt} = f \ . \tag{19}$$

If background noise, η, is present, assumed to be Gaussian–Markovian ("white" noise), then this affects the description of changing M by

$$\dot{M} = f + \hat{g}\eta \ ,$$
$$<\eta(t)>_\eta = 0 \ ,$$
$$<\eta(t)\eta(t')>_\eta = \delta(t-t') \ , \tag{20}$$

where $\hat{g}^2$ is the (constant here) variance of the background noise. Here η is assumed to have a zero mean. Eq. (20) is referred to as a Langevin rate–equation in the scientific literature.

Physicists and engineers, e.g., in fluid mechanics, recognize an equivalent "diffusion" equation to Eq. (20), defining a differential equation for the conditional probability distribution, $P[M(t+\Delta t)| \, M(t)]$, of finding M at the time $t+\Delta t$, given its value at time t.

$$\frac{\partial P}{\partial t} = \frac{\partial(-fP)}{\partial M} + \frac{1}{2}\frac{\partial^2(\hat{g}^2 P)}{\partial M^2} \tag{21}$$

is known as a Fokker–Planck equation.

Some physicists, e.g., in elementary–particle physics, are familiar with yet another representation of Eq. (20) or (21). For small time epochs, the conditional probability P is

$$P[M_{t+\Delta t} | \, M_t] = (2\pi \hat{g}^2 \Delta t)^{-1/2} \exp(-\Delta t L) \ ,$$
$$L = (\dot{M} - f)^2 / (2\hat{g}^2) \ . \tag{22}$$

L is defined to be the Lagrangian. This representation for P permits a "global'' path–integral description of the evolution of P from time t_0 to a long time t, i.e., in contradistinction to the ''local'' differential Eq. (21). Labelling u intermediate time epochs by s, i.e., $t_s=t_0+s\Delta t$, in the limits $\lim_{u\to\infty}$ and $\lim_{\Delta t\to 0}$, and assuming $M_{t_0}=M(t_0)$ and $M_t=M(t\equiv t_{u+1})$ are fixed.

$$P[M_t|\,M_{t_0}]=\int\cdots\int dM_{t-\Delta t}\,dM_{t-2\Delta t}\cdots dM_{t_0+\Delta t}$$
$$\times P[M_t|\,M_{t-\Delta t}]P[M_{t-\Delta t}|\,M_{t-2\Delta t}]\times\cdots P[M_{t_0+\Delta t}|\,M_{t_0}]\,,$$
$$P[M_t|\,M_{t_0}]=\int\cdots\int \underline{D}M\exp(-\sum_{s=0}^{u}\Delta t L_s)\,,$$
$$\underline{D}M=(2\pi\hat{g}_0^2\Delta t)^{-1/2}\prod_{s=1}^{u}(2\pi\hat{g}_s^2\Delta t)^{-1/2}dM_s\,,$$
$$\int dM_s\to\sum_{\alpha=1}^{N}\Delta M_{\alpha s}\,,M_0=M_{t_0}\,,M_{u+1}=M_t\,, \tag{23}$$

where α labels the range of N values of M. For notational simplicity, the indices s and α often will be dropped in the following, but these time and range discretizations must of course be explicitly programmed in all actual numerical calculations.

There are some advantages to the path–integral representation over its equivalent Fokker–Planck and rate–equation representations. For example, there exists a variational principle wherein a set of Euler–Lagrange differential equations exist for the Lagrangian L, directly yielding those values or trajectories of M which give the largest contribution to the probability distribution P.

Because P is a *bona fide* probability distribution, there exist Monte Carlo numerical algorithms, sampling the M–space without having to calculate all values of M at all intermediate time epochs from t_0 to t to find P. This numerical algorithm also has the nice feature of avoiding traps in local minima when there are deeper minima to be had, representing more probable states. This is so useful that noise is sometimes artificially added to otherwise deterministic systems, e.g., as in simulated annealing [69] to derive optimum circuitry on chips, by hypothesizing a cost function similar to the potential Φ in Eq. (14) in Section III. More efficient simulated annealing algorithms for finding a global minimum of a cost function or set of data have been discussed by Harold Szu at this conference.

In practice, some of these benefits are often illusory. Monte Carlo methods are notoriously poor for most nonstationary systems with multiple minima. However, a new method has been developed for explicitly solving the path integral, thereby obtaining the dynamic evolution of all states (minima) of the system. [63,64] This cannot be done with the differential equation representations. Calculating P via the path integral facilitates the inclusion of boundary conditions, and the new methods also can take advantage of the Gaussian–Markovian nature of the system to produce an efficient numerical algorithm.

(B) Many Nonlinear Variables It is possible to formulate Langevin equations generalized from Eq. (20),

$$\dot{M}^G=f^G+\hat{g}_i^G\eta^i\,,$$
$$i=1,\cdots,\Xi\,,$$
$$G=1,\cdots,\Theta\,, \tag{24}$$

where G corresponds to any number of Θ variables, e.g., target and decision–making variables in (IA) and (IB), f^G and $\hat{g}_i^G$ are arbitrarily nonlinear functions of any or all M^G, and of t, and the index i corresponds to recognizing that there can be many different sources contributing to the variance of M^G. The time of evaluation of $\hat{g}_{si}^G$ during s–epochs intermediate between t_0 and t, $\bar{t}_s$ between t_s and $t_{s+1}=t_s+\Delta t$, must now be explicitly prescribed. Unless otherwise specified, a midpoint Stratonovich rule will be chosen here, using $M^G(\bar{t}_s)=\frac{1}{2}(M_{s+1}^G+M_s^G)$, $\dot{M}^G(\bar{t}_s)=(M_{s+1}^G-M_s^G)/\Delta t$, and $\bar{t}_s=t_s+\Delta t/2$. This choice is consistent with other physical systems, and allows the use of standard calculus in Eq. (24).

The path integral generalized from Eq. (23) is written as

$$P=\int\cdots\int\underline{D}M\exp(-\sum_{s=0}^{u}\Delta t L_s)\,,$$
$$\underline{D}M=g_0^{1/2}(2\pi\Delta t)^{-1/2}\prod_{s=1}^{u}g_s^{1/2}\prod_{G=1}^{\Theta}(2\pi\Delta t)^{-1/2}dM_s^G\,,$$
$$\int dM_s^G\to\sum_{\alpha=1}^{N^G}\Delta M_{\alpha s}^G\,,M_0^G=M_{t_0}^G\,,M_{u+1}^G=M_t^G\,,$$
$$L=\frac{1}{2}(\dot{M}^G-h^G)g_{GG'}(\dot{M}^{G'}-h^{G'})+\frac{1}{2}h^G{}_{;G}+R/6-V\,,$$
$$[\cdots]_{,G}=\frac{\partial[\cdots]}{\partial M^G}\,,$$
$$h^G=g^G-\frac{1}{2}g^{-1/2}(g^{1/2}g^{GG'})_{,G'}\,,$$
$$g_{GG'}=(g^{GG'})^{-1}\,,$$
$$g_s[M^G(\bar{t}_s),\bar{t}_s]=\det(g_{GG'})_s\,,g_s=g_s[M_{s+1}^G,\bar{t}_s]\,,$$
$$h^G{}_{;G}=h_{,G}^G+\Gamma_{GF}^F h^G=g^{-1/2}(g^{1/2}h^G)_{,G}\,,$$
$$\Gamma_{JK}^F\equiv g^{LF}[JK,L]=g^{LF}(g_{JL,K}+g_{KL,J}-g_{JK,L})$$
$$R=g^{JL}R_{JL}=g^{JL}g^{JK}R_{FJKL}\,,$$
$$R_{FJKL}=\frac{1}{2}(g_{FK,JL}-g_{JK,FL}-g_{FL,JK}+g_{JL,FK})+g_{MN}(\Gamma_{FK}^M\Gamma_{JL}^N-\Gamma_{FL}^M\Gamma_{JK}^N)\,. \tag{25}$$

Note that the variance $g^{GG'}$ is the GG'–matrix inverse of the G–space metric $g_{GG'}$. R is calculated to be the Riemannian curvature scalar, and Γ_{JK}^F is the affine connection in this space.

(C) Many Cells For many cells, i.e., Λ cells indexed by ν, the path integral in Eq. (25) is further generalized, essentially by expanding the parameter space from the set $\{G\}$ to the set $\{G,\nu\}$.

Constraints may be placed on variables by adding them to the potential $\tilde{V}_s$, e.g., as $J_{sG\nu}M_s^{G\nu}$ with Lagrange multipliers $J_{sG\nu}$.

If a prepoint–discretization rule is adopted, transforming from the midpoint–discretized Feynman $\tilde{L}_s$ and $\tilde{g}_s$, to define $\dot{M}^{G\nu}(\bar{t}_s)=(M_{s+1}^{G\nu}-M_s^{G\nu})/\Delta t$, $M^{G\nu}(\bar{t}_s)=M_s^{G\nu}$, $\bar{t}_s=t_s$, and $\tilde{g}_s=\tilde{g}_s$, then a simpler expression is obtained for the Lagrangian, one in which the Riemannian terms are not explicitly present.

$$\tilde{L}'=\frac{1}{2}(\dot{M}^{G\nu}-g^{G\nu})g_{GG'\nu\nu'}(\dot{M}^{G'\nu'}-g^{G'\nu'})-\tilde{V}\,. \tag{26}$$

However, although $\tilde{P}$ is invariant under this transformation, $\tilde{L}'$ does not possess the variational principle possessed by the Feynman Lagrangian $\tilde{L}$, so that if the prepoint–discretized $\tilde{L}'$ and $\tilde{g}_s$ are used to fit the data, then some tests must still be made to see how efficiently the path integral can be calculated using $\tilde{L}'$ instead of $\tilde{L}$ to globally scan the data.

Eq. (25) (or first its equivalent prepoint discretization) will be fit to the data by assuming functional forms for $\tilde{V}_s$, $g_s^{G\nu}$ and $g_s^{GG'\nu\nu'}$. The convergence of $\tilde{L}$ or $\tilde{L}'$ is expected to be quite good. I.e., even polynomial forms for $g_s^{G\nu}$ and $g_s^{GG'\nu\nu'}$, with coefficients to be fit, define a Padé rational approximate to $\tilde{L}$ usually giving better convergence than obtained for $g_s^{G\nu}$ or $g_s^{GG'\nu\nu'}$ separately. Also, note that $\tilde{L}_s$ is a single scalar function to be fit.

$$g^G=X^G+X_{G'}^G\underline{M}^{G'}+X_{G'G''}^G\underline{M}^{G'}\underline{M}^{G''}+\cdots\,,$$
$$g_{GG'}=Y_{GG'}+Y_{GG'G''}\underline{M}^{G''}+Y_{GG'G''G'''}\underline{M}^{G''}\underline{M}^{G'''}+\cdots\,,$$
$$\underline{M}_s^{G\nu}=M_s^{G\nu}-\ll M_s^{G\nu}\gg\,. \tag{27}$$

Once the parameters $\{X,Y,\ll M\gg\}$ are fit, the theory is ready to track or predict. Science is not only empiricism. Modeling and chunking of information is required, not only for aesthetics, but also to reduce required computational resources of brains as well as machines.

3. Future Research and Development

Given a complex system possessing many variables, I believe it appropriate to initially apply some non–parametric statistical methods as a coarse ''macroscopic'' filter to discover, even in real time, some systematics of the system. An example is mentioned in the next Section V.

These macroscopic systematics can form the basis of a first–order set of trial functions for a ''mesoscopic'' filter, e.g., modeled as a parametric nonlinear nonequilibrium Gaussian Markovian statistical mechanics, as discussed above. [25] This filter can be used to ascertain just what scope of the underlying variable space should be allocated further detailed, more expensive and time–consuming processing by relatively microscopic algorithms. Or, this mesoscopic filter may be sufficient, e.g., for ''shotgun'' responses to clusters of targets.

The final level of detailed processing most likely needs to be performed by a ''microscopic'' fine filter which is *not* explicitly dependent on macroscopic or mesoscopic properties. Markovian or Gaussian properties generally are only appropriate and useful for aggregates of microscopic details. Typically, specific complex systems at the microscopic level exhibit even fewer typical features than the typically novel features discovered even at the mesoscopic level.

My work in neuroscience discussed above suggests an approach for implementing the mesoscopic filter into hardwiring. Consider each cell of the "radar" screen above now be represented as one ν–cell at a given time labeled by s. Each circle consists of $\sim 10^2$ on–off bits, representing N^G α–states of one G–variable $M^{G\nu}_{\alpha s}$ in that ν–cell at time s, which therefore represents a field rather than a simple binary node. Each circle statistically reacts to the other circles in that cell and in ν_{NN} cells at time $s-1$, according to an algorithm encoded in each ν–cell. Long–ranged constraints might be added by superimposing (magnetic) fields, i.e., modeling the J_{sG_ν} constraints described in Section (2C) above.

V. COMBAT SIMULATIONS

An important class of problems confronting C^3I systems concerns how to pass through enough, but not too much, timely information to decision–makers to permit them to assess the overall "macroscopic" nature of detailed "microscopic" operations unfolding in time. Similarly, there must also be a reasonable information–conduit through which their macroscopic decisions can be effectively implemented at the microscopic level.

It is proposed that modern methods of nonlinear nonequilibrium statistical mechanics be utilized to approach such problems, not just to merely model abstract scenarios. Basically, this approach seeks to define a "mesoscopic" scale, established between the microscopic and macroscopic scales, specifically appropriate to each C^3I system: nonlinear multivariate functions describing drifts (trends) and diffusions (risks) must be sought. This requires trial and error, intelligence and creativity, and much experience to be gained by dealing with at least several C^3I systems. These functional forms and their coefficients must be fit to real empirical data, e.g., initial, intermediate and final resources, to develop a time–dependent multivariable probability distribution of order parameters defining the mesoscopic scale. Then, after this algebraic and numerical development, there is the possibility that the resulting codes can be implemented on small computers in the field, affording useful software support for decision–making and intelligence–gathering, while being robust against perturbations in these functional fits.

At NPS, Stephen Upton and I are developing statistical mechanical C^3I models of combat simulations. As pointed out in the Introduction Section I, simulations can be an important source of empirical data, only if their assumptions are clearly recognized, I.e., they are at best only as good as they model actual combat. [70]

Our primary focus is an NPS simulation reported at this conference by Mike Sovereign and Joe Stewart, Interim Battle Group Tactical Trainer (IBGTT). IBGTT is rather unique in possessing a high degree of human–machine interactions. It is hoped that by fitting nonlinear statistical mechanical models to this data, we may capture the essence of realistic combat operations. The previous work ar NPS has accomplished a coarse macroscopic linear regression of three years of data, e.g., as discussed in Section IV.3. We plan to construct the mesoscopic model.

Another simulation we are investigating for a similar mesoscopic analysis is to model the C^3 system of a Marine Air–Ground Task Force (MAGTF), composed of four elements: Command, Ground Combat, Aviation Combat, Combat Service Support. An example of such a simulation is the Tactical Warfare Simulation, Evaluation and Analysis System (TWSEAS). These are located at: MC Development and Education Center (MCDEC), Quantico, VA; Camp Lejune, NC; Camp Pendelton, CA. There are three types of MAGTF's: Marine amphibious unit (MAU), Marine amphibious brigade (MAB), Marine amphibious force (MAF). The C^3 structure of all MAGTF's is given in the chart below.

The MAGTF order parameters M^G of the air support might include measures of readiness (aggregated by other relatively microscopic algorithms) of: (1.a) weapons carried and (1.b) personnel carried; in turn (1.a) and (1.b) depend on the order parameters defining capabilities of the ground troops and the logistic systems. By establishing a functional probability distribution that might truly describe the dynamic MAGTF, i.e., admitting arbitrarily nonlinear drifts and diffusions, alternative scenarios can be objectively assessed by commanders who are presented with information at a level commensurate with their tasks, and their decisions can be established as constraints on the mesoscopic cells of microscopic networks of the MAGTF.

VI. STATISTICAL BI DECISION–MAKING

A typical scenario that might take advantage of previous analysis that has fit a Lagrangian to previous data follows. For example, assume that in the middle of an engagement, a commander (human or machine) has available data representing measures of readiness of his forces and those of his enemy. He makes a judgement as to which of several established classes of conflict he is engaged in, e.g., possibly severely or moderately stochastic, possibly overwhelming resources in, or not in, his favor, etc. He chooses one of previously established Lagrangians which is a coarse description of his present engagement, and sets the initial time boundary condition according to his present data.

He chooses some time in the future when he feels he will be called on to make a judgement with regard to the deployment of his resources. He uses a small computer to determine the distribution of his variables at the future time. Most likely, he will obtain several possible likely states, with varying degrees of first moments ("probability") and second moments ("risk").

He might do this for several alternative initial parameter settings, especially if he can exercise some immediate control of their values, thereby obtaining another possible set of future states of the engagement. He also might have to fold in some constraints, in the form of Lagrange multipliers, to accommodate orders he has received from a higher command. He could also use the associated Euler–Lagrange variational equations to determine the most likely trajectory that his resources would follow enroute from his present state to his selected future state.

Thus, the commander has obtained a valuable source of information to aid him in making decisions, and in determining sets of orders of constraints which he should pass down to his subordinates. Conversely, his subordinates, by aggregating their data into the specified order parameters, can communicate information to their commander in a language readily accessible to his decision–making process.

REFERENCES

[1] B.G. Blair, *Strategic Command and Control* (Brooking Institution, Washington, D.C., 1985).

[2] Eastport Study Group, *A Report to the Director Strategic Defense Initiative Organization* (SDIO, Washington, D.C., 1985).

[3] G.E. Orr, *Combat Operations C3I: Fundamentals and Interactions* (Air University, Maxwell Air Force Bace, AL, 1983).

[4] L. Ingber, "Physics of karate techniques," Sensei (Instructor) Thesis, Japan Karate Association, Tokyo, Japan, 1968.

[5] L. Ingber, *The Karate Instructor's Handbook* (PSI–ISA, Solana Beach, CA, 1976).

[6] L. Ingber, *Karate: Kinematics and Dynamics* (Unique, Hollywood, CA, 1981).

[7] L. Ingber, *Elements of Advanced Karate* (Ohara, Burbank, CA, 1985).

[8] L. Ingber, "Towards a unified brain theory," *J. Social Biol. Struct.* **4**, 211–224 (1981).

[9] L. Ingber, "Statistical mechanics of neocortical interactions. I. Basic formulation," *Physica D* **5**, 83–107 (1982).

[10] L. Ingber, "Statistical mechanics of neocortical interactions. Dynamics of synaptic modification," *Phys. Rev. A* **28**, 395–416 (1983).

[11] L. Ingber, "Statistical mechanics of neocortical interactions. Derivation of short–term–memory capacity," *Phys. Rev. A* **29**, 3346–3358 (1984).

[12] L. Ingber, "Statistical mechanics of neocortical interactions. EEG dispersion relations," *IEEE Trans. Biomed. Eng.* **32**, 91–94 (1985).

[13] L. Ingber, "Statistical mechanics of neocortical interactions: Stability and duration of the 7±2 rule of short–term–memory capacity," *Phys. Rev. A* **31**, 1183–1186 (1985).

[14] L. Ingber, "Towards clinical applications of statistical mechanics of neocortical interactions," *Innov. Tech. Biol. Med.* **6**, 753–758 (1985).

[15] L. Ingber, "Statistical mechanics of neocortical interactions," *Bull. Am. Phys. Soc.* **31**, 868 (1986).

[16] L. Ingber, "Editorial: Learning to learn," *Explore* **7**, 5–8 (1972).

[17] L. Ingber, "Attention, physics and teaching," *J. Social Biol Struct.* **4**, 225–235 (1981).

[18] L. Ingber, "Statistical mechanics algorithm for response to targets (SMART)," in *Workshop on Uncertainty and Probability in Artificial Intelligence*, (AAAI–RCA, Menlo Park, CA, 1985), p. 258–264.

[19] L. Ingber, "Nuclear forces," *Phys. Rev.* **174**, 1250–1263 (1968).

[20] L. Ingber, "Riemannian Corrections to velocity–dependent nuclear forces," *Phys. Rev. C* **28**, 2536–2539 (1983).

[21] L. Ingber, "Path–integral Riemannian contributions to nuclear Schrödinger equation," *Phys. Rev. D* **29**, 1171–1174 (1984).

[22] L. Ingber, "Riemannian contributions to short–ranged velocity–dependent nucleon–nucleon interactions," *Phys. Rev. D* **33**, 3781–3784 (1986).

[23] L. Ingber, "Statistical mechanics of nonlinear nonequilibrium financial markets," *Math. Modelling* **5**, 343–361 (1984)

[24] J.G. Taylor, *Lanchester Models of Warfare*, Vols. I and II (Operations Research Society of America, Arlington, VA, 1983).

[25] National Research Council Committee on the Applications of Mathematics, *Computational Modeling and Mathematics Applied to the Physical Sciences* (National Academy Press, Washington, D.C., 1984).

[26] H. Haken, *Synergetics*, 3rd ed. (Springer, New York, 1983).

[27] N.G. van Kampen, *Stochastic Processes in Physics and Chemistry* (North–Holland, Amsterdam, 1981).

[28] J.D. Bekenstein and L. Parker, "Path integrals for a particle in curved space," *Phys. Rev. D* **23**, 2850–2869 (1981).

[29] K.S. Cheng, "Quantization of a general dynamical system by Feynman's path integration formulation," *J. Math. Phys.* **13**, 1723–1726 (1972).

[30] H. Dekker, "Functional integration and the Onsager–Machlup Lagrangian for continuous Markov processes in Riemannian geometries," *Phys. Rev. A* **19**, 2102–2111 (1979).

[31] B.S. DeWitt, "Dynamical theory in curved spaces. I. A review of the classical and quantum action principles," *Rev. Mod. Phys.* **29**, 377–397 (1957).

[32] H. Grabert and M.S. Green, "Fluctuations and nonlinear irreversible processes," *Phys. Rev. A* **19**, 1747–1756 (1979).

[33] H. Grabert, R. Graham, and M.S. Green, "Fluctuations and nonlinear irreversible processes. II.," *Phys. Rev. A* **21**, 2136–2146 (1980).

[34] R. Graham, "Covariant formulation of non–equilibrium statistical thermodynamics," *Z. Physik* **B26**, 397–405 (1977).

[35] R. Graham, "Lagrangian for diffusion in curved phase space," *Phys. Rev. Lett.* **38**, 51–53 (1977).

[36] T. Kawai, "Quantum action principle in curved space," *Found. Phys.* **5**, 143–158 (1975).

[37] F: Langouche, D. Roekaerts, and E. Tirapegui, *Functional Integration and Semiclassical Expansions* (Reidel, Dordrecht, 1982).

[38] M.M. Mizrahi, "Phase space path integrals, without limiting procedure," *J. Math. Phys.* **19**, 298–307 (1978).

[39] L.S. Schulman, *Techniques and Applications of Path Integration* (J. Wiley & Sons, New York, 1981).

[40] R.P. Feynman and A.R. Hibbs, *Quantum Mechanics and Path Integrals* (McGraw–Hill, New York, 1965).

[41] S.J. Williamson, L. Kaufman, and D. Brenner, "Evoked neuromagnetic fields of the human brain," *J. Appl. Phys.* **50**, 2418–2421 (1979).

[42] G.M. Shepherd, *The Synaptic Organization of the Brain*, 2nd ed. (Oxford Univ., New York, NY, 1979).

[43] P.S. Goldman and W.J.H. Nauta, "Columnar distribution of cortico–cortical fibers in the frontal association, limbic, and motor cortex of the developing rhesus monkey," *Brain Res.* **122**, 393–413 (1977).

[44] D.H. Hubel and T.N. Wiesel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," *J. Physiol.* **160**, 106–154 (1962).

[45] T.J. Imig and R.A. Reale, "Patterns of cortico–cortical connections related to tonotopic maps in cat auditory cortex," *J. Comp. Neurol.* **192**, 293–332 (1980).

[46] E.G. Jones, J.D. Coulter, and S.H.C. Hendry, "Intracortical connectivity of architectonic fields in the somatic sensory, motor and parietal cortex of monkeys," *J. Comp. Neurol.* **181**, 291–348 (1978).

[47] V.B. Mountcastle, "An organizing principle for cerebral function: The unit module and the distributed system," in *The Mindful Brain*, ed. by G.M. Edelman and V.B. Mountcastle (Massachusetts Institute of Technology, Cambridge, 1978), p. 7–50.

[48] T.H. Bullock, "Reassessment of neural connectivity and its specification," in *Information Processing in the Nervous System*, ed. by H.M. Pinsker and W.D. Willis, Jr. (Raven Press, New York, NY, 1980).

[49] R.W. Dykes, "Parallel processing of somatosensory information: A theory," *Brain Res. Rev.* **6**, 47–115 (1983).

[50] R.P Erickson, "The across–fiber pattern theory: An organizing principle for molar neural function," *Sensory Physiol.* **6**, 79–110 (1982).

[51] J.A. Anderson, J.W. Silverstein, S.A. Ritz, and R.S. Jones, "Distinctive features, categorical perception and probability learning: Some applications of a neural model," *Psych. Rev.* **84**, 413–451 (1977).

[52] L.N. Cooper, "A possible organization of animal memory and learning," in *Collective Properties of Physical Systems*, ed. by B. Lundqvist and S. Lundqvist (Academic Press, New York, NY, 1973), p. 252–264.

[53] S.–K. Ma, *Modern Theory of Critical Phenomena* (Benjamin/Cummings, Reading, MA, 1976).

[54] A.H. Kawamoto and J.A. Anderson, "A neural model of multistable perception," *Acta Psychologica* **59**, 35–65 (1985).

[55] J.J. Hopfield, "Neurons with graded responses have collective computational properties like those of two–state neurons," *Proc. Natl. Acad. Sci. USA* **81**, 3088–3092 (1984).

[56] P.L. Nunez, *Electric Fields of the Brain: The Neurophysics of EEG* (Oxford Univ., New York, 1981).

[57] Y. Tsal, "Movements of attention across the visual field," *J. Exp. Psychol.* **9**, 523–530 (1983).

[58] J.D. Cowan, "Spontaneous symmetry breaking in large scale nervous activity," *Int. J. Quant. Chem.* **22**, 1059–1082 (1982).

[59] G.A. Miller, "The magical number seven, plus or minus two," *Psychol. Rev.* **63**, 81–97 (1956).

[60] K.A. Ericsson and W.G. Chase, "Exceptional memory," *Am. Scientist* **70**, 607–615 (1982).

[61] G. Zhang and H.A. Simon, "STM capacity for Chinese words and idioms: Chunking and acoustical loop hypotheses," *Memory & Cognition* **13**, 193–201 (1985).

[62] B.B. Murdock, Jr., "A distributed memory model for serial–order information," *Psychol. Rev.* **90**, 316–338 (1983).

[63] M.F. Wehner and W.G. Wolfer, "Numerical evaluation of path–integral solutions to Fokker–Planck equations. I.," *Phys. Rev. A* **27**, 2663–2670 (1983).

[64] M.F. Wehner and W.G. Wolfer, "Numerical evaluation of path–integral solutions to Fokker–Planck equations. II. Restricted stochastic processes," *Phys. Rev. A* **28**, 3003–3011 (1983).

[65] S.R. Shenoy and G.S. Agarwal, "First–passage times and hysteresis in multivariable stochastic processes: The two–mode ring laser," *Phys. Rev. A* **29**, 1315–1325 (1984).

[66] G.S. Agarwal and S.R. Shenoy, "Observability of hysteresis in first–order equilibrium and nonequilibrium phase transitions," *Phys. Rev. A* **23**, 2719–2723 (1981).

[67] R. Graham and T. Tél, "Existence of a potential for dissipative dynamical systems," *Phys. Rev. Lett.* **52**, 9–12 (1984).

[68] B. Libet, "Brain stimulation in the study of neuronal functions for conscious sensory experience," *Human Neurobiol.* **1**, 235–242 (1982):

[69] S. Kirkpatrick, C.D. Gelatt, Jr., and M.P. Vecchi, "Optimization by simulated annealing," *Science* **220**, 671–680 (1983).

[70] Comptroller General, *Models, Data, and War: A Critique of the Foundation for Defense Analyses* (U.S. General Accounting Office, Washington, D.C., 1980).

List of Attendees of the 9th Workshop on C^3 Systems

Maurice F. Aburdene
Associate Professor
Dept. of EE&CS
Bucknell University
Lewisburg, PA 17837
Tel: (717) 524-1449

Les Anderson
Scientist
Code 443
Naval Ocean Systems Center
271 Catalina Blvd.
San Diego, CA 92152-5000
Tel: (619) 225-7796

Stamatios K. Andreadakis
Graduate Student
Massachusetts Institute of Technology
Room 35-417/LIDS
77 Massachusetts Avenue
Cambridge, MA 02139
Tel: (617) 253-2346

Michael Athans
Professor EECS
Massachusetts Institute of Technology
Room 35-406/LIDS
77 Massachusetts Avenue
Cambridge, MA 02139
Tel: (617) 253-6173

James P. Baker
Manager
Space Systems Mission Analysis
General Electric Company
P.O. Box 8555 M-3041
Philadelphia, PA 19101
Tel: (215) 354-4839

Richard Bisbey, II
Information Sciences Institute
4676 Admiralty Way
Marina Del Rey, CA 90292

Linda G. Bushnell
Graduate Student
Dept. of EE&CS
The University of Connecticut
Box U-157
Storrs, CT 06268
Tel: (203) 486-2210

Kenneth Roger Casey
DPIV
Code 84
Naval Ocean Systems Center
271 Catalina Blvd.
San Diego, CA 92152-5000
Tel: (619) 225-2188

David A. Castanon
Research Scientist
ALPHATECH, Inc.
2 Burlington Executive Park
111 Middlesex Turnpike
Burlington, MA 01803
Tel: (617) 273-3388

Ceasar Castro
Code 841
Naval Ocean Systems Center
271 Catalina Blvd.
San Diego, CA 92152-5000

Samuel Chamberlian
Systems Engineer
Director, USABRL
ATTN: SLCBR-SECAD-S
APG, MD 21005-5066
Tel: (301) 278-6660

Robert W. Choisser
Technical Advisor
Defense-Wide C3 Directorate
Defense Communications Agency
Code A702
Arlington Hall Station
Arlington, VA 22212-5409
Tel: (202) 692-6280

Chee-Yee Chong
Department Manager
Advanced Decision Systems
201 San Antonio Circle
Suite 286
Mountain View, CA 94040
Tel: (415) 941-3912

Gerald A. Clapp
USMC C3I Exploratory Development
Code 808
Naval Ocean Systems Center
271 Catalina Blvd.
San Diego, CA 92152-5000

Marvin S. Cohen
Vice President
Decision Science Consortium, Inc.
7700 Leesburg Pike
Suite 421
Falls Church, VA 22043
Tel: (703) 790-0510

Gino J. Coviello
Deputy Chief
Office of Advanced Technology
11440 Isaac Newton Square, North
Reston, VA 22090-5087
Tel: (703) 437-2506

Jacqueline L. Dana
Member Technical Staff
Hughes Aircraft Company
618/P311
P.O. Box 3310
Fullerton, CA 92634
Tel: (714) 732-1112

Martin P. Dana
Sr. Scientist
Hughes Aircraft Company
1901 W. Malvern Avenue
M/S Bldg. 618, Room Q311
Fullerton, CA 92634
Tel: (714) 732-1112

F. D. Deffenbaugh
PAR Technologies
690 W. Knox Street
Suite 210
Torrance, CA 90502
Tel: (213) 516-9022

William J. Dejka
Code 411
Naval Ocean Systems Center
271 Catalina Blvd.
San Diego, CA 92152-5000

John R. Delaney
Technical Staff Member
MIT Lincoln Laboratory
P.O. Box 73
Lexington, MA 02173
Tel: (617) 863-5500

Hugh Dempsey
Senior Analyst
2333 Village Drive, N.E.
Lawton, OK 73507
Tel: (415) 351-5707

George Dillard
Scientist
Code 7402
Naval Ocean Systems Center
271 Catalina Blvd.
San Diego, CA 92152-5000

Robin A. Dillard
Mathematician
Code 444
Naval Ocean Systems Center
271 Catalina Blvd.
San Diego, CA 92152-5000
Tel: (619) 225-7778

Donald R. Edmonds
Department Staff
Nicholas Research Corporation
1764 Old Meadow Lane
Suite 150
McLean, VA 22102-4307
Tel: (703) 893-9720

Elliott Entin
Psychologist
ALPHATECH, Inc.
2 Burlington Executive Park
111 Middlesex Turnpike
Burlington, MA 01803
Tel: (617) 273-3388

Philip Feld
Director C3I Programs
Defense Systems Inc.
8500 Leesburg Pike
6th Floor Suite
Vienna, VA 22180
Tel: (703) 883-1063

George W. Futch
Staff Engineer
Sperry Coporation
1555 Wilson Blvd.
Suite 501
Arlington, VA 22209-2457
Tel: (703) 558-7250

C. J. Gadsden
Principal Scientific Officer
Admiralty Research Establishment
Ministry of Defense
Portsdown, Portsmouth, Hants
ENGLAND
Tel: 0705-379411 Ext. 3355

Bernard W. Galing
TRAC MTRY
P.O. Box 8692
Naval Postgraduate School
Monterey, CA 93943-0692

Sheldon Gardner
EE
Code 5709
Naval Research Laboratory
Washington, D.C. 20375
Tel: (202) 767-1167

Sherman Gee
Manager
Navy C3 Exploratory Dept.
Code 221
Office of Naval Research
800 N. Quincy Street
Arlington, VA 22217
Tel: (202) 696-4791

Evaggeios Geramotis
Assistant Professor
Electrical Engineering Dept.
University of Maryland
College Park, MD 20742
Tel: (301) 454-8840

Jeffrey M. Gilbert
Associate Engineer
Johns Hopkins University
Applied Physics Laboratory
Bldg. 1W, Room 129
Laurel, MD 20707
Tel: (301) 953-5000 Ext. 7294

I. R. Goodman
Code 421
Building 600 Seaside
Naval Ocean Systems Center
271 Catalina Blvd.
San Diego, CA 92152-5000
Tel: (619) 225-2015

Kenneth Gotberg
Scientist
Code 443
Naval Ocean Systems Center
271 Catalina Blvd.
San Diego, CA 92152-5000
Tel: (619) 225-7367

Danny M. Hardin
Physicist
General Research Corporation
307 Wynn Drive
Huntsville, AL 35805
Tel: (205) 837-7900

Henry A. Heidary
Sr. Scientisti
Hughes Aircraft Company
1901 W. Malvern Avenue
M/S Bldg. 618, Room Q311
Fullerton, CA 92634
Tel: (714) 732-5445

Dennis Hollingworth
USC-Information Sciences Institute
4676 Admiralty Way
Marina Del Rey, CA 90292

Lester Ingber
National Research Council
Operations Research - Code 55
Naval Postgraduate School
Monterey, CA 93943-5100
Tel: (408) 646-2801

Thomas Jasinki
Calspan Corporation
Buffalo, NY

L. B. Jocic
Aerospace Corporation
P.O. Box 92957
Los Angeles, CA 90009

Carl R. Jones
Professor
Information and Telecommunication Systems
C^3 Academic Group (Code 74)
Naval Postgraduate School
Monterey, CA 93943
Tel: (408) 646-2767

Jerry L. Kaiwi
Engineer
Code 7121
Navy Personnel Research & Development
Center
San Diego, CA 92152-6800
Tel: (619) 225-2081

Suzanne Kelly
2d Lt., Research Engineer
AAMRL/HEC
Wright-Patterson AFB
Ohio 45433-6573
Tel: 513) 255-8807

William H. King
Head, Tracking and Data Fusion
Systems Division
Hughes Aircraft Company
P.O. Box 3310
Fullerton, CA 92634-3310
Tel: (714) 732-1046

David L. Kleinman
Professor
Dept. of EE&CS
University of Connecticut
Box U-157
Storrs, CT 06268

Thomas Kurien
Member Technical Staff
ALPHATECH, Inc.
2 Burlington Executive Center
111 Middlesex Turnpike
Burlington, MA 01803

Manchi Kwong
Scientist
Code 443
Naval Oean Systems Center
271 Catalina Blvd.
San Diego, CA 92152-5000
Tel: (619) 225-7796

William L. Lakin
Admiralty Research Establishment
Portsdown
Portsmouth P06 4AA
ENGLAND
Tel: (44) 705-374911

John Paul Lehoczky
Professor
Department Head -Statistics
Carnegie Mellon University
Schenley Park - Baker Hall #232
Pittsburgh, PA 15213
Tel: (412) 268-8725

Alexander H. Levis
Senior Research Scientist
Massachusetts Institute of Technology
35-410/LIDS
77 Massachusetts Avenue
Cambridge, MA 02139
Tel: (617) 253-7262

Nhung Thi Lu
Computer Scientist
Code 443
Naval Ocean Systems Center
271 Catalina Blvd.
San Diego, CA 92152-5000
Tel: (619) 225-7196

Gerald S. Malecki
Scientific Officer
Code 1142
Office of Naval Research
800 N. Quincy Street
Arlington, VA 22217-5000
Tel: (202) 696-4741

T. L. Martin
Lt. Col. U.S. Marine Corps
Aviation C2 Section Head
C2 Branch, C3 Division, MCDEC
131 Tackett's Mill Road
Stafford, VA 22554
Tel: (703) 540-3161

Andrej Martinovic
Project Manager
W. R. Grace & Company
1114 Avenue of the Americas
New York, NY 10036-7794
Tel: (212) 819-5860

Orin E. Marvel
Chief Scientist
Hughes Aircraft Company
MS 606/M236
P.O. Box 3310
Fullerton, CA 92634
Tel: (714) 732-5869

Thomas G. Mattoon
Engineer
Code 443
Naval Ocean Systems Center
271 Catalina Blvd.
San Diego, CA 92152-5000
Tel: (619) 225-2572

Israel Mayk
Electronics Engineer
U.S. Army CECOM
COMM/ADP Center
AMSEL-COM-AC
Fort Monmouth, NJ 07703
Tel: (201) 544-4996

William C. McDonald
Senior Staff Engineer
System Development Corporation
4810 Bradford Blvd.
Huntsville, AL 35805
Tel: (205) 837-7610

Thomas McGregor
McGregor Associates
105 Squire Hill Road
Longwood, Florida 32779
Tel: (305) 852-7235

Dennis R. Mensh
Professor of Physics
Code 61Mh
Naval Postgraduate School
Monterey, CA 93943
Tel: (408) 624-7060

John R. Miller
Sr. Staff Engineer
TRW Defense Systems Group
1 Space Park Drive
M/S 136/1238
Redondo Beach, CA 90278
Tel: (213) 516-9330

Peter D. Morgan
Managing Consultant
SCICON Ltd.
Abbey House
282 Farnborough Road
Farnborough GU14 7NA
UNITED KINGDOM
Tel: (0252) 541402

Prem K. Munjal
Project Manager
The Aerospace Corporation
2350 E. El Segundo Blvd.
Mail Station M5-643
El Segundo, CA 90245
Tel: (213) 648-6406

Bruce M. Nagy
Lt. USN
Electronic Test Engineer
Code 815
Naval Ocean Systems Center
271 Catalina Blvd.
San Diego, CA 92152-5000
Tel: (619) 225-6791

David Noble
Engineering Research Associates
8618 Westwood Center Drive
Vienna, VA 22180
Tel: (703) 734-8800

Jason Papastravou
Graduate Student
Massachusetts Institute of Technology
Room 35-407/LIDS
77 Massachusetts Avenue
Cambridge, MA 02139
Tel: (617) 253-3631

William Perrizo
Professor
Computer Science Dept.
300 Minard
North Dakota State University
Fargo, ND 58105
Tel: (701) 237-7248

Glenn E. Racine
Computer Scientist
AIRMICS
115 O'Keefe Bulding
Georgia Institute of Technology
Atlanta, GA 30332
Tel: (404) 894-3107

David Roberts
UNT Director
9311 Golodrinna
La Mesa, CA 92041
Tel: (619) 225-7701

Martha L. Robinette
Operations Research Analyst
Command-USACAORA
ATTN: ATOR-CAS-C
Ft. Leavenworth, KS 66027-5200
Tel: (913) 684-4309

Izhak Rubin
IRI Corporation
4544 Totana Drive
Tarzana, CA 91356
Tel: (818) 996-1698

Joan Ryder
Pacer Systems Inc.
300 Welsh Road
4 Horshan Business Center
Horsham, PA 19044

Richard M. Sabat
Member Technical Staff
The MITRE Corporation
MS-W660
1820 Dolley Madison Blvd.
McLean, VA 22102
Tel: (703) 883-6129

Nils Sandell, Jr.
President
ALPHATECH, Inc.
2 Burlington Executive Park
111 Middlesex Turnpike
Burlington, MA 01803
Tel: (617) 273-3388

Walter Schoppe
Technology Block Manager
Navigation and Aircraft C^3
Code 40B
Naval Air Development Center
Warminster, PA 18974-5000
Tel: (215) 441-2378

Daniel Serfaty
Graduate Student
Dept. of EE&CS
University of Connecticut
Box U-157
Storrs, CT 06268
Tel: (203) 486-3261

Lui Sha
Carnegie-Mellon University
Computer Science Department
Schenley Park - Wean Hall
Pittsburgh, PA 15213

J. Randolph Simpson
Scientific Officer
Code 111SP
Office of Naval Research
800 N. Quincy Street
Arlington, VA 22217
Tel: (202) 696-4324

Dana L. Small
Computer Scientist
Code 443
Naval Ocean Systems Center
271 Catalina Blvd.
San Diego, CA 92152-5000
Tel: (619) 225-7796

Yale Smith
Chief, Decision Aids Section
(COAD)
Rome Air Development Center
Criffiss AFB, NY 13441
Tel: (315) 330-7764

Michael G. Sovereign
Professor
Code 74
Naval Postgraduate School
Monterey, CA 93943-5000
Tel: (408) 646-2618

Joseph S. Stewart
Commander
Code 74
Naval Postgraduate School
Monterey, CA 93943
Tel: (408) 646-2618

Ingabee R. Stone
Captain, USAF
HQ SAC/SICCP
Offutt AFB, NE 68113
Tel: (402) 294-5932

Conrad W. Strack
SPC
1500 Wilson Blvd.
Arlington, VA 22201

Harold Szu
Research Physicist
Naval Research Laboratory
Code 5-709
Washington, D.C. 20375-5000
Tel: (202) 767-1493

Gail M. Sullivan
Computer Scientist
Code 443
Naval Ocean Systems Center
271 Catalina Blvd.
San Diego, CA 92152-5000
Tel: (619) 753-9109

Leonard P. Sweet
Head
Counter C3I Staff
Code 5709
Naval Research Laboratory
Washington DC 20375-5000
Tel: (202) 767-2304

Ricki Sweet
Adjunct Professor
Code 74
Naval Postgraduate School
Monterey, CA 93943-5000
Tel: (408) 646-2618

Debra M. Teasdale
Computer Scientist
Code 443
Naval Ocean Systems Center
271 Catalina Blvd.
San Diego, CA 92152-5000
Tel: (619) 225-7485

Doyle W. Thomas
Electrical Engineer
U.S. Army Strategic Defense Command
DASD-H-SB
P.O. Box 1500
Huntsville, AL 35807
Tel: (205) 895-3256

Paul Thompson
Senior Research Scientist
System Development Corporation
4810 Bradford Blvd.
Huntsville, AL 35805
Tel: (205) 837-7610

Hidiyaki Tokuda
Carnegie-Mellon University
Computer Science Dept.
Schenley Park - Wean Hall
Pittsburgh, PA 15213

Martin A. Tolcott
Senior Scientist
Decision Science Consortium, Inc.
7700 Leesburg Pike
Suite 421
Falls Church, VA 22043
Tel: (703) 790-0510

Andre van Tilborg
Professor
Computer Science Dept.
Carnegie-Mellon University
Schenley Park - Wean Hall
Pittsburgh, PA 15213

Willard S. Vaughan, Jr.
Code 1142
Office of Naval Research
800 N. Quincy Street
Arlington, VA 22217-5000
Tel: (202) 696-4741

James Walton
Graduate Student
Massachusetts Institute of Technology
Room 35-407/LIDS
77 Massachusetts Avenue
Cambridge, MA 02139
Tel: (617) 253-3631

Elbert J. Wells
Engineer
Code 443
Naval Ocean Systems Center
271 Catalina Blvd.
San Diego, CA 92152-5000
Tel: (619) 225-7485

Patrick W. Williams
Staff Engineer
Hughes Aircraft Co.
P.O. Box 3310
MS 618/Q311
Fullerton, CA 92634
Tel: (714) 732-7418

Gary Witus
Program Director
Vector Research, Inc.
P.O. Box 1506
Ann Arbor, MI 48106
Tel: (313) 973-9210

Derek Wong
Computer Scientist
Code 443
Naval Ocean Systems Center
271 Catalina Blvd.
San Diego, CA 92152-5000

A. E. R. Woodcock
Senior Consulting
Synectics Corporation
111 East Chestnut Street
Rome, NY 13440
Tel: (315) 337-3510

Ronald E. Wright
Chief Engineer
New Ventures Department
FERRANTI Computer Systems Ltd.
Western Road
Bracknell, Berkshire RG12 1RA
UNITED KINGDOM
Tel: (344) 483232 Ext. 3565

Al M. Zied
Code 443
Naval Ocean Systems Center
271 Catalina Blvd.
San Diego, CA 92152-5000
Tel:

Stanley Zionts
Professor
School of Management
State University of New York
Buffalo, NY 14260
Tel: (716) 636-3260

9th WORKSHOP ON C^3 SYSTEMS

JUNE 2 TO JUNE 5, 1986

PROGRAM

(AS OF APRIL 28, 1986)

Sponsored by

LABORATORY FOR INFORMATION AND DECISION SYSTEMS
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
CAMBRIDGE, MASSACHUSETTS 02139

and

MATHEMATICS PROGRAM
OFFICE OF NAVAL RESEARCH
UNDER CONTRACT ONR/N00014-77-C-0532

and in cooperation with

IEEE CONTROL SYSTEMS SOCIETY
Technical Committee on C^3

MONDAY MORNING, JUNE 2, 1986

SESSION 1: SURVEILLANCE, I ROOM: Ingersoll Hall - **122**

Chairman: M. Athans, MIT

8:00 - 3:00 P.M. REGISTRATION

8:30 - 9:00 A.M. WELCOME AND INTRODUCTION

M. Athans, LIDS/MIT
J. R. Simpson, ONR

9:00 - 9:30 A.M. MULTIPLE TARGET ESTIMATION USING MULTIPLE BEARING-ONLY SENSORS

P. R. Williams, Hughes Aircraft Company

9:30 - 10:00 A.M. AUTOMATIC TRACKING INITIATION IN AUTOMATED SURVEILLANCE SYSTEMS

H. A. Heidary, Hughes Aircraft Company

10:00 - 10:30 A.M. BREAK

10:30 - 11:00 A.M. PARALLELISM IN MULTITARGET TRACKING AND ADAPTATION TO MULTIPROCESSOR ARCHITECTURES

T. Kurien, T. G. Allen, and R. B. Washburn, Jr., ALPHATECH, Inc.

11:00 - 11:30 A.M. TRACKING IN DISTRIBUTED SENSOR NETWORKS

C-Y. Chong, K-C. Chang, and S. Mori, Advanced Decision Systems

11:30 - 12:00 P.M. REGISTRATION TECHNIQUES FOR MULTIPLE SENSOR SURVEILLANCE

M. P. Dana, Hughes Aircraft Company

12:00 - 1:30 P.M. LUNCH

MONDAY AFTERNOON, JUNE 2, 1986

SESSION 2: COMPUTER SYSTEMS ROOM: Ingersoll Hall - **122**

Chairman: O. Marvel, Hughes Aircraft Co.

1:30 - 2:00 P.M. DISTRIBUTED INFORMATION SYSTEM FOR FUTURE TACTICAL AIR CONTROL SYSTEMS

W. Perrizo, North Dakota State University

2:00 - 2:30 P.M. COMBAT SERVICE SUPPORT (CSS) CONTROL SYSTEM (CSSCS) RESEARCH

G. E. Racine, AIRMICS

2:30 - 3:00 P.M. REAL-TIME DECENTRALIZED OPERATING SYSTEM SOFTWARE FOR DISTRIBUTED COMMAND AND CONTROL

H. Tokuda, and E. D. Jensen, Carnegie Mellon University

3:00 - 3:30 P.M. BREAK

3:30 - 4:00 P.M. SCHEDULING SOFTWARE TASKS WITH HARD DEADLINES

J. P. Lehoczky, and L. Sha, Carnegie Mellon University

4:00 - 4:30 P.M. REAL-TIME DATA BASE MANAGEMENT

D. L. Small, NOSC

4:30 - 5:00 P.M. ARCHITECTURES FOR FUTURE COMMAND AND CONTROL SYSTEMS

O. E. Marvel, Hughes Aircraft Company

MONDAY AFTERNOON, JUNE 2, 1986

SESSION 3: C^3 THEORY, I   ROOM: Ingersoll Hall - 368

Chairman: I. Mayk, CECOM

1:30 - 2:00 P.M. **FORMAL DEFINITION OF INTEROPERABILITY PROTOCOLS**

P. D. Morgan, SCICON, Ltd.

2:00 - 2:30 P.M. **LANCESTER EQUATIONS, COMBAT SYSTEMS**

T. Woodcock, Synetics Corp.

2:30 - 3:00 P.M. **MODELING AND ANALYSIS OF MARKOVIAN MULTI-FORCE C^3 PROCESSES**

I. Rubin, IRI Corp.

3:00 - 3:30 P.M. BREAK

3:30 - 4:00 P.M. **A SYSTEMS THEORY APPROACH TO SURVEILLANCE AND C^3I**

S. Gardner, and F. Polkinghorn, Naval Research Lab.
R. Daves, Martingale Research Corp.

4:00 - 4:30 P.M. **NONLINEAR NONEQUILIBRIUM STATISTICAL MECHANICS APPROACH TO C^3 SYSTEMS**

L. Ingber, Naval Postgraduate School

4:30 - 5:00 P.M. **REAL TIME PARALLEL PROCESSING OF CONTROL STRATEGIES**

A. Martinovic, W. R. Grace & Co.

TUESDAY MORNING, JUNE 3, 1986

SESSION 4: BATTLE MANAGEMENT IN SDI   ROOM: Ingersoll Hall - 122

Chairman: D. W. Thomas, USASDC

8:00 - 3:00 P.M. REGISTRATION

8:30 - 9:00 A.M. **ALGORITHMS FOR SDI BATTLE MANAGEMENT**

D. W. Thomas, USASDC

9:00 - 9:30 A.M. **AN SDI BATTLE MANAGEMENT/C^3 TESTBED EXPERIMENT** (Film)

D. W. Thomas, USASDC

9:30 - 10:00 A.M. **A TESTBED FOR EVALUATING BM/C^3 ALGORITHMS**

W. C. McDonald, System Development Corp.

10:00 - 10:30 A.M. BREAK

10:30 - 11:00 A.M. **MODELING AND SIMULATION**

General Research Corp.

11:00 - 11:30 A.M. **RESOURCE MANAGEMENT FOR SDI**

N. R. Sandell, Jr., ALPHATECH, Inc.

11:30 - 12:00 P.M. **COST FUNCTION OPTIMIZATION**

P. Thompson, System Development Corp.

12:00 - 1:30 P.M. LUNCH

TUESDAY AFTERNOON, JUNE 3, 1986

SESSION 5: EXPERT SYSTEMS ROOM: Ingersoll Hall - **122**

Chairman: R. A. Dillard, NOSC

1:30 - 2:00 P.M. **EXPERT SYSTEM TECHNIQUES FOR RECONSTRUCTION AND POST-ANALYSES**

R. A. Dillard, NOSC

2:00 - 2:30 P.M. **INTELLIGENT DATA FUSION AND SITUATION ASSESSMENT**

W. L. Lakin, Admiralty Research Establishment

2:30 - 3:00 P.M. **KNOWLEDGE BASED HIERARCHICAL UNDERSTANDING SYSTEM FOR CRITICAL COMBAT NODE ANALYSIS**

F. D. Deffenbaugh, J. R. Miller, and J. H. Swaffield
TRW Defense Systems Group

3:00 - 3:30 P.M. BREAK

3:30 - 4:00 P.M. **EXPERT SYSTEMS FOR INTELLIGENCE ANALYSIS SUPPORT**

W. H. King, Hughes Aircraft Co.

4:00 - 4:30 P.M. **NAVINT: A NAVAL INTELLIGENCE ANALYST'S AID**

T. D. Garvey, J. D. Lowrance, and T. M. Strat, SRI International

4:30 - 5:00 P.M. **HUMAN COMPUTER INTERFACE FOR AN ADVANCED C^3 WORKSTATION**

S. Kelley, USAF

TUESDAY AFTERNOON, JUNE 3, 1986

SESSION 6A: HUMAN DECISIONMAKING, I ROOM: Ingersoll Hall - **368**

Chairman: M. Metersky, Naval Air Development Center

1:30 - 2:00 P.M. **A CHANGE IN SYSTEM DESIGN EMPHASIS-FROM MACHINE TO MAN**

M. Metersky, Naval Air Development Center

2:00 - 2:30 P.M. **VALIDATION RESULTS FOR A COMPUTERIZED ASW COMMANDER MODEL**

E. E. Entin, R. M. James, and J. C. Deckert, ALPHATECH, Inc.

2:30 - 3:00 P.M. **EMPIRICAL INVESTIGATION OF HUMAN FUNCTIONS IN DISTRIBUTED C^3 SYSTEMS**

D. Serfaty, D. L. Kleinman, and L. G. Bushnell, University of Connecticut

3:00 - 3:30 P.M. BREAK

TUESDAY AFTERNOON, JUNE 3, 1986

SESSION 6B: COMMUNICATIONS SYSTEMS ROOM: Ingersoll Hall - **368**

Chairman: F. Deckelman, SHAPE

3:30 - 4:00 P.M. **DEVELOPING C^3 SYSTEMS FOR THE NATO ENVIRONMENT**

F. Deckelman, SHAPE

4:00 - 4:30 P.M. **HYBRID LAN COMMUNICATION SYSTEM**

C. P. Carnes, Intermetrics, Inc.

4:30 - 5:00 P.M. **OPTIMUM TRANSMISSION RANGES AND THROUGHPUT OF MULTI-HOP SPREAD-SPECTRUM NETWORKS**

E. Geraniotis, University of Maryland

WEDNESDAY MORNING, JUNE 4, 1986

SESSION 7: C^3 THEORY, II ROOM: Presidio

Chairman: A. H. Levis, MIT

8:00 - 3:00 P.M. REGISTRATION

8:30 - 9:00 A.M. PERFORMANCE AND TIMELINESS IN COMMAND AND CONTROL ORGANIZATIONS

S. Andreadakis, and A. H. Levis, MIT

9:00 - 9:30 A.M. MATHEMATICAL MODELS OF DYNAMIC RESOURCE ALLOCATION

D. A. Castanon, and P. Luh, ALPHATECH, Inc.

9:30 - 10:00 A.M. MATHEMATICAL MODELS OF DISTRIBUTED INFORMATION PROCESSING

D. A. Castanon, ALPHATECH, Inc.

10:00 - 10:30 A.M. BREAK

10:30 - 11:00 A.M. DISTRIBUTED DETECTION WITH COSTLY COMMUNICATIONS IN A TWO-PERSON ORGANIZATION

J. Papastavrou, and M. Athans, MIT

11:00 - 11:30 A.M. MINIMAX ROBUST DISTRIBUTED DISCRETE-TIME SEQUENTIAL DETECTION IN UNCERTAIN ENVIRONMENTS

E. Geraniotis, University of Maryland

11:30 - 12:00 P.M. A POSSIBILISTIC APPROACH TO MODELING C^3 SYSTEMS

I. R. Goodman, NOSC

12:00 - 1:30 P.M. LUNCH

WEDNESDAY AFTERNOON, JUNE 4, 1986

SESSION 8A: PETRI NETS ROOM: Presidio

Chairman: R. R. Tenney, ALPHATECH, Inc.

1:30 - 2:00 P.M. A METHOD FOR MODELING C^3 SYSTEMS

L. C. Kramer, and R. R. Tenney, ALPHATECH, Inc.

2:00 - 2:30 P.M. ON THE DESIGN OF ALTERNATIVE C^3 ORGANIZATIONAL FORMS

P. Remy, A. H. Levis, and V. Jin, MIT

2:30 - 3:00 P.M. METHODOLOGY FOR EVALUATION OF C^3 SYSTEM CONFIGURATIONS

D. R. Edmonds, The MITRE Corp.

3:00 - 3:30 P.M. BREAK

3:30 - 4:00 P.M.

4:00 - 4:30 P.M.

4:30 - 5:00 P.M.

WEDNESDAY AFTERNNON, JUNE 4, 1986

SESSION 9: C^3 SYSTEM ANALYSIS, I ROOM: Vista

Chairman: R. Sweet, NPG

1:30 - 2:00 P.M. **AN EVOLVING C^2 EVALUATION TOOL - MCES: THEORY**

R. Sweet, and D. R. Mensh, NPG

2:00 - 2:30 P.M. **AN EVOLVING C^2 EVALUATION TOOL - MCES: APPLICATION**

D. R. Mensh and R. Sweet, NPG

2:30 - 3:00 P.M. **SYSTEM BOUNDING (C^3I MISSION ANALYSIS METHODOLOGY)**

B. R. Nagy, USN, NOSC

3:00 - 3:30 P.M. BREAK

3:30 - 4:00 P.M. **DISTRIBUTED TACTICAL C^2 SURVIVABILITY ANALYSIS**

F. A. Bausch, E-Systems, Inc.

4:00 - 4:30 P.M. **MEASUREMENT OF THE VALUE ADDED BY THE MANEUVER CONTROL SYSTEM**

P. Feld, Defense Systems Inc.

4:30 - 5:00 P.M. **FlyPAST: AN INTELLIGENT SYSTEM FOR NAVAL RESOURCE ALLOCATION**

J. A. Gadsden, Admiralty Research Establishment

THURSDAY MORNING, JUNE 5, 1986

SESSION 10: HUMAN DECISIONMAKING, II ROOM: Presidio

Chairman: G. Malecki, ONR

8:30 - 9:00 A.M. **HUMAN PERFORMANCE AND MATHEMATICAL MODELING IN C^2: WHERE IS THE DISCONNECT?**

M. A. Tolcott, Decision Science Consortium, Inc.

9:00 - 9:30 A.M. **PRELIMINARY ANALYSES OF A GENERIC CONSOLE/DATABASE SYSTEM FOR AFLOAT BATTLE GROUP COMMANDERS**

R. A. Fleming, Navy Personnel R & D Center

9:30 - 10:00 A.M. **SCHEMA BASED DECISIONMAKING**

D. Noble, Engineering Research Associates

10:00 - 10:30 A.M. BREAK

10:30 - 11:00 A.M. **EXPERIMENTAL PERFORMANCE ANALYSIS OF TEAM DECISIONMAKING**

D. Serfaty, E. E. Entin, D. L. Kleinman, ALPHATECH, Inc.

11:00 - 11:30 A.M. **AN EXPERIMENTAL INTERVIEW SYSTEM FOR MULTIPLE INTERACTING DECISION MAKERS**

R. L. Stewart, and B. W. Hamill, JHU, Applied Physics Lab

11:30 - 12:00 P.M. **THE DEFINITION, IMPLEMENTATION, AND CONTROL OF AGENTS IN AN INTERVIEW SYSTEMS FOR DISTRIBUTED TACTICAL DECISION MAKING**

J. M. Gilbert, and R. L. Stewart, JHU Applied Physics Lab

12:00 - 1:30 P.M. LUNCH

THURSDAY AFTERNNON, JUNE 5, 1986

SESSION 11: SURVEILLANCE, II ROOM: Presidio

Chairman: H. Szu, NRL

1:30 - 2:00 P.M. **A COMPARISON OF MANUAL AND AUTOMATIC DATA ASSOCIATION FACILITIES IN COMMAND AND CONTROL**

G. Brander, Admiralty Research Establishment

2:00 - 2:30 P.M. **APPLICATIONS OF FAST SIMULATED ANNEALING ALGORITHM EMBEDDED IN NEURAL NETWORK ARCHITECTURE TO SDI SURVEILLANCE PROBLEMS**

H. Szu, NRL

2:30 - 3:00 P.M. **A PERSPECTIVE ON MOE'S FOR WIDE AREA SURVEILLANCE**

L. Sweet, NRL

3:00 - 3:30 P.M. BREAK

3:30 - 4:00 P.M. **AIR FORCE BATTLE MANAGEMENT: FROM A DATA TO A KNOWLEDGE WORLD**

Y. Smith, RADC/COAD

4:00 - 4:30 P.M. **EXPERIMENTAL RESEARCH ON COGNITIVE PROCESSES IN TACTICAL DECISION MAKING**

M. Cohen, Decision Sciences Consortium, Inc.

4:30 - 5:00 P.M. **A FAST CONVERGING REAL TIME ADAPTIVE NOISE CANCELLER**

M. El-Sharkawy, and M. Aburdene, Bucknell University

THURSDAY AFTERNOON, JUNE 5, 1986

SESSION 12: C^3 SYSTEMS ANALYSIS, II ROOM: Vista

Chairman: I. Mayk, CECOM

1:30 - 2:00 P.M. **NEW CONCEPTS IN BRL ADDCOMPE FIRE SUPPORT APPLICATION**

S. C. Chamberlain, US Army Ballistic Research Lab

2:00 - 2:30 P.M. **EMPIRICAL DATA ON COMMANDER/STAFF INTERACTIONS IN COMMAND POST EXERCISES**

G. Witus, Vector Research, Inc.

2:30 - 3:00 P.M. **IMPACT OF C^3I ON THE OVER-THE-HORIZON-TARGETING ON TASM**
(An Example of Mission Systems Engineering)

R. M. Sabat, The MITRE Corp.

3:30 - 3:30 P.M. BREAK

3:30 - 4:00 P.M. **DYNAMIC STOCHASTIC C^3 SYSTEM MODELS AND THEIR PERFORMANCE EVALUATION**

I. Rubin, IRI Corp.
I. Mayk, CECOM

4:00 - 4:30 P.M. **AN EXPERIMENTAL COMMAND SYSTEM FOR FORCE LEVEL ANTI-SUBMARINE WARFARE**

C. J. Gadsden, Admiralty Research Establishment

4:30 - 5:00 P.M. **MOVEPLAN**

M. L. Robinette, USACAORA

AUTHOR INDEX