HYBRID INTELLIGENT SYSTEMS INTEGRATION INTO
COMPLEX MULTI-SOURCE INFORMATION SYSTEMS

by

Raymond Louie

Submitted to the Department of Electrical Engineering and Computer Science
In Partial Fulfillment of the Requirements for the Degree of
Masters of Engineering in Electrical Engineering and Computer Science
at the Massachusetts Institute of Technology

August 13, 1999

Copyright 1999 Massachusetts Institute of Technology. All rights reserved.
The author hereby grants to M.I.T. permission to reproduce and
distribute publicly paper and electronic copies of this thesis
and to grant others the right to do so.
HYBRID INTELLIGENT SYSTEMS INTEGRATION INTO COMPLEX MULTI-SOURCE INFORMATION SYSTEMS

by

Raymond Louie

Submitted to the Department of Electrical Engineering and Computer Science

August 13, 1999

In Partial Fulfillment of the Requirements for the Degree of Masters of Engineering in Electrical Engineering and Computer Science

at the Massachusetts Institute of Technology

ABSTRACT

A recent innovation in artificial intelligence research has been the integration of multiple artificial intelligence techniques into hybrid intelligent systems. Hybrid artificial intelligent systems seek to overcome the deficiencies of traditional artificial techniques by combining techniques with complementary capabilities. Using these hybrid systems, researchers have been able to solve a variety of complex classes of problems otherwise intractable by use of individual systems. This research examines the state of the art in hybrid intelligent systems that combine such popular techniques as expert systems, case-based reasoning systems, fuzzy logic, neural networks, genetic algorithms. Comparisons and conclusions concerning the general capabilities of such systems are given. Additionally a case study is conducted of a complex multi-source information system. The applicability of hybrid artificial intelligence techniques to optimize the system is ascertained and a hybrid artificial intelligent system design proposed and justified.

Thesis Supervisor: Amar Gupta

Title: Co-Director, Productivity from Information Technology (PROFIT) Initiative. MIT School of Management
ACKNOWLEDGEMENTS

I would like to take the opportunity to thank the following people for their assistance and support:

DR. AMAR GUPTA
For his supervision of the thesis and his willingness to support my last minute revisions

JAMES VAN ZANDT
A JTE engineer who provided me with invaluable information at the end of a long day

MY FAMILY
For all their patience and support in seeing me through MIT

MY FRIENDS
For the wonderful memories I will treasure forever. Erwin, Ed, Felicia, Clare, David, Joe, Bob, Karen, Sharon, Theresa, Dave, Brooke, Don, and especially Janet. I’ll never forget what you did for me in my direst hours. Thank you.

ALL THE OTHER PEOPLE I’VE FORGOTTEN TO MENTION
You’ve made your mark on my life – I’m just too tired now to realize…
ACKNOWLEDGEMENTS .................................................................................................................... 3

LIST OF TABLES AND FIGURES .......................................................................................................... 10

1.0 INTRODUCTION .......................................................................................................................... 12

1.1 SIGNIFICANCE OF ARTIFICIAL INTELLIGENCE ................................................................... 12

1.2 THESIS ROADMAP .................................................................................................................. 13

2.0 TRADITIONAL AI TECHNIQUES .................................................................................................. 16

2.1 SYMBOLIC AND CONNECTIONIST/NUMERICAL TECHNIQUES ............................................. 16

2.1.1 Symbolic .............................................................................................................................. 16

2.1.2 Connectionist/Numeric ....................................................................................................... 16

2.1.3 Information Processing Spectrum ..................................................................................... 17

2.2 EXPERT SYSTEMS ...................................................................................................................... 18

2.2.1 Overview ............................................................................................................................. 18

2.2.2 Advantages ........................................................................................................................ 20

2.2.3 Disadvantages ................................................................................................................... 21

2.2.4 Conclusions ....................................................................................................................... 22

2.3 CASE-BASED REASONING ...................................................................................................... 23

2.3.1 Overview ............................................................................................................................. 23

2.3.2 Advantages ........................................................................................................................ 24

2.3.3 Disadvantages ................................................................................................................... 24

2.3.4 Conclusions ....................................................................................................................... 25

2.4 FUZZY LOGIC ............................................................................................................................ 25

2.4.1 Overview ............................................................................................................................. 25

2.4.2 Advantages ........................................................................................................................ 28

2.4.3 Disadvantages ................................................................................................................... 28
2.4.4 Conclusions .......................................................................................................................... 28

2.5 NEURAL NETWORKS ...................................................................................................................... 29
  2.5.1 Overview ............................................................................................................................... 29
  2.5.2 Advantages ............................................................................................................................ 31
  2.5.3 Disadvantages ....................................................................................................................... 32
  2.5.4 Conclusions .......................................................................................................................... 32

2.6 GENETIC ALGORITHMS ............................................................................................................. 32
  2.6.1 Overview ............................................................................................................................... 32
  2.6.2 Advantages ............................................................................................................................ 34
  2.6.3 Disadvantages ....................................................................................................................... 34
  2.6.4 Conclusions .......................................................................................................................... 35

2.7 COMPARISON OF BASIC ARTIFICIAL TECHNIQUES ..................................................................... 35

3.0 HYBRID AI TECHNIQUES ........................................................................................................... 37

3.1 INTRODUCTION TO HYBRID SYSTEMS ..................................................................................... 37

3.2 METHODS OF COMBINING HYBRID SYSTEMS ............................................................................. 38
  3.2.1 Stand-alone ........................................................................................................................... 38
  3.2.2 Transformational .................................................................................................................... 39
  3.2.3 Semi-Integrated ..................................................................................................................... 40
  3.2.4 Fully integrated ...................................................................................................................... 41

3.3 FUTURE OF RESEARCH AND DEVELOPMENT ............................................................................ 43

SURVEY OF HYBRID SYSTEMS ...................................................................................................... 43

3.4 EXPERT SYSTEMS AND NEURAL NETWORK HYBRIDS .................................................................... 43
  3.4.1 Complementary characteristics .......................................................................................... 43
  3.4.2 Methods of integration .......................................................................................................... 44
  3.4.3 Advantages ............................................................................................................................ 47
  3.4.4 Disadvantages ....................................................................................................................... 47
  3.4.5 Conclusions .......................................................................................................................... 47
3.5 Expert Systems and Fuzzy Logic Hybrids ................................................................. 48
  3.5.1 Complementary characteristics ........................................................................ 48
  3.5.2 Methods of Integration .................................................................................... 48
  3.5.3 Advantages ....................................................................................................... 50
  3.5.4 Disadvantages ................................................................................................. 51
  3.5.5 Conclusions ..................................................................................................... 51
3.6 Expert Systems and Genetic Algorithm Hybrids ....................................................... 52
  3.6.1 Complementary Characteristics ....................................................................... 52
  3.6.2 Methods of Integration .................................................................................... 52
  3.6.3 Conclusions ..................................................................................................... 52
3.7 Neural Networks and Fuzzy Logic Hybrids ............................................................. 53
  3.7.1 Complementary Characteristics ....................................................................... 53
  3.7.2 Methods of integration .................................................................................... 54
  3.7.3 Conclusions ..................................................................................................... 56
3.8 Neural Networks and Genetic Algorithm Hybrids ..................................................... 56
  3.8.1 Complementary Characteristics ....................................................................... 56
  3.8.2 Methods of Integration .................................................................................... 57
  3.8.3 Conclusions ..................................................................................................... 57
3.9 Expert System and Case-based Reasoning Hybrids .................................................. 58
  3.9.1 Complementary Characteristics ....................................................................... 58
  3.9.2 Methods of Integration .................................................................................... 58
  3.9.3 Conclusions ..................................................................................................... 59
3.10 Neural Networks and Case-based Reasoning Hybrids ............................................. 59
  3.10.1 Complementary Characteristics ..................................................................... 59
  3.10.2 Methods of Integration .................................................................................. 60
  3.10.3 Conclusions ................................................................................................... 60
3.11 Fuzzy Logic and Case-based Reasoning Hybrids .................................................... 60
  3.11.1 Conclusions .................................................................................................. 60
7.3.6 Components .......................................................................................................................... 81
7.3.8 TCTA .................................................................................................................................... 82

8.0 AREAS FOR IMPROVEMENT ................................................................................................ 83
8.1 Imagery ...................................................................................................................................... 83
8.2 Track Data .................................................................................................................................. 83

9.0 RECOMMENDED SYSTEM ................................................................................................ 85
9.1 Design ......................................................................................................................................... 85
9.2 Hybrid Design Process ............................................................................................................. 85
   9.2.1 Main characteristics and applicable techniques ................................................................. 85
   9.2.3 Selection of optimal techniques .......................................................................................... 87
   9.2.4 Hybridization and integration level ..................................................................................... 88
   9.2.5 Total solution ..................................................................................................................... 88
9.3 Implementation Details ............................................................................................................ 88
   9.3.1 Overview ............................................................................................................................ 88
   9.3.2 Interfacing with JTE system ............................................................................................... 90
   9.3.3 Fuzzy expert agents ......................................................................................................... 90
   9.3.4 Numerical module ............................................................................................................ 92
   9.3.5 Labeling module ............................................................................................................... 92
9.4 Justification .............................................................................................................................. 93
   9.4.1 ATR module ..................................................................................................................... 93
   9.4.2 Fuzzy expert agent module ............................................................................................... 93
9.6 Significance of Case Study ..................................................................................................... 94

10.0 CONCLUSIONS AND FURTHER RESEARCH ................................................................. 95
10.1 Conclusions ............................................................................................................................. 95
   10.1.1 Background research ..................................................................................................... 95
   10.1.2 Case study ..................................................................................................................... 96
10.2 Further Research ...................................................................................................................... 96
LIST OF TABLES and FIGURES

FIGURE 1: HUMAN INFORMATION PROCESSING SPECTRUM 15
FIGURE 2: EXPERT SYSTEM 16
FIGURE 3: CASE-BASED REASONING SYSTEM 22
FIGURE 4: MEMBERSHIP FUNCTIONS USING LINGUISTIC VARIABLES 25
FIGURE 5: FUZZY INFERANCE SYSTEM 26
FIGURE 6: ARTIFICIAL NEURAL NETWORK 29
FIGURE 7: BASIC ALGORITHM FOR GENETIC ALGORITHMS 33
FIGURE 8: EXPERT NETWORK 40
FIGURE 9: OVERVIEW OF MULTIPLE TARGET RECOGNITION SYSTEM 44
FIGURE 10: CONNETIONIST EXPERT SYSTEM FOR MEDICAL DIAGNOSIS 46
FIGURE 11: ADAPTIVE FUZZY EXPERT SYSTEM 48
FIGURE 12: MEMBERSHIP FUNCTIONS FOR DEGREES OF SPEED 49
FIGURE 13: FUZZY NEURON 55
FIGURE 14: PHASES OF JOINT TARGETING CYCLE 72
FIGURE 15: TARGET CLASSIFICATION 73
FIGURE 16: CONTROL STRUCTURE 77
FIGURE 17: JTE EPOCHS 80
PART I - FUNDAMENTALS

1.0 Introduction

Human beings process information using a variety of intelligent techniques. We can reason for extended periods of time at high levels concerning what choices to make. We can also recognize and react almost instantly to sensory cues. This type of hybrid intelligence has inspired artificial intelligence researchers to combine artificial techniques that represent diverse models and techniques for dealing with knowledge in an attempt to simulate human thought. Collectively these systems are called hybrid artificial intelligent systems [1].

The ultimate goal of this thesis is to evaluate the possibility of using a hybrid artificial intelligent system to facilitate a complex multi-source information system. This thesis is broken up into two main sections: background research on artificial intelligence and hybrid artificial intelligent systems, and specific research toward applying a hybrid system to a real world system.

1.1 Significance of Artificial intelligence

Artificial intelligence is the study of computations that make it possible to perceive, reason, and act [5]. The study can be classified further by the goal of the specific research. One goal is a scientific goal, which seeks to discover which ideas about representing knowledge, using knowledge, and assembling systems can explain various sorts of intelligence. Research in this area tends to focus on discovering new techniques and models for representing and using knowledge. The other goal is an
engineering goal, which seeks to solve real world problems utilizing artificial intelligence ideas and systems. Research in this area tends to focus on applying current artificial techniques rather than discovering new methods. Often the two goals are satisfied in one project, but even in those cases, one goal is more highly emphasized.

This research will focus on the latter engineering goal. Though some purely scientific advancement may be made in terms of hybrid system design, the design of the hybrid system is not intended to gain knowledge about new representations of knowledge. The main goal of this project is to examine an engineering problem and apply the most effective artificial intelligence techniques available. Consequently, the most practical, robust artificial intelligence techniques, hybrid system designs, and target identification techniques will be used in preference over the more experimental methods.

Artificial intelligence has evolved to the point where it is being used in many facets of technology. Often the systems are so well integrated, they can hardly be recognized as artificial intelligence without close examination. Anti-lock brakes, intelligent microwave settings, modern thermostat controls, and grammar checking tools are some examples of highly integrated artificial intelligent systems. The success of these and other intelligent systems has led to increased interest in incorporating artificial intelligence techniques into more complex systems.

1.2 Thesis Roadmap

The first section begins with a brief overview of the main artificial intelligence techniques in use at this time. These techniques form the building blocks for the vast
majority of hybrid systems being researched and include expert systems, neural networks, case-based reasoning, fuzzy logic, and genetic algorithms. Following the overview is a review of the major types of hybrid system in use. This review focuses on the complementary characteristics of the artificial intelligence techniques and how they can best be combined. At the end of the review, the hybrid system types are compared and their capabilities tabulated.

The second half of this thesis aims to explore the possibility of using hybrid artificial intelligence techniques to aid in the process of detecting, identifying, and destroying or disabling time-critical targets (TCT) during military conflict. Time-critical targets are targets whose destruction or disablement represents a significant tactical gain for friendly forces and whose attack window of opportunity is limited. The specific sub-problem that the latter half of this thesis seeks to address is improving the identification stage.

The structure of the second part of this thesis begins with a brief background of the sponsoring corporation and its goals. Following this is a description of the time-critical target (TCT) problem with which the corporation has been commissioned to investigate. Next the project with which the hybrid system would interface is explained in detail.

Specific areas for possible improvement are then determined for the target system. Next, artificial intelligence techniques that address these deficiencies are discussed.
Finally, a hybrid system is suggested which combines the most applicable techniques. Justifications for this design and details of implementation are then addressed. The analysis is completed with conclusions and suggestions for further research.
2.0 Traditional AI Techniques

2.1 Symbolic and Connectionist/Numerical Techniques

The study of artificial intelligence can be divided into two basic models of knowledge: symbolic and connectionist/numerical [2]. Most artificial intelligence research can be classified as pursuing advancement of techniques in one area, the other, or both.

2.1.1 Symbolic

The symbolic representation of knowledge is based on the work of Newell and Simon [3]. In their paper, they propose that the human mind is a serial information processing machine using production rules [2]. According to their theory, the human mind stores knowledge in the form of symbol structures. Each symbol structure consists of symbols connected by relations. For example, the symbol structure for a chair may contain the symbols seat and legs, and the relation “seat on top of legs”. Problem solving, then, involves a goal-directed search through the knowledge structures, using production-like (If-Then) rules [2].

2.1.2 Connectionist/Numeric

Connectionist or numeric techniques are based on the Parallel Distributed Processing Models of McClelland [4]. In contrast to the symbolic model, connectionist models view the human thought process as involving multiple constraints that take place through the interactions of a large number of simple processing elements that send and receive excitatory and inhibitory signals [2]. Instead of knowledge being located in one
memory structure which is accessed as needed, knowledge is thought to be distributed throughout the network. Numeric models similarly do not model knowledge as distinct from the reasoning mechanism and combine the two into one numerical technique. Genetic algorithms are such an example.

2.1.3 Information Processing Spectrum

Figure 1. Human Information Processing Spectrum [2]

Fuzzy
Symbolic ← Analytical
Interpretive
Deliberate
Hard Constraints
→ Interpretive
Soft Constraints
Automated
Non-interpretive
Soft Constraints
Numeric

Symbolic methods capture knowledge processes such as planning and deduction well, but do not represent problem solving which involves multiple constraints, such as recognizing a word [2]. Connectionist or numeric techniques, however, excel at problems such as pattern recognition or optimization, but have difficulty representing complex thought processes. Symbolic and connectionist/numeric theories, therefore, can be seen as representing two ends of the human information-processing spectrum (see Figure 1). Symbolic techniques represent the analytical, interpretive and deliberate processing behavior. Connectionist and numeric systems represent the intuitive, non-interpretive automated processing behavior. Along this spectrum there are a variety of intelligence models, which will be discussed in the following sections.
2.2 Expert Systems

2.2.1 Overview

Expert systems are a class of symbolic processing models that function by applying production rules to a knowledge base. An expert system consists of the following pieces: knowledge base, inference engine, database of facts and conclusions, explanation facility, and a user interface [1]. These are diagrammed in Figure 2.

Figure 2. Expert System [2]

The knowledge base consists of a set of production rules, generally in the form of IF-THEN rules. The IF portion of the rule specifies the antecedent or conditional clause, whereas the THEN portion represents the consequent or action clause [2].

The database component serves as a repository for information that the expert system can use to reach conclusions based on rules in the knowledge base [2]. It differs from the knowledge base in that the knowledge base contains information in the form of rules, but the database contains information in the form of declarative facts. These facts
are used to satisfy antecedents in rules to activate the consequent clauses. Consequent clauses in turn generate deductions that are stored in the database as new facts.

The inference mechanism operates by comparing data in the database with rules in the knowledge base and deciding which rules in the knowledge base apply to the data [2]. There are two modes in which expert systems may draw conclusions: forward chaining and backward chaining.

Forward chaining works as follows [5]. The system collects all input facts and places these values into a data set called the working set of knowledge. Then, the system checks through all the rules in the rule set. Should the IF clause of a rule be satisfied by facts in the working set of knowledge, the rule is said to fire and the THEN clause is executed. This most often adds deductions to the working set of knowledge. Once all rules have been examined, the system cycles through the rules again, using the newly modified working set of knowledge. This process continues until a cycle occurs where no rules are fired. The system then outputs the conclusions it has drawn from the initial input.

Backward chaining works in a reverse fashion [5]. The system begins by choosing a hypothesis, and begins trying to satisfy rules that conclude that hypothesis. To satisfy these rules, the system asks for user input about the facts necessary to satisfy IF clauses of those rules. Should a fact be unknown to the user, the system then seeks to satisfy rules that conclude that fact. If no rules are then found which satisfy that fact, the
fact is assumed false and the rule can not be verified, ending that line of reasoning. If all lines of reasoning leading to the hypothesis fail, the hypothesis can not be concluded and a new hypothesis is chosen [2]. If any line of reasoning leading to the hypothesis is satisfied, the hypothesis is proven.

Forward and backward chaining are not mutually exclusive, and many modern expert systems use both forward and backward chaining in their inference engines [5].

The explanation facility of an expert system allows it to answer user inquiries of how a conclusion was drawn. Generally, it uses a trace of rules fired to provide reasons for the decisions made [2]. The user interface acts as a liaison between the user and the explanation and inferencing mechanisms.

2.2.2 Advantages

Expert systems provide several major benefits. First, by separating the rule base from the inferencing engine, expert systems separate the knowledge from the reasoning mechanism. This immediately makes the system modular and thereby highly maintainable [1]. The rule base allows new rules to be added simply and efficiently as new knowledge is accumulated. The inference engine supports explanation, whereby the user can trace back the reasoning of the expert system. This allows checks on the logic process of the system.

Once the expert system has been developed, they are often less expensive to solve problems with than with a real expert [6]. Expert systems can be run continuously
without breaks and can offer advice in almost any location. If only a few human experts exist in a domain, expert systems provide efficient leverage of their abilities [6].

2.2.3 Disadvantages

There are disadvantages, however, to using expert systems. Foremost among these is the acquisition of knowledge from experts. This can be a tedious process and often it is uncertain if all the important rules have been included [2]. Knowledge engineering tends to be ad hoc; there are no scientific techniques to guide knowledge engineering [6]. As a result, many experts do not wish to undergo the lengthy knowledge acquisition process [9].

Another factor is the growth of rules. As the expert system gains capabilities to reason about more concepts, the number of rules in the system quickly becomes unmanageable. Thus, expert systems excel in narrow, well-defined domains but are ill suited to reasoning about vague or dissimilar concepts [1]. Complex problems where the number of combinatorial possibilities is large or the solution is non-deterministic are generally not solvable by expert systems [2]. Most pattern matching problems can not be solved by expert systems either [9].

Scientifically, expert systems do not offer much insight into how the human brain works. Instead expert systems represent an implementation of logical reasoning using stored knowledge and user input. This most likely does not capture the true thought processes of experts, especially such aspects as intuition [9].
Also, expert systems can not devise new solutions on the fly. If a solution does not exist in the rules of the knowledge base, the system fails [6]. Incomplete or incorrect data causes problems for many expert systems [2]. Additionally, since rule chaining generally is a tedious process, expert systems may not be fast enough for embedding in real-time applications [2].

2.2.4 Conclusions

Expert systems have been used in a variety of engineering projects including aerospace applications, design, diagnosis, and monitoring of engineering systems, interpretation of data, and prediction of events [6]. Two famous expert systems are MYCIN [7], an expert system to diagnosis infectious diseases, and XCON [8], a configuration system for Digital Equipment Corp.

Expert systems are suitable for specific types of problems. Jain and Jain [6] identify several key requirements for expert system use in a problem:

1) Symbolic reasoning problem
2) Problem can be solved using heuristics
3) An expert is willing to work on project
4) Conventional programming techniques are not satisfactory

In conclusion, expert systems provide a simple, flexible, robust platform to develop an intelligent system. Expert systems work best with concise systems with exact inputs that lead to logical outputs. Expert systems should not be used to solve vague or generic problems which could both cause the knowledge acquisition time or the rule base
to become unmanageable. Although expert systems should not be used for real-time applications, higher level decision making may still benefit from expert system use.

2.3 Case-Based Reasoning

2.3.1 Overview

Case-based reasoning systems are a form of symbolic knowledge representation based on the principle of storing past experiences or cases and retrieving relevant ones to aid in solving a current problem [2]. A case-based reasoning system consists of a case base, an indexing scheme, a case retrieval mechanism, and a case adaptation mechanism [1]. These are diagramed in Figure 3.

Figure 3. Case-based reasoning system [1]
The case base contains cases consisting of problem-solution pairs [9]. These cases are indexed using the indexing scheme, generally based on specifications of the problem. The case-based system operates by applying the indexing scheme to the input problem, generating a key. The retrieval mechanism then searches through the case base for cases whose index closely matches this key [2].

Once the relevant cases are found, the case adaptation mechanism notes differences between the specifications of the retrieved cases and the specification of the current case. If the differences are significant, the case-based reasoning system uses an adaptation algorithm to generate alternatives to the cases [2]. The resulting solutions are presented to the user.

2.3.2 Advantages

Case-based reasoning is ideally suited when rules for the knowledge domain are difficult to obtain or the number and complexity of the rule base is too large for alternative logical, analytical systems [1]. This method is also useful when a large number of precedents exists to store in the case database. Knowledge acquisition is much easier than expert systems, where tedious rules must be made for each case [2]. Unlike expert systems, case-based systems can be created when experts in an area do not exist [9].

2.3.3 Disadvantages

Case-based reasoning systems are hampered by the need for a fast, efficient, flexible indexing system [9]. If implemented incorrectly or if the case base is very large,
the exhaustive search can become slow [2]. Like expert systems, case-based reasoning systems can not handle complex problems where the number of combinatorial possibilities is large or the solution is non-deterministic in nature [2].

Also, algorithms for modifying solutions can vary greatly and can be difficult to design [9]. Capturing deep knowledge may be difficult as well; generally case-based reasoning systems only do one level of logic – matching indices. This prevents complex logical relations from being incorporated into these systems without the addition of other techniques, such as expert systems.

2.3.4 Conclusions

In summary, case-based reasoning should be used in situations where rules are hard to discern or unwieldy in number [1]. A large set of problem-solution pairs must exist to train the system, and reasonable algorithms should be available to modify past solutions for new problems [2]. A fast, efficient indexing system must be developed to keep system speed within bounds as well [9]. When all of these conditions are met, case-based reasoning systems can be a valuable tool in artificial intelligence applications. Numerous case-based applications exist for domains such as law, design, and medicine, all of which meet these requirements [1], [2].

2.4 Fuzzy Logic

2.4.1 Overview

Fuzzy logic was pioneered by Lotfi Zadeh to provide a mathematical basis for human reasoning [1]. Fuzzy logic provides a means of dealing with inexactness,
imprecision, and ambiguity [2]. Instead of using mathematical equations, fuzzy logic relies on imprecise concepts [1] such as “hot”, “big”, “fat”, “apply strong force”, and “high angular velocity”.

Figure 4. Membership functions using linguistic variables

Fuzzy logic encapsulates this concept by reasoning with fuzzy values. These values are determined using the concept of membership within a set of values (see Figure 4). For any given value, the membership function tells to what degree a value is described by the fuzzy term for the set [2]. A fuzzy value may have membership in one or more categories. The degree of membership is generally rated on a scale of zero to one: zero meaning no membership, one meaning total membership, and values between meaning partial membership [6]. Rules are used to link fuzzy input and output variables, much as the rules in an expert system link facts to conclusions [1].
Table 1. Example of a fuzzy associative map [1]

<table>
<thead>
<tr>
<th>Speed</th>
<th>Cold</th>
<th>Warm</th>
<th>Hot</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low</td>
<td>Large increase</td>
<td>Small increase</td>
<td>Small decrease</td>
</tr>
<tr>
<td>Medium</td>
<td>Medium increase</td>
<td>No change</td>
<td>Medium decrease</td>
</tr>
<tr>
<td>High</td>
<td>Small increase</td>
<td>Small decrease</td>
<td>Large decrease</td>
</tr>
</tbody>
</table>

One convenient method to represent and design rules is a fuzzy associative map (see Table 1). The column and row labels give possible fuzzy values and the corresponding conclusions to make or actions to take. Generally, rules can be executed with different strengths depending on the strength of the inputs.

Figure 5. Fuzzy inference system [2]

A typical structure of a fuzzy inference system is shown in Figure 5. If input values are not fuzzy, a fuzzification module transforms the data into fuzzy values using membership functions [2]. Fuzzy rules are then applied to the values, which produces
fuzzy actions or conclusions. If necessary, the fuzzy output is transformed back into crisp values using defuzzification functions [2].

2.4.2 Advantages

Fuzzy systems provide several unique benefits. First, the use of fuzzy terms allows users to input imprecise terms and use them to produce fuzzy or precise advice [9]. Since fuzzy systems often use linguistic terms as basic building blocks for rules, every day language can be used for knowledge acquisition [2], speeding up the process. This also makes understanding the output of the system simpler, since a user can easily read the rules [9]. Through the use of fuzzy membership sets, fuzzy systems are useful for handling decision making problems that are not easily defined by practical methods [2].

2.4.3 Disadvantages

Disadvantages to fuzzy systems include the added burden of tuning and optimizing membership functions and rules [2]. Since the final decision is combination of mathematical techniques and rules, the results are not as easy to explain as in an expert system [9]. Additionally, the unmanageable rule growth issue of expert systems hampers fuzzy logic systems as more rules are added to capture complex knowledge [6].

2.4.4 Conclusions

The field of fuzzy logic lies somewhere along the boundary between symbolic and connectionist. In some respects, fuzzy logic systems can implement high-level logic along the lines of fuzzy expert systems. However, fuzzy logic can also deal with imprecise or incomplete data [9] similar to connectionist technologies.
Fuzzy logic has been applied in many control systems, including automobile, business, consumer electronics, diagnosis, engineering and industrial control [6]. Khosla [2] identifies key characteristics for their use:

1) The system is non-linear
2) Mathematical methods do not exist or are difficult to build
3) Parameters to the system change
4) Sensor accuracy is a problem
5) The system may receive conflicting or uncertain input but must make correct decisions

Overall, fuzzy systems present an alternative to logical, analytical systems such as expert systems. Fuzzy systems tend to function in a manner more akin to everyday experiences of humans as decision-makers. When exact values or exact reasoning is required, it is best to use expert systems or other more numerical analytical methods. However, in cases where input is imprecise or no precise mathematical relations between input and output are available, fuzzy systems may provide a viable alternative.

2.5 Neural Networks

2.5.1 Overview

Artificial neural networks are connectionist models of thought motivated by the study of the human brain [2]. They consist of nodes, which mimic the functions of a biological neuron though simple mathematical functions [1].
Neural network nodes are interconnected to form an architecture with input, hidden, and output nodes (see Figure 6). Not all nodes in each layer need be connected to every node in the next. Neural network nodes receive input, generally in the form of a numerical value and typically apply a sigmoid mathematical function that produces an output value [1]. Depending on whether this output value meets or exceeds a threshold value, the node transmits the value to one or more neighboring nodes as input values. The transmission between these nodes is modified by numerical weighting factors that increase or decrease the magnitude of the transmitted value. This process continues until the output nodes attain a stable value, which is returned as the result of the neural network computation [5].

Neural networks can be classified into two general categories based on how they are trained: supervised training and unsupervised training [2]. Supervised neural
networks are trained by input and target output patterns. The input is fed into the network and the resulting output is compared to the target output. Through numerical methods such as back-propagation, the weights of the network are adjusted until the network output matches the target output. Examples of supervised neural networks are the single layer perceptron and the multi-layer perceptron [2].

Unsupervised neural networks are trained by input patterns only [2]. Thus the network has no target output values to match. Instead, the network learns features of the input and reflects these features in the output, basically classifying input. Examples of unsupervised neural networks are Kohonen neural networks and Radial Basis Function neural networks [2].

2.5.2 Advantages

Neural networks benefit from the ability to use data to train the network as opposed to lengthy knowledge acquisition [6], [9]. Another benefit of neural networks is the ability to make sense of noisy or incomplete data [1], [9], [10]. The network may still reach the same output values if only some of the input nodes receive proper input. In neural networks, noise tends to be filtered out through the use of sigmoid functions, thresholds and weights. Neural networks benefit from high execution speeds and can easily be implemented on parallel architectures due to their distributed architecture [2]. Unsupervised neural networks, such as Kohonen networks and Radial Basis Function networks have the capability to classify data without supervision [2]. Neural networks generally excel at low level signal processing tasks [9].
2.5.3 Disadvantages

Unfortunately, neural networks do not handle high level sequential tasks well [9]. Since the knowledge within a neural network is stored as numeric weights between nodes, the rules and reasoning process in neural networks is not easily interpretable or explainable [2], [9]. Also detracting from the widespread use of neural networks is the large quantities of input data needed to train the systems [1].

2.5.4 Conclusions

Recently, neural networks have been heavily researched and many architectures and training techniques have been proposed [2]. Neural networks have the potential to provide more human characteristics of problem solving that are difficult to capture in logical, analytical techniques. Overall neural networks excel in situations where mathematical relationships between input and output are unknown or speed is required. Neural network drawbacks such as the lack of explanation facilities and large training data sets prevent neural networks from use in situations where reasons for output are needed or only small amounts of training data are available. Critical systems such as medical or command decision problems fall into this category and knowledge-based or other techniques are more suitable [9].

2.6 Genetic Algorithms

2.6.1 Overview

Genetic algorithms are based on concepts of biological evolution, such as survival of the fittest [2]. In these numerical systems, computational equivalents of biological systems self-organize and adapt to their environment [6]. From an engineering
standpoint, genetic algorithms can be viewed as a robust search procedure for complex spaces [2].

The main components of genetic algorithms are a solution set, a fitness function, and an algorithm for generating new solutions from previous solutions [1]. The solution set is a set containing finite length strings of characters or numbers. Each string represents a solution encoding values for parameters of the problem [2]. The fitness function takes a possible solution and generates a ranking of the viability of the solution relative to the other solutions. The generator function incorporates algorithmic and random factors to create new solutions. Three main generator functions involve the computational equivalents of reproduction, crossover, and mutation [2]. Reproduction models the biological process of natural selection, whereby the most fit individuals reproduce. Crossover represents the biological process of mating, while mutation is the analog of biological gene mutation.

Genetic algorithms begin by calculating the fitness values of each solution in the solution set using the fitness function. The least fit solutions are dropped from the solution set, and the generation function is applied to solution pairs to create new solutions. These new solutions are joined with the solution set to form a new generation of solutions. The fitness function is applied again and the cycle repeats until a suitable solution is found. This process is summarized in Figure 7.
Figure 7. Basic algorithm for genetic algorithms [2]

Initialize population (arbitrarily or otherwise)
Calculate fitness values of population members
LOOP While (Termination Criteria is Not Reached)
   Apply selection criteria
   Generate offspring
   Apply crossover operator
   and/or Apply mutation operator
   Generation = Generation + 1
   Calculate fitness values of population members
END LOOP

2.6.2 Advantages

Genetic algorithms can be considered an alternative search technique to traditional artificial intelligence searches [1]. Genetic algorithms work well with solving non-convex optimization problems that are generally intractable [2]. For complex problems with prohibitively large search spaces, genetic algorithms excel where most search algorithms fail [1].

Since each gene sequence represents a possible solution, many points of the search space are considered simultaneously, reducing the chances of converging to local minima [6]. Additionally, probabilistic methods are used in addition to deterministic ones, lessening the chances even more.

2.6.3 Disadvantages

Drawbacks to using genetic algorithms include the format required for the problems genetic algorithms solve. For genetic algorithms to be used, it must be possible for the problem to be formulated as a series of characters or numbers [1]. Also, a fitness function must exist to rate solutions against one another, along with a generator function to form new solutions in each generation. One final drawback is the variable execution
time. When random factors are involved, it becomes almost impossible to predict when
the genetic algorithm will return a reasonable result, if ever.

2.6.4 Conclusions

In general, genetic algorithms are good for scheduling and resource allocation
problems. However, they have little capability to function as logical or analytical
artificial intelligent systems. Genetic algorithms are best used in situations where there is
a large problem space to examine, only an optimal solution is required, and execution
time is not a relevant factor. Job scheduling, optimization problems, and performance
enhancement are some of the applications genetic algorithms have been applied [6].

2.7 Comparison of basic artificial techniques

The different artificial intelligence techniques outlined above vary greatly in their
problem solving properties. Below is a table outlining the general capabilities of these
intelligent techniques (see Table 2). The following chapter will examine methods of
combining these basic techniques in hybrid systems.
Table 2. Property assessment of intelligent techniques

<table>
<thead>
<tr>
<th>Technologies</th>
<th>Properties</th>
<th>Automated Knowledge Acquisition</th>
<th>Adaptation to new problems</th>
<th>High-level reasoning</th>
<th>Low-level reasoning</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expert Systems</td>
<td></td>
<td>5</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>Case-based reasoning</td>
<td></td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>Fuzzy Systems</td>
<td></td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Neural Networks</td>
<td></td>
<td>1</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>Genetic Algorithms</td>
<td></td>
<td>2</td>
<td>5</td>
<td>4</td>
<td>5</td>
<td>4</td>
</tr>
</tbody>
</table>

Key: Rated from 1-5, 1 most capable, 5 least capable
3.0 Hybrid AI techniques

3.1 Introduction to hybrid systems

In one form or another, artificial intelligence has been around for over four decades [1]. Over this time, a large assortment of artificial intelligence techniques have been developed, the most prominent being expert systems, neural networks, case-based reasoning, fuzzy logic, and genetic algorithms. Each of these techniques has been or is currently being heavily researched to explore their capabilities [10]. However, obvious limitations to each technique exist, as shown in the previous section of this thesis.

Artificial intelligence researchers have examined this problem and many have come to the conclusion that none of these techniques fully model the intelligent process of the human mind [9]. Furthermore, they conclude that through the combination of techniques and models of thought, significant advancement can be made in the area of intelligent systems. This new breed of intelligent systems has become known as hybrid artificial intelligent systems, and research in the past ten years has yielded much knowledge about the best ways to combine these systems, their capabilities, and their limitations [1].

The following section first examines the basic methods of combining artificial intelligence techniques into hybrid systems. Then, the different techniques are compared to determine their relative problem solving capabilities. These results are summarized in a table at the end of the section.
3.2 Methods of combining hybrid systems

Four main methods of combining hybrid systems have been found: stand-alone, transformational, semi-integrated, and fully integrated [1]. The following sections discuss each of these strategies, providing basic concepts and descriptions, variations and suitable uses of the model, and benefits and limitations.

3.2.1 Stand-alone

The stand-alone model is a degenerate form of integrating hybrid systems [1]. However examining it provides a foundation with which to compare the other models. In the stand-alone hybrid system model, each artificial intelligence component works independently of the components. Each component processes the input separately and produces its own output. There is no interaction between components in any way, except that outputs may be compared.

Benefits of using this model include redundancy in processing. Results can be compared between alternative computational methods, which can verify or invalidate the findings of one another. Similarly, the results can be used to test the problem solving capabilities of each technique. Statistics such as computational duration, accuracy, and robustness can be measured to determine which technique is most suitable for a given problem-solving task.

In terms of hybrid system development, stand-alone models are the most simple to develop. Commercial packages can be used to create each component, with minimal integration to synchronize the input and collect the output of both systems.
Unfortunately, the strengths of one technique can not easily support the other technique. Maintenance is doubled since each system must be independently updated. Additionally, these updates must be simultaneous to avoid confusion, and updates to one system can not help the other.

3.2.2 Transformational

The transformational model takes the stand-alone model one step further in integration [1]. Both have individual components that operate independently. However, instead of having separate process streams for the individual components, the output of one component becomes the input for the other. An example of such a system is an expert system that feeds results to a neural network.

One benefit of such a model is the speed of development. Not much more effort is required to develop a transformational model than from a stand-alone model. Maintenance is now limited to one system, and changes to one component can improve processing in the other component.

Unfortunately the tighter integration adds disadvantages to the system. Significant modifications to the system can require a complete rewrite. Fundamentally, the model is limited to the processing of the target technique. In the example above, the system is still a neural network. The expert system merely provides a pre-processing interface to the neural system.
3.2.3 Semi-Integrated

Semi-integrated hybrid systems differ from transformational models in that they share data between the components. This sharing is usually done through shared files or memory. Instead of a linear process whereby one component does computations and passes the data to the next component, this structure allows processes to alternate or even run simultaneously.

One important type of semi-integrated system is the blackboard model [1]. Here, two or more intelligent agents use a shared data structure, called the blackboard, and operate independently on the data. Typically these agents have been logical or analytical in technique, but there is potential to add neural and other non-analytical techniques to this type of system. The blackboard model has been used in complex pattern recognition, fault isolation and repair, and advanced decision support.

Another type of semi-integrated system is embedded systems [1]. Here, one technique is embedded within the intelligent process of another. For example, a genetic algorithm may be embedded into the modification engine of a case-based reasoning system. This allows the case-based reasoning to operate as usual, but when a solution must be modified to suit a new problem, the genetic algorithm can be used to generate a new solution from previous examples.

Semi-integrated systems hold several advantages over less integrated systems. First, there are more advanced interactions between components, allowing the strengths
of each technique to complement one another. This allows for better leveraging of capabilities and specialization.

However, there are drawbacks to using semi-integrated systems. Development and maintenance complexity increases compared to less integrated systems. Verification and validation of results from such a system become much more difficult since errors are not as easily tracked to their sources.

3.2.4 Fully integrated

The fully integrated model of hybrid system development involves combining two or more intelligent techniques so tightly that in essence they become one technique. One prime example of a fully integrated hybrid system is the expert network.

Figure 8. Expert network

In an expert network (see Figure 8), nodes of a neural network correspond to symbols in an expert system. The weights of the neural network correspond to
relationships between these symbols, such as AND, OR, NOT, GREATER THAN, EQUAL TO, etc. When operating on data, the system operates much like an expert system, reasoning using an inference engine technique. However, the neural network supplies much of the intermediate computation as nodes are activated through propagation of facts. For training, the system behaves like a neural network; it modifies relationships and associations between symbols using neural network training techniques. Also, an operator may choose to rearrange or rewrite rules manually.

Another alternative design is to only use the input and output nodes of a neural network as symbols within an expert system. Intermediate nodes do not affect the computation of the expert system except by propagating signals from the input nodes to output nodes. This is especially useful when the relationships between symbols is complex or undetermined.

Fully integrated systems provide several major benefits. First, performance improves since the individual components do not need to transfer data or transform data from one structure to another. Fully integrated systems offer the capability to solve problems previous unsolvable by using either technique alone or in series. By fully integrating techniques, one technique can lend capabilities to another, such as expert systems providing explanation capabilities to a neural network. Fully integrated systems can adapt in several ways, such as by training or manual manipulation in the expert network example. Furthermore, generalization capabilities, noise tolerance, and logical deduction can all be combined into one system.
Despite these numerous advantages, there are prominent disadvantages to fully integrated systems. Complexity increases as more components are integrated into the system. Design of such fully integrated systems is highly difficult, requiring an in-depth understanding of the problem. There is only one complex system in which to trace an error, which could make debugging difficult. However, verification, validation and maintenance most likely will be much more difficult than in less integrated systems, depending on how well the system was designed.

3.3 Future of research and development

Clearly, each of these hybrid system development models has distinct advantages and disadvantages. Generally as integration becomes tighter, complexity and cost increase, along with capabilities and performance. When choosing a hybrid system design, issues such as development time, maintenance, performance, and cost must be adjusted to suit the task at hand.

Survey of hybrid systems

3.4 Expert Systems and Neural Network Hybrids

3.4.1 Complementary characteristics

Expert systems and neural network have many complementary characteristics that make them well suited for combination in a hybrid system. Expert systems have the ability to reason logically about a problem, incorporate expert knowledge in the form of rules, are easily modified to add new rules, and can explain their answers. Unfortunately,
knowledge acquisition and the growth of the rule base make using expert systems difficult in certain situations.

Neural networks, on the other hand, can discover relationships between input and output that are not easily described by experts, can be trained using historical data without the aid of expert knowledge, can be implemented easily on parallel architectures, and are resistant to noisy or incomplete data. But, neural networks lack an explanation facility, and need a large training set before becoming useful. By combining neural networks and expert systems into a hybrid system, their combined strengths can make up for the weaknesses of each.

3.4.2 Methods of integration

3.4.2.1 Semi-integrated

Most hybrid neural and expert systems use the semi-integrated model. In these systems, the neural networks and expert systems share data and simultaneously process information to produce a solution. One particularly relevant example of this technique is a multiple target recognition system developed by Charles River Analytics, Inc. [17]. The system examines the spatio-temporal attributes of target trajectories and classifies multi-sensor data [1]. The system uses conventional signal processing and probabilistic tracking algorithms along with neural networks and knowledge-based modules.
As shown in Figure 9, the system is organized hierarchically, with an executive expert system module performing the overall decision making, management, and coordination functions. Below this module, neural network classifiers and knowledge based classifiers use analytic algorithm modules to generate possible classifications for the upper expert system module. At the lowest level, target and ownship models create training data to test the system.

In their study, Caglayan and Gonsalves developed two hybrid systems, with different degrees of interaction within the algorithmic components and modeling. The results of their study show that the performance of the hybrid system is more accurate than for systems that only use one of the technologies [17].

3.4.2.2 Fully integrated

Fully integrated hybrid expert and neural network systems generally take the form of an expert network. Here the nodes of the neural network are the symbols used in reasoning by the expert system. Weights between the nodes of the network correspond to
the relationships between symbols in the expert system. An example of such a system is a connectionist expert medical diagnosis system [26].

Figure 10. Connectionist expert system for medical diagnosis (modified) [1]

In Figure 10, the nodes of the neural network represent specific facts or aspects of the knowledge domain. Input nodes represent different symptoms, with input values of -1, +1, and 0 indicating that the symptom was absent, present, or not checked respectively. The system uses training data, which consist of symptoms with known diagnoses, to set the appropriate weights between nodes to yield the desired performance. Additional nodes are used to prescribe and aggregate treatments for the diagnosed diseases.

In this model the knowledge base of experts is effectively translated into weights within the system. The inference engine further interprets results, and directs questions from the user to minimize input. By incorporating an expert system, the model can
provide explanations of the results. By using a neural network, the system can be retrained through files of data without knowing or rewriting the rules.

3.4.3 Advantages

As shown in these two examples, combining expert systems and neural networks creates systems that exhibit the benefits of each technique while minimizing the drawbacks of either individual system. These hybrid systems have explanation capabilities, can learn from both expert input and historical data, reason logically, can generate relationships that are difficult or impossible to derive manually, and show some tolerance to noisy or incomplete data.

3.4.4 Disadvantages

Unfortunately, designing an expert network is quite difficult. This is due to the need to decide which symbols to use and which connections to make. Rule bases can be rather sparse and thus lead to thin neural networks, which might not respond well to training. Also, the intricacy of the expert network makes adding new rules quite difficult.

3.4.5 Conclusions

Overall, expert systems and neural networks have many complementary characteristics, making them ideal candidates for integration in hybrid systems. By putting together the logical, analytical method of expert systems with the connectionist, associative technique of neural networks, systems with logical and associative capabilities can be developed. These systems generally can explain results, learn from both user modifications and training from historical data, and exhibit resilience to noisy or incomplete data.
3.5 Expert Systems and Fuzzy Logic Hybrids

3.5.1 Complementary characteristics

Fuzzy logic and expert systems share many characteristics that make hybrid system development relatively simple. Both are suitable for decision making and other knowledge oriented problems. Both use a logical, analytical engine to develop solutions. Systems developed with these technologies exhibit development efficiency and consistent performance.

3.5.2 Methods of Integration

3.5.2.1 Fuzzy expert systems

Although there are several ways to integrate these two techniques, fully integrated fuzzy expert systems are the dominant form of integration. A fuzzy expert system is an expert system with fuzzy logic embedded in the rules. For example, one rule in such a system could be “if engine temperature is too hot, decrease throttle”. In Figure 11 the architecture for a adaptive fuzzy expert system developed by Cox [27] is shown. The system is used to control the speed of an automobile on a highway.

Figure 11. Adaptive fuzzy expert system [1]
The system maintains a required speed by feeding back a tachometer reading to a throttle controller. The controller also monitors load torque and reacts, such as to changes in road grade. The adaptive quality of the system allows it to modify its rules, fuzzy sets, and defuzzification method depending on the performance of the controller. In the diagram, the fuzzy expert system includes the function of fuzzification of the input data for the speed and change in grade.

Figure 12. Membership functions for degrees of speed [1]

To use torque and speed data in the fuzzy rules, membership functions had to be defined for all the input and output variables. The membership functions for degrees of speed are shown in Figure 12. Labels are assigned to each range of values for each variable and the system designer initializes the membership functions. During operation, the shapes and positions of these membership functions are subject to change. Similar membership functions are defined for the load and for the output throttle movement. In Table 3, a fuzzy associative map defines the relationship between the fuzzy inputs and outputs.

49
Table 3. Fuzzy associative map for inputs (speed, load) and output (throttle movement)

<table>
<thead>
<tr>
<th>Load torque</th>
<th>Tachometer</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Very slow</td>
</tr>
<tr>
<td>Zero</td>
<td>Large positive</td>
</tr>
<tr>
<td>Small positive</td>
<td>Large positive</td>
</tr>
<tr>
<td>Moderate positive</td>
<td>Large positive</td>
</tr>
<tr>
<td>Large positive</td>
<td>Large positive</td>
</tr>
</tbody>
</table>

The system operates as follows. Specific values of the tachometer and load torque are fuzzified before use in fuzzy rules. For each input, appropriate linguistic variables are assigned according to fuzzification functions. Degree of membership within each membership set are determined using the membership functions. Since a value may be a member of one or more sets, one or more rules may be activated, yielding multiple outputs. For example, a tachometer reading might be assigned to (very slow, 0.15) and (slow, 0.40). With a load torque of (moderate positive, 1.0), two conclusions would be drawn: (large positive, 0.15) and (small positive, 0.40) throttle movements. The defuzzification sequence would combine these two results into a single crisp throttle action.

3.5.3 Advantages

Fuzzy logic and expert systems are combine relatively easily since fuzzy concepts are naturally expressed in the form of rules. Fuzzy expert systems provide higher levels of abstraction for system designers to interact with experts while retaining the reasoning capabilities of traditional expert systems. Membership functions allow parameters to be
adjusted to tune the system and reduce the amount of detail needed in knowledge acquisition. Adaptive fuzzy expert systems automate this learning process and can improve performance over time.

3.5.4 Disadvantages

Although combining expert systems and fuzzy logic together gains several benefits, problems remain. Complex fuzzy systems can become difficult to design. For these problems, membership functions become hard to define and may take tedious trial and error. As parameters to the system grow, the matrix representing relationships becomes unwieldy and unmanageable.

3.5.5 Conclusions

Adding fuzzy logic to expert systems adds numerous benefits. Precise values for rules within the system can be abstracted, speeding up development time and simplifying maintenance. Since all reasoning is done on the level of linguistic variables, the system can easily adapt to new situations by modifying the membership functions or fuzzy rules. Also due to the use of fuzzy linguistic variables is the ability to deal with imprecise inputs. Fuzzy expert systems effectively combine the beneficial traits of both techniques, however, the added complexity of design must be taken into consideration. For most problems with precise relationships between inputs and outputs, expert systems would suffice. In situations where inputs are imprecise, the environmental factors may force adjustments, or the acquisition of expert knowledge would be tedious, fuzzy expert systems can provide an effective alternative to the traditional techniques.
3.6 Expert Systems and Genetic Algorithm Hybrids

3.6.1 Complementary Characteristics

Expert systems and genetic algorithms come from distinct categories of artificial intelligence. Expert systems function using logical, analytical methods. Genetic algorithms excel at search problems that require optimal solutions. Relatively little research has been done in the area of combining expert systems with genetic algorithms. Generally, the work that has been done have used a semi-integrated technique with embedded components.

3.6.2 Methods of Integration

3.6.2.1 Embedded systems

Expert systems embedded in genetic algorithms are used to limit the search space for the genetic algorithm. These expert systems incorporate heuristics that guide the genetic algorithm when selecting candidates for recombination and mutation. Without this guidance, genetic algorithms must search a much larger set of solutions, which may be excessively slow or never complete.

3.6.3 Conclusions

The use of genetic algorithms to assist expert systems is a relatively unexplored area of hybrid systems. Some implementations could include using genetic algorithms to develop rules from data, speeding up the knowledge acquisition process. Another possibility is to use genetic algorithms to fine tune rules. Both of these areas have not been researched enough to be practical in engineering purposes.
3.7 Neural Networks and Fuzzy Logic Hybrids

3.7.1 Complementary Characteristics

Fuzzy logic offers important advantages in a variety of application areas. Fuzzy systems incorporate the high level reasoning capability of expert systems, but also allow for the use of uncertainty and imprecision common in real world problems. Explanation capabilities are featured which permit understanding and justification of results. When precision is not possible or not needed, fuzzy rule-based systems allow for approximate reasoning and estimation.

Neural networks provide similar problem solving capabilities. Systems using neural techniques can learn from historical data and adapt to the environment as new data arrives. Through either supervised training or self-organization, neural networks can categorize data and recognize complex patterns. In terms of robustness, system performance degrades slowly as nodes fail, and noisy or incomplete data can often be tolerated.
Table 4. Characteristics of Fuzzy and Neural Systems [1]

<table>
<thead>
<tr>
<th>Properties</th>
<th>Fuzzy Systems</th>
<th>Neural Networks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function estimator</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Trainable, dynamic</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Improvement with use</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Parallel implementations</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Numerical</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Tolerance for imprecision</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Explicit knowledge representation</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Adaptive</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Optimizing</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Interpolative</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Tolerance for noise</td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>

As seen in the Table 4, neural networks and fuzzy systems share several characteristics. Both are model-free function estimators that can be adjusted or trained for improved performance. They are both easily implemented on parallel architectures due to their distributed natures. Neural networks consist of nodes linked by weighted connections, while fuzzy systems process rules that associate, in parallel, fuzzy output sets with fuzzy inputs.

3.7.2 Methods of integration

The high degree of symmetry between the two techniques allows for a natural combination of the systems into a fully integrated fuzzy expert neural network. As an alternative, fuzzy and neural components may be combined in a semi-integrated system.
3.7.2.1 Fully integrated fuzzy expert network

A fuzzy expert network follows the model set forth by expert networks. However the expert system is replaced by a fuzzy logic system. The additional benefit of this technique over using an expert network is the ability to deal with imprecise inputs and reasoning.

3.7.2.2 Fully integrated fuzzy neural network

Figure 13. Fuzzy neuron [1]

A fuzzy neural network is a neural network modified to incorporate fuzzy computations within its nodes [1]. The fundamental building block of these systems is the fuzzy neuron. In networks that map fuzzy input to crisp output, modified neurons make up every layer of the network, as shown in Figure 13. Input values and weights both have fuzzy values determined by membership functions, and a modified summation process takes the place of the sigmoid function of most neural networks. The system then uses a centroid function on the result to obtain a crisp value for the output of the node.

3.7.2.3 Semi-integrated fuzzy logic neural network hybrid
Fuzzy logic and neural networks can also be used as components within a semi-integrated system. Here one or modules of fuzzy logic and neural networks are run independently on input data. Often a fuzzy logic component processes data before sending it to a neural network, or a fuzzy logic unit processes the results of a neural network [1].

3.7.3 Conclusions

Fuzzy logic and neural networks share many characteristics as shown in Table 4. This makes it natural to combine the systems, yet the overlaps make some of the characteristics redundant. Both are function estimators, are trainable, can be implemented in parallel architectures, and deal with numerical data. The high degree of overlap in their capabilities brings up the question of whether it is worth the effort to combine two similar systems. In the opinion of the author, the fuzzy expert networks are the most useful of these hybrids since it combines explanatory and reasoning capabilities with the ability to train and associate data. The drawbacks of added complexity and maintenance should decline as more research is invested in this area, and the capabilities of this system offer the most promise in terms of representing the though processes of humans.

3.8 Neural Networks and Genetic Algorithm Hybrids

3.8.1 Complementary Characteristics

Neural networks and genetic algorithm hybrids take advantage of the associative problem solving characteristics of neural networks and the efficient search capabilities of genetic algorithms.
3.8.2 Methods of Integration

3.8.2.1 Semi-integrated embedded systems

The majority of neural network and genetic algorithm hybrids have been semi-integrated embedded systems, where genetic algorithm components are used to improve the performance of neural networks [1]. Genetic algorithms have been applied as pre-processors to prepare data for input to neural networks, choosing the best way to represent data or selecting the most appropriate data sets from a domain. Genetic algorithms have also been used to reduce the trial-and-error process of finding good values of the weights and parameters.

3.8.2.2 Transformational hybrids

Transformational hybrids include the use of genetic algorithms to find optimal initial values that speed up the process of training a network. Research has been also directed towards using genetic algorithms to evolve neural network designs, however this area has not been deeply explored.

3.8.3 Conclusions

Overall, neural genetic hybrid systems tend to either embed genetic algorithms into neural networks to increase the efficiency of tasks such as preparing data for input or choosing proper weights for connections. In transformational models genetic algorithms are not involved in the actual computation of input, but instead aid in the development of a neural network to suit the problem.
3.9 Expert System and Case-based Reasoning Hybrids

3.9.1 Complementary Characteristics

Expert systems and case-based reasoning both rely on logical, analytical techniques. Whereas expert systems gather input and draw conclusions, case-based reasoning gathers input and produces an answer from a historical database of solutions. Since both systems have similar applicability in terms of problem domains, they are natural alternatives in independent stand-alone systems [1].

3.9.2 Methods of Integration

3.9.2.1 Stand-alone hybrids

In stand-alone hybrids, multiple expert system and case-based reasoning components examine the same data and draw conclusions or recall previous solutions respectively. The expert system allows the overall system to incorporate expert knowledge about the problem, while the case-based reasoning gives the system the capability to learn from experience. By running both types of processes in parallel, results can be compared, allowing for increased confidence in solutions. In addition, where large amounts of training data are not available, the expert system components can provide reasonable solutions while the case-based reasoning components gather experience.

3.9.2.2 Embedded hybrids

Expert systems can also be embedded within a case-based reasoning framework, which represent the majority of semi-integrated expert system/case-based reasoning
hybrids [1]. Expert systems enhance the performance of the case-based reasoning system by providing heuristics for case retrieval, and by adapting existing cases to match more closely the current problem.

3.9.3 Conclusions

Expert systems and case-based reasoning overlap in several major characteristics. Both use logical reasoning to find solutions to problems. Both can incorporate expert knowledge – expert systems in the form of rules, case-based reasoning in the form of solutions. Both can explain their output, although expert systems do so more explicitly. Despite the redundancies in capabilities, expert systems and case-based reasoning can effectively be combined to create a system with marked improvements in performance over either technique alone. Stand-alone hybrids of this type combine explicit explanation with the ability to learn and redundancy in computation. Embedded hybrids improve the performance of case-based reasoning and allow explanations of why a case was retrieved and how a solution was modified to fit the problem.

3.10 Neural Networks and Case-based Reasoning Hybrids

3.10.1 Complementary Characteristics

Neural networks and case-based reasoning both require large amounts of training examples that are stored and later recalled to solve future problems. As a result the two technologies represent alternative approaches and hybrid systems that incorporate both tend to be stand-alone models [1].
3.10.2 Methods of Integration

3.10.2.1 Stand-alone hybrids

In stand-alone systems, neural networks and case-based reasoning provide redundancy of computation that verifies results or fortifies decision-making. However one drawback to these hybrids is the large amount of training examples needed to develop the system. Neither component aids in reducing this problem.

3.10.2.2 Embedded hybrids

Case-based reasoning frameworks can also be improved by embedding neural components. Neural networks can be used to analyze the case data and establish a good set of seed cases to start the case-based reasoning operations. Neural networks have also been used to speed up case retrieval [1].

3.10.3 Conclusions

Neural networks and case-based reasoning systems represent connectionist and symbolic approaches to problem solving. Combining the two allows system engineers to leverage both problem-solving techniques in one solution. However this synergy is hampered severely by fact that both techniques require large amounts of training data before being functional.

3.11 Fuzzy Logic and Case-based Reasoning Hybrids

3.11.1 Conclusions

Relatively little research has been done in the area of combining case-based reasoning and fuzzy logic [1]. However, fuzzy logic systems are similar to expert
systems in methodology, making much of the discussion on expert systems and case-based reasoning relevant. Embedding fuzzy logic as a pre-processor for input could make interfaces user-friendlier. Additionally, fuzzy techniques could be applied to case descriptions and case retrieval, speeding up recovery time and making the system capable of interpreting and reasoning with imprecise input. Also, fuzzy systems can be used as a post-processor to evaluate solutions retrieved by the case-based reasoning mechanism. As more research is conducted in fuzzy logic and case-based reasoning hybrids, the systems should become more useful and abundant.

3.12 Genetic Algorithms and Case-based Reasoning Hybrids

3.12.1 Conclusions

Research in combining the relatively recent techniques of genetic algorithms and case-based reasoning are scarce, however a few models do exist [1]. One hybrid involves embedding genetic algorithms in case-based reasoning systems to aid the adaptation process. When multiple solutions are suggested, a genetic algorithm can be applied to produce a hybrid solution. Unfortunately, a fitness function for the generated solutions may not exist and trial and error may have to be used to test each proposed solution.

3.13 Comparison of hybrid system strengths/weaknesses

This chapter has provided a brief overview of current hybrid artificial intelligence techniques. A comparison of key characteristics of these various techniques is listed in Table 5. Overall, it can be observed that the best combinations of techniques take pieces from both symbolic and the connectionist/numerical ends of the information processing
spectrum. Such systems exhibit the capability to perform high level reasoning and low level pattern matching and/or optimization, along with a host of other useful traits.

Table 5. Comparison of hybrid artificial intelligent systems

<table>
<thead>
<tr>
<th></th>
<th>High Level Reasoning</th>
<th>Low Level Reasoning</th>
<th>Learning</th>
<th>Explanation</th>
<th>Parallel</th>
<th>Ease of maintenance</th>
<th>Well researched</th>
</tr>
</thead>
<tbody>
<tr>
<td>ES/NN</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>ES/FL</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>ES/GA</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NN/FL</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>NN/GA</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CBR/ES</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>CBR/NN</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CBR/FL</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>CBR/GA</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Key:  
X – indicates capability  
ES – Expert System  
CBR – Case-based Reasoning  
FL – Fuzzy Logic  
NN – Neural Network  
GA – Genetic Algorithms
3.14 Implications for artificial intelligence design – a Hybrid Design Process

Combining artificial intelligence techniques creates systems that are capable of solving problems either technique alone could not accomplish. However, it is apparent that the resulting hybrid system exhibits characteristics originally found in the individual techniques. For example, expert networks exhibit the high level reasoning of expert systems along with the adaptability of neural networks. But individually, the expert system can perform high level reasoning and individually, the neural network can adapt to new data. In combining hybrid systems, often no new characteristics are formed; the whole is not greater than the sum.

To an artificial intelligence engineer, hybrid systems can potentially present a daunting array of choices. There exist almost an infinite array of hybrid systems to choose from if you consider combinations of three or more techniques and the range of integration methods. However, the observation of minimal new functionality added by hybrid systems suggests a Hybrid Design Process.

First, the primary characteristics of the problem should be determined, such as the levels of reasoning involved (high-level, low-level, or both). These characteristics will suggest the use of certain artificial intelligence techniques. High-level reasoning for example suggests the use of symbolic processing. Low-level reasoning suggests the use of connectionist or numerical techniques. Once the appropriate techniques have been chosen, comparisons should be made to determine which techniques are most suitable to the problem.

At this point, it may be determined that a choice is appropriate for the problem, however its capabilities do not satisfy all the constraints imposed. This is when hybrid
techniques should be examined, since the addition of another technique may complement
the original. The additional technique should undergo the same scrutiny applied to the
original technique; similar techniques should be compared to determine the optimal
choice.

Finally, if a hybrid technique is required to satisfy the constraints of the problem,
the level of integration should be determined. Tighter integration produces better
interaction between components, but comes at the cost of higher levels of complexity and
costlier maintenance. It is the opinion of the author that minimal integration should be
applied to accomplish the hybridization.

This Hybrid Design Process should yield a system that fulfills the requirements of
the problem without adding undue complexity. The following chapters detail a case
study of a complex information system and the application of this Hybrid Design Process.
Part II – Application

4.0 Case Study Introduction

4.1 Company background

The MITRE Corporation is a not-for-profit corporation that was formed to deal with critical national issues such as defense, commercial aviation, and taxation. The majority of work done at MITRE is concentrated in three Federally Funded Research Development Centers (FFRDC) which service the Department of Defense (DOD), the Federal Aviation Administration, and the Internal Revenue Service and Treasury Department.

Unlike most corporations, FFRDC are prohibited from manufacturing products or competing with industry. Instead, they work together with commercial corporations from planning and concept design to technology insertion and integration. As agents for the government, FFRDC specialize in scientific research and analysis, systems development, and systems acquisition.

The FFRDC of MITRE working with the DOD is currently focusing on developing more thorough cooperation among command and control systems, with the ultimate goal of a single, integrated command and control system that can support joint and coalition operations worldwide. The research within this thesis is directly in line with this goal, as it examines the use of a hybrid system to streamline the command and control system involved in detecting and identifying mobile missile launchers.
4.2 Case study process and roadmap

This case study presents the results and the problems overcome by a hybrid artificial intelligent system that was designed for use in a military system under development at the MITRE Corporation. For several months, the author worked alongside MITRE engineers to examine the system including input, workflow, and output. This information was used to design a hybrid intelligence system that could enhance the operation of the aforementioned system by automating parts of the workflow. In the course of the research and design, it became apparent that a key bottleneck in the system was human-related. The first part of this study establishes the applicability of artificial intelligence techniques to military systems. The second part involves identifying the relevant issues in this specific military system. Finally, the third segment proposes a hybrid intelligence solution that is compatible with the workflow of the system. These steps are presented in detail in the following subsections of this case study. The first section summarizes the applicability of artificial intelligence techniques in military systems. Following it is a section that presents the system under development and the main areas for possible improvement. The next section reviews numerical and artificial techniques used previously to address these areas of improvement. Finally, the last section presents the hybrid artificial intelligence solution and justifications for the design.
5.0 Applicability of Artificial Intelligence to Military Systems

5.1 Previous applications

Artificial intelligence techniques have been used in a variety of military systems. The following is a brief listing of military systems that employ or have employed artificial intelligence techniques.

5.1.1 Expert Systems

One example military system, which uses expert system techniques, is the Interactive Tactical Management System, ITEMS [3]. ITEMS provided an environment where simulated companies, battalions, and battalion level commanders interacted using a complex hierarchy of rules. The rules are developed based on military doctrine and encode behaviors such as fire positioning, evasive maneuvering, withdrawing, and halting. ITEMS proved that expert systems could be used to control computer generated forces [2].

Expert systems have also been used in the Close Combat Tactical Trainer, CCTT [4]. This was another military simulation that modeled the decision making from the lowest vehicle level behaviors through battalion. The system focused on tactical decision making at the platoon level and illustrated that knowledge bases could be used to make tactical decisions and choose between competing behaviors from different sources [2].
5.1.2 Neural Networks

Military systems that have used artificial neural networks range from automatic target recognition to sensor fusion activities. At the National Test Facility (NTF), an artificial neural network was implemented to rapidly assign weapons to targets in a large simulation [5]. The system controlled a variety of simulated interceptors to remove both Inter-Continental Ballistic Missile and strategic ballistic missile targets from the sky. From this study, it was shown that neural networks could out perform many algorithmic statistical optimization techniques [2].

A neural network was also used to develop a system to build maintain and test large knowledge bases at TRADOC Analysis Command [2]. The work focused on “automated development of knowledge bases” [2]. In situations that were familiar to the system, expert system rules were applied. However in unfamiliar situations, the neural network would try to improve the rule-base. This work suggests that neural networks can be applied to allow knowledge-based systems to adapt to changing situations [2].

5.1.3 Fuzzy Logic

The fundamental nature of fuzzy logic (reasoning with imprecise data), have made them ideal candidates for use in military command systems. One such example is a fuzzy logic decision support system to determine mission effectiveness for special operations forces [2]. The project examined the launch phase of Navy SEAL missions and analyzed the effect of various deployment options on mission effectiveness. The Simulator Systems Research Unit at the Army Research Institute also worked on a
project using fuzzy logic [2]. The system uses fuzzy logic command entity to deploy and control communications equipment.

5.1.4 Genetic Algorithms and Evolutionary Programming

Genetic algorithms have been used in path planning activities such as the vehicle route planning research done by Pellazar [6]. Another use of genetic algorithms is tactical behavior research. Adamson and Joshi [7] use genetic algorithms to optimize the number of friendly forces killed and the time taken to reach an objective. The system takes into account behavioral factors such as changes in speed, sensors, and types of fire and generates a sequence of activities. This work showed significant success in the use of genetic algorithms to develop tactical behavior [2].

5.1.5 Case-based Reasoning

One notable example of the use of case-based reasoning in a military system is research done by Kiersey et al. [7]. The system uses a case-based reasoning component to control low-level behaviors for simulated air combat units. Another system which implemented case-based technology was the RAND Integrated Simulation Environment (RISE) [8]. RISE used a case-based planning scheme in which state information and world-view were used as indices to a database of plans. The plans incorporated a series of steps and contingencies in case of plan failures.

5.2 Future military use of artificial intelligence

The previous examples show that artificial intelligence is prevalent in many military research endeavors. Artificial intelligence techniques have been used in military simulations, optimization tasks, planning tasks, and decision support. These instances
suggest possible uses of artificial intelligence in military applications and validate the applicability of using artificial intelligence in military systems.

In the following chapters, a brief introduction is given to the problem of targeting time-critical targets (TCT).
6.0 Problem background

6.1 Joint Targeting

In modern warfare, multiple branches of the armed forces must work together to accomplish military objectives. To coordinate forces, the Department of Defense designates Joint Force Commanders (JFC) to control each branch of the military at large-scale confrontations. Together, the JFC decide, at a high-level, the priority of targets. For example, weapons of mass destruction may be assigned high priority while a small tank division may be assigned lower priority.

At lower levels of the command structure, these objectives serve as guidelines for the targeting and destruction of enemy units. Joint Targeting is the process in which targets are selected and matched to appropriate responses based on the objectives of the JFC. The joint targeting process has six basic phases: commander’s objective and guidance, target development, weaponeering assessment, force application, execution planning/force execution, and combat assessment [9]. These phases are shown in Figure 14.
6.2 Time Critical Targets (TCT)

“A surface TCT [time critical target] is a lucrative, fleeting, land, or sea target of such high priority to friendly forces that the JFC or component commander designates it as requiring immediate response” [9]. A target is designated as being time-critical when either a) the target poses or will pose an imminent threat to friendly forces, or b) the target currently represents a highly favorable tactical opportunity.

Examples of surface TCT include mobile rocket launchers, mobile surface-to-air missiles, theater ballistic missiles, mobile weapons of mass destruction, or mobile command and control (C2) vehicles and facilities. Surface TCT may also be non-mobile and include newly discovered operational-level command centers that must be destroyed quickly to permit further friendly force activities. Nuclear or chemical weapons depots may be considered TCT when transportation of the stored weapons is imminent or when
newly discovered. Newly discovered fixed surface-to-surface missile sites are designated as TCT when threatening to launch. Ordinary fixed-surface targets can also be classified as TCT if their timely destruction would provide substantial gains for friendly forces. For example, an airfield may become a TCT if it will soon support enemy aircraft. Or, a road which was used to channel enemy forces could be designated a TCT when its destruction would prevent enemy escape.

Figure 15. Target classification [9]

As seen in Figure 15, targets are classified as either planned or immediate. Planned surface TCT are usually fixed targets that are known to exist in a given area. Their status is generally upgraded to TCT because they present an exceptional opportunity to friendly forces and must be acted upon within a given time frame for maximum effect [9].
Immediate surface TCT comprise of two types: unplanned and unanticipated. Unplanned immediate surface TCT are known to exist in an operational area but have not been scheduled for any fire or attacks. Often these targets are hidden and once discovered, they must be dealt with quickly. Generally, the JFC set aside sufficient forces to deal with all known unplanned immediate surface TCT in a given area.

Unanticipated immediate surface TCT are not known to exist in an operational area and present the greatest challenge to targeting and response by friendly forces [9]. These targets require immediate response with possibly sub-optimal resources. Since these targets are unexpected, response is highly reactive and an efficient response system is required to deal with them effectively and efficiently.

The next chapter will detail the JTE system, which addresses the TCT surface targeting problem.
7.0 JTE system

7.1 Overview

The Joint Target Execution (JTE) system is a project whose development is being supervised by the MITRE Corporation. Since MITRE is an FFRDC, it contracts out the actual development of the system to companies. AlphaTech of Cambridge, Massachusetts and Motorola are the main corporations currently working on components of the system.

The goal of the JTE project is to develop a system to address the targeting of surface TCT. Specifically the system is directed towards destroying Transporter/Erector/Launcher (TEL) vehicles, such as SCUD missile launchers. Engineers at MITRE have determined that these targets pose the greatest challenges to a TCT targeting system due to their high threat, their large numbers, and the relatively small window of opportunity in which to eliminate them. However, the system will have the capability to deal with other TCT targets.

Much of the documented information about the JTE system is classified. As a result, the author has obtained much of the information about the system from interviews with engineers working on the project or through live demos of the system. Where possible, publicly available documents on the internet were used to obtain descriptions of systems, such as JTIDS and JSTARS.
7.2 Basic functionality

The JTE system provides several basic functions. First, it serves as a central location to collect information useful to identify surface TCT. Various sensors will feed into the JTE system, providing such information as radar, imagery, and intelligence data. Second, by analyzing this information, the JTE system identifies surface TCT and nominates them for destruction. Third, the JTE handles the task of assigning weapons platforms to these nominated TCT.

7.3 System details

7.3.1 System structure

JTE is a distributed system, consisting of JTE nodes located at Command and Control (C2) centers throughout the theater of battle (see Figure 16). Each node is responsible for a given rectangular area of the battlefield. These nodes function collaboratively, sharing information feeds and target nominations. These nodes also collectively decide which weapons should be assigned to which targets.
Figure 16. Control Structure. Multiple Command and Control (C2) centers control weaponry in a theater of battle. JTE nodes reside at each C2 center and consist of multiple JTE terminals. JTE nodes share information via the JTIDS/Link16 communications system.
7.3.2 JTE node and terminal

A JTE node consists of one or more UNIX based terminals each manned by a JTE operator. The unit has capabilities of receiving information from relevant real-time and static information sources via UHF signals and other classified means.

The terminal displays a map of the battlefield overlaid with target track data that appear as labeled lines and dots. Menus and dialog boxes allow the JTE operator to interface with the system. Imagery data pop up in separate windows.

7.3.4 Inputs

Inputs to the JTE system consist of real-time data, static data, and requested data. These are described in the sections below.

7.3.4.1 Real-time inputs

Real-time inputs consist of Joint Tactical Information Distribution System (JTIDS) data, Moving Target Indicator (MTI) data, and intelligence updates. Real-time inputs are updated regularly such that the information can be considered up-to-date.

JTIDS is a radio system that supports information distribution, location, and identification capabilities for application to military operations [28]. Information is distributed along this system at a high data rate and encrypted to provide security. JTIDS can interconnect multiple sources and users of information and serves airborne and surface units. The relevant information transmitted through JTIDS to JTE is location and
identification reports for various enemy ground targets. These target locations appear on
the JTE screen as dots.

MTI data is tracking data received via the Joint Surveillance Target Attack Radar System
(JSTARS). Tracking data consists of location and rate of movement for targets. This information
appears on the JTE screen as trails of dots.

Intelligence updates consist of information supplied by various intelligence sources
concerning the location, identification, and/or behavior of targets. Behaviors concerning TELs
include reports of a launch, possible launch sites, or possible “hide” sites where these launchers
wait for use. These appear on the JTE screen as dots, labels, or circles.

### 7.3.4.2 Static inputs

Static inputs to the JTE system consist of historical intelligence data and maps. Data in these input sources are updated periodically, such as daily.

Historical intelligence data contains information such as historical “hide” and
launch sites, along with other information useful for detecting TCT. Similar to the
intelligence updates, this information appears on the screen as dots, lines, or circles.

Maps are provided for the JTE system from the National Imagery & Mapping
Agency (NIMA). These provide the background for the JTE screen, on which tracking
and other data are overlaid. Details such as elevation, roads, and bodies of water are
included in the maps.
7.3.4.3 Requested input

The JTE operator can also request imagery concerning an area of the map or of a specific target. The sources of imagery information are primarily airborne surveillance such as the U2 spy plane, unmanned aerial vehicles (UAV), and the JSTARS aircraft. Satellite imagery can also be used. The maximum power and resolution of each of these imaging sources is classified. However, resolutions as small as one foot have been reported [29].

7.3.5 Workflow Process

7.3.5.1 Epochs

In JTE, TCT targeting occurs in two distinct phases (see Figure 17). These phases are separated into epochs of time, whose duration remains to be specified. Engineers on the project estimate epoch length will be from 90 seconds to 15 minutes.

Figure 17. JTE Epochs
In the first phase, nominations for TCT are collected from each JTE node. In the second phase, available weapons are tabulated at each C2 node. Both phases operate simultaneously and are synched such that an epoch of each phase begin and end at the same time (see Figure 17). At the end of each epoch, available weapons tabulated in the second phase (current epoch) are assigned to targets nominated in the first phase (previous epoch) using an auction algorithm.

### 7.3.5.2 Process

Information sources feed into the JTE terminal and appear on the JTE terminal screen as lines, dots, circles, and labels. The JTE operator examines this data and determines which tracks appear to be TCT targets. At this point, imagery data for the target is often requested to confirm a TCT. Once the operator concludes that a target is a TCT, he or she enters the nomination into the JTE system via a menu box. From here, the automated component takes over.

The automated component of the JTE terminal transmits the nomination to other JTE nodes such that each obtains a complete list of target nominations by the end of each epoch. Using this complete list, each JTE terminal matches weapons under its control to targets in the global target list. At the end of each epoch, the weapons target pairs are sent to the weapons platforms for execution.

### 7.3.6 Components

The JTE system consists of human and automated components.
7.3.6.1 Human component

The human component of the JTE system is the JTE operator. Generally the operator is an intelligence agent knowledgeable about the region and the enemy units operating within the region. These operators rely on the incoming information, expert knowledge, opinions of other operators, and the rules determined by their commanding officer to guide their decision making process.

7.3.6.2 Automated component

The automated component of the JTE system is the JTE terminal and the supporting information distribution infrastructure. The JTE terminal manages storing and distributing nominations, communications between operators, collecting available weapons lists, and pairing weapons to targets. The pairing of weapons to targets is done via the use of a distributed auction algorithm.

7.3.8 TCTA

The JTE system is built on top of an existing military system, the Time Critical Target Aid (TCTA). The TCTA provides the windowing system, map background, track representations, and imagery feeds. JTE adds the target nomination system, nomination distribution, weapons listing, and weapon-target pairing capabilities.
8.0 Areas for improvement

Two main areas for possible improvement have been identified in the JTE system: imagery recognition and track data overload.

8.1 Imagery

JTE operators rely on imagery information to make positive identifications of targets. Thus, imagery provides a highly useful means of determining target identity. However, JTE operators can not view all imagery that may be available for all tracks. This inefficiency is the first of the areas for possible improvement.

The relevant technology for this inefficiency is Automatic Target Recognition (ATR). ATR research is several decades old and recent advances in computational technology have made it feasible for use in real time systems [30]. ATR systems take imaging data such as synthetic aperture radar (SAR) and use feature extraction, neural networks and a variety of other techniques to identify targets within the images. A detailed examination of ATR technologies and recommendations for or against their use is beyond the scope of this thesis, however it is the opinion of the author that these techniques should prove useful in assisting with the imagery overload problem.

8.2 Track data

JTE operators also may receive too many tracks to monitor at once. Engineers estimate that in heavy conflict hundreds of tracks may be generated within the area monitored by a JTE operator. Clearly improvements can be made in this area.
JTE operators use heuristics to determine which tracks to examine. However, the inability of one operator to apply these heuristics to all tracks at once suggests that by encapsulating this knowledge in the form of an artificial intelligence agent, more tracks can be examined, increasing overall efficiency.

The following chapter details the design of an artificial intelligence system to address the track data overload problem of the JTE system.
9.0 Recommended system

9.1 Design

The goal of this recommended system is to replicate, with as much detail and accuracy as possible, the analytical process of JTE operators in an artificial intelligence agent. This agent will assist the JTE operator by providing preliminary analysis of all tracks, allowing the JTE operator to pick and examine the targets that are most likely TCT.

The design process followed by the author is based on the conclusions formed from the analysis of hybrid systems in the first section. Specifically, the main characteristics of the problem were first determined. Next, applicable artificial intelligence techniques were identified and compared. Then the most promising of these techniques were selected. Hybridization and integration of these techniques was then examined to determine applicability. Finally a total solution was devised from the applicable techniques. As determined in the analysis of hybrid systems, this process tends to produce minimally complex designs that complete the required task, while taking consideration of all relevant designs.

9.2 Hybrid Design Process

9.2.1 Main characteristics and applicable techniques

As mentioned previously, JTE operators use heuristics and behavioral studies of the enemy to determine which tracks are likely to be TCT. JTE operators are intelligence
agents who basically represent experts in the domain of targeting TCT. These factors point towards the use of expert systems that can capture this domain knowledge in the form of rules and reason on a high-level about the behavior of the targets.

However, the heuristics often use non-precise terminology. For example, one rule might be “TELs cannot travel fast over rough terrain”. Fast and rough are clearly imprecise terms, but the rule still has significance. Thus, fuzzy logic is another possible candidate for use in the system.

Occasionally, a rule may inherently contain a pattern-matching task. For example, a rule may be “TELs travel in small convoys of four to six vehicles to hide their radar signature”. This implies that convoys of four to six vehicles should be searched for possible TCTs. To recognize a convoy of vehicles, low-level processing may be used. Thus, neural networks are candidates for use in the system.

A case-based reasoning application is not very feasible in this situation. The data for this problem are the characteristics of the tracks – where they originate, where they go, how fast the vehicle moves, etc. The solutions to this problem are nominations of the target as TCT. Therefore, there are many different inputs with only two possible outputs. Case-based reasoning systems work best with domains containing small numbers of inputs to aid in indexing and retrieval, and a large variety of outputs. Thus, case-based reasoning is not a possible candidate for consideration as a major technology in this problem.
Similarly, genetic algorithms are not applicable to this problem. Genetic algorithms are useful for optimizing parameters or in search problems along a problem space. In this problem, the goal is only to capture existing heuristics used by experts, not to generate new heuristics.

9.2.3 Selection of optimal techniques

According to the hybrid design process, the optimal set of techniques should be determined after determining the applicable technologies. Here the choice is among expert systems, fuzzy logic and neural networks. To decide which techniques to use, the minimal set of technologies should be applied that solve all aspects of the problem while reducing complexity.

Clearly, using expert systems, fuzzy logic, and neural networks would solve the problem. However, the use of neural networks can be eliminated through the use of symbolic reasoning to implement the low-level pattern matching. For example, a symbol in the reasoning system could represent a convoy. The rule for a convoy would be “if vehicles are close to one another and moving together, they are a convoy”. The rule for flagging the convoy would be “if a convoy is small it is likely to contain a TCT”. Fuzzy logic membership functions and additional rules would determine if a convoy were small, vehicles are close to one another, and moving together.

Here a design decision was made weighing the added complexity of additional rules and symbols versus the overall decrease in system complexity by removing neural
networks from the mix. The author decided that uniformity in architecture would improve prototyping, developing and maintenance enough to validate this decision.

9.2.4 Hybridization and integration level

At this stage, the hybrid design process requires an examination of the method of hybridization and the level of integration for the system. In this situation, hybridization of the fuzzy logic and expert system techniques is relatively simple and has been proven in previous systems [1]. Thus, a fuzzy expert system was chosen as the method of hybridization and level of integration.

9.2.5 Total solution

The final recommended artificial intelligence solution to assist in the track data overload problem is therefore a hybrid fuzzy expert system. Additionally, it was previously determined that ATR techniques should be used to address the problem of imagery recognition. Details of the recommended system appear in the following section.

9.3 Implementation details

9.3.1 Overview

The main design of the JTE artificial intelligence system is diagramed in Figure 18. The ATR module consists of a variety of ATR techniques, each picked specifically to address a different imagery source. Although this part of the system does not necessarily include artificial intelligence techniques it is mentioned for completeness.
Figure 18. JTE hybrid artificial intelligence system

JTE terminal window

- stationary target track
- moving target track

Missile launch site

Urgent flag from fuzzy expert agent

Imagery window

TCTA infrastructure (UNIX based platform)

Fuzzy expert agent (C++ or Java code)

Numerical module (C++ or Java code)

Labeling module (C++ or Java code)

ATR modules (unknown, but likely C++ or Java code)
The track data assistance module (Figure 18) consists of a large set of fuzzy expert system agents that act as filters for all the track data. Each agent represents a heuristic used in selecting tracks for further examination. Within the module, agents can be activated or deactivated to change the mix of heuristics applied. This allows the JTE operator to adjust how many targets are flagged for examination.

9.3.2 Interfacing with JTE system

The track data assistance module will be built on top of the existing TCTA infrastructure. Data feeds from the various information sources are transmitted as simple values such as latitude, longitude, velocity, heading, and textual labels. Therefore, integration of this data into the fuzzy expert modules should be relatively simple.

9.3.3 Fuzzy expert agents

A sample fuzzy expert agent for the heuristic “targets nearby and moving away from a missile launch event are very likely to be TCTs” is described as an example of a fuzzy expert agent. The fuzzy expert agent was constructed as follows.

Figure 19. Symbolic representation of heuristic
First, a fuzzy relation was developed to capture the high-level logic involved (see Figure 19). Using this, it is apparent that the relations between the target and the missile launch event are the movement direction and the vicinity. Thus the movement direction and vicinity need to be calculated somehow. The calculations of variables for use in fuzzy membership functions are accomplished through numerical modules that interface with the data feeds. These calculate values such as distances and directions between targets and events, headings, and velocities.

Once these input values are computed, they are fuzzified using the membership functions shown in Figure 20.

**Figure 20. Membership function for movement direction**

- **Towards**
- **Neutral**
- **Away**

**Figure 21. Membership function for distance**

- **Small**
- **Medium**
- **Large**
Figure 22. Fuzzy associative map for TCT likelihood. Based on distance and direction.

<table>
<thead>
<tr>
<th>Distance</th>
<th>Small</th>
<th>Medium</th>
<th>Large</th>
</tr>
</thead>
<tbody>
<tr>
<td>Towards</td>
<td>Medium</td>
<td>Low</td>
<td>Low</td>
</tr>
<tr>
<td>Direction</td>
<td>Neutral</td>
<td>Medium</td>
<td>Low</td>
</tr>
<tr>
<td>Away</td>
<td>High</td>
<td>Medium</td>
<td>Low</td>
</tr>
</tbody>
</table>

A fuzzy associative map that determines the strength to apply the rule in raising priorities of tracks is shown in Figure 22. Once a high priority is assigned to a track, the labeling module is activated to label the track as urgent via the TCTA infrastructure.

The simplicity of the fuzzy expert agents and the existence of previous fuzzy expert systems should make implementation fairly simple with generic C++ or Java. JTE is based on a UNIX platform which supports such development tools.

9.3.4 Numerical module

The numerical module performs basic computations to prepare the data for use in the fuzzy expert agent. This module likewise can easily be implemented using standard C++ or Java techniques.

9.3.5 Labeling module

The labeling module receives notification from the fuzzy expert agent to flag tracks as urgent. This is done through messaging with the JTE system via the TCTA infrastructure. Messages passed through the TCTA consist of plain text and digits thus, this module needs no special technology to implement other than C++ or Java code.
9.4 Justification

9.4.1 ATR module

ATR systems use is beyond the scope of expertise in this thesis. However, ATR systems have been well studied and have shown promising results [30]. ATR techniques are planned for use in the JSTARS system [18], thus proving the applicability of this technique to imagery processing.

9.4.2 Fuzzy expert agent module

9.4.2.1 Capturing expert knowledge

The fuzzy expert system module satisfies the need to capture expert knowledge in the form of heuristics, as shown in the sample heuristic.

9.4.2.2 Applying expert knowledge

Through the use of numerical modules, the fuzzy expert agents can apply the heuristics to the data feeds. Via interactions with the TCTA infrastructure, flags can be added or removed from the tracks, thereby applying the expert knowledge to select targets likely to be TCT.

9.4.2.3 Increased efficiency

Instead of sifting through numerous tracks, JTE operators should be able to scan through flagged tracks where the likelihood of finding TCT are greater. This should lead to increased TCT detection and ultimate destruction. Since JTE operators can switch on or off each agent, an optimal amount of flagged tracks will appear on the screen, preventing too much or too little advice.
9.6 Significance of case study

Within this case study, the hybrid design process outlined in the hybrid systems background study was applied to propose a viable artificial intelligence solution aid in the data intensive process of detecting surface TCT. The system was designed to incorporate proven artificial intelligence techniques reviewed in the background research in a manner which would keep complexity to a minimum while satisfying all the constraints imposed on the system. Since multiple artificial techniques were used in an integrated fashion, the proposed design can be classified as a hybrid intelligent system. As a hybrid intelligent system, it combines the characteristics and capabilities of its constituent techniques to solve a problem intractable by either method alone. Thus this case study illustrates the viability using hybrid artificial intelligent systems to aid in the processing of data intensive systems.
10.0 Conclusions and Further Research

10.1 Conclusions

10.1.1 Background research

In this thesis, traditional artificial intelligence techniques have been found to have serious limitations in their abilities to solve complex problems. Symbolic approaches have difficulty dealing with low-level reasoning. Connectionist and numeric approaches lack the ability to handle high-level logic. These deficiencies prevent any one of these techniques from fully capturing the knowledge process of humans. The capabilities and limitations of each technique were summarized to highlight this observation.

In the background research of this thesis, the current state of hybrid intelligent systems was analyzed to highlight the advantages, disadvantages, and recommended uses for the various systems. A summary of hybrid system characteristics was provided as a reference for hybrid system development.

Based on the hybrid system analysis, it was concluded that though hybrid systems combine characteristics of the underlying techniques, no significantly new capabilities are added to the overall systems. From this observation, a hybrid design process was developed to guide the development of hybrid systems.
10.1.2 Case study

The case study of this thesis examined the applicability of hybrid intelligent systems in the complex information management problem of identifying surface time-critical targets. Through the design of a fuzzy expert agent module to optimize the JTE system, the hybrid design process outlined in the background research was verified as a viable method for developing efficient, effective hybrid systems. In the process it was also shown that hybrid intelligent systems can aid in managing the complexity of multi-source information systems.

10.2 Further research

10.2.1 Case study

The fuzzy expert agent module and associated modules designed in the case study should be prototyped and tested for validity. For the ATR component, the most effective ATR techniques should be chosen and implemented.

10.2.2 Basic research

Research is continuing in hybrid artificial intelligence techniques and should yield novel ways in which to combine traditional technologies. As these new combinations come into existence, more research will be required to evaluate their capabilities.

However, combining these techniques yields hybrids that exhibit characteristics of the original techniques. In order to incorporate new capabilities not represented in these original techniques, new models of knowledge must be discovered. Therefore efforts
should continue to be made to expand the array of available technologies for hybrid system development.
REFERENCES


