A Human-Centered Approach to Developing Safe Systems

by

Mirna Daouk

Engineering Degree (2000)
Ecole Polytechnique (Palaiseau, France)

Submitted to the Department of Aeronautics and Astronautics
in Partial Fulfillment of the Requirements for the Degree of

MASTER OF SCIENCE
at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

FEBRUARY 2002

Signature of Author:.....
Department of Aeronautics and Astronautics
January 14, 2002

Certified by:..
Nancy G. Leveson
Professor of Aeronautics and Astronautics
Thesis Supervisor

Accepted by:.....
Wallace E. Vander Velde
Professor of Aeronautics and Astronautics
Chair, Committee on Graduate Students

A Human-Centered Approach to Developing Safe Systems

by

Mirna Daouk

Submitted to the Department of Aeronautics and Astronautics
on January 15, 2002 in Partial Fulfillment of the
Requirements for the Degree of Master of Science

Abstract

On account of the several advantages it provides, such as the improvement in performance, increasingly complex automation is being built into existing systems. As a result, human-automation interactions are changing in nature, and new sources of errors and hazards are being introduced. The need for reducing human errors without sacrificing the benefits of computers has led to the idea of *human-centered system design*; little work, however, has been done as to how one would achieve this goal. This paper provides a methodology for safe, human-centered design of systems including both humans and automation. It also describes a new approach to structuring specifications, called Intent Specifications, which captures design rationale and assumptions made throughout the design process. The proposed methodology combines task allocation, task analysis, simulations, human factors experiments, formal models, and several safety, usability, and performance analyses. An air traffic control conflict detection tool, MTCD, is used to illustrate the methodology.

Thesis Supervisor: Nancy G. Leveson

Title: Professor of Aeronautics and Astronautics

Acknowledgments

"Come to the edge
He said. They said: We are afraid.
Come to the edge
He said. They came.
He pushed them, and
they flew..."
-Guillaume Apollinaire

During my three semesters at MIT I faced the edge more than once. And god did I want to turn around and run away. But fortunately there was always someone to push me forward. Now is the time for me to express my thanks.

I would like first to thank my advisor, Professor Nancy Leveson, for all her support and guidance. Prof. Leveson's unarguable success in her field helped me set the standards for my own work and gave me a role model to look up to. Thank you Nancy for believing in my capabilities.

Thanks is of course owed to EUROCONTROL, especially Geraldine Flynn, Nadine Pilon, and Alistair Jackson, for their precious comments and for tolerating my nagging to get the documents and information I needed. My thanks also go to all my lab-mates, Natasha, Marc, Maxime, Kristina, Ed, Israel, Masa, JK, and John, for the work achieved together and for the good times. It was a pleasure working with you in such a nice and friendly environment. Thanks also to Katie, Nick, Thomas, Victor, Polly, and Elwin for the good laughs (yeeaah!). Especial thanks are also due to Walt Disney Imagineering for giving me the opportunity to apply the knowledge I acquired at MIT: my internship gave me good insight on the applicability of my work in industry. Thanks also to the Ousseimi Foundation and to the Fondation de l'Ecole Polytechnique for their financial support

A warm thank you is due to my friend Joe. My MIT experience wouldn't have been the same had I not known you. Thanks also to Karen, Hayley, Richard, Nadine, and Rana for their kind friendship. An enormous thank you goes of course to Adrien- for more reasons than I could state in this page. But most of all, thank you for being there when most people would have given up on me.

Last but not least, I would like to thank my entire family: my parents, my sister Carine, my brother Ihab, as well as Ghina, Maher, Hassan, Elaine, and all the others. Thank you for your love. I hope you are as proud of me as I am of you.

Table of Contents

List of Figures

List of Tables

Introduction

The term "*human-centered system design*" is used frequently in the literature (e.g., [**7**,**8**]), but there have been few proposals or methodologies addressing how exactly one might achieve this goal, especially for safety-critical systems. When the automation is being designed or implemented, it is commonly believed that both the human and the automation should be taken into account. No single methodology, however, addresses all the issues involved, for example: Is the behavior of the automation (or software) transparent enough so as to support the operator in his/her tasks? Is there a risk of mode error and loss of situation awareness [**65**]? Are the human and the task he/she is required to accomplish matched correctly [**57**]? Does the operator have a correct and sufficient understanding of the automation's behavior? Numerous questions can be and have been raised, and several have been discussed in the literature. Most of the answers proposed, however, are partial, reserved to the Graphical User Interface (GUI)/Human Machine Interface (HMI), or hard to implement. More importantly, few authors identify the criticality of recording design rationale and the assumptions underlying the design choices for safe change of the design and for system evolution. In this paper, we describe a methodology for human-centered, safety-driven design of systems that include both humans and computers.

The proposed methodology covers the whole system life cycle, starting with the definition of its high-level goals and purposes and continuing through operation. Safety and human factors are often considered at too late a stage in system development to have adequate impact on the system design. It has been estimated that 70-90% of the decisions relevant to safety are made in the early conceptual design stages of a project [**35**]. Relying on after-the-fact safety assessment emphasizes creating an assessment model that proves the completed design is safe rather than constructing a design that eliminates or mitigates hazards. Too often, after-the-fact safety assessment leads to adjusting the model until it provides the desired answer rather than to improving the design. In the same way, when the human role in the system is considered subsequent to the basic automation design, the choices to ensure usability and safety are limited to GUI/HMI design, training, and humans adapting to the newly constructed tools. Also, if the involvement of human factors occurs late, then the impact of the human-system interaction on the system performance may not be fully analyzed, or alterations will be required at a later stage than is desirable, i.e. incurring delays to the program or

extra cost in re-design, or both [**39**]. The latter approach has been labeled "*technology-centered design*" and has been accused of leading to "*clumsy automation*" [**72**] and to new types of accidents in high-tech systems.

In previous work, Leveson has defined what she calls Intent Specifications [**47**], which are based on means-ends abstraction. The design of Intent Specifications, using ideas from Rasmussen's means-ends abstraction hierarchy, was based on fundamental knowledge about human problem solving and also on system theory and basic system engineering principles. While the Rasmussen/Vicente specifications [**71**] are aimed at design of the user interface, we have tried to extend the idea to design of the entire system, including the automation. Intent Specifications have been used to specify several complex systems, including TCAS II (an aircraft collision avoidance system). This previous work, however, did not integrate the design of the operator tasks or detail a human-centered design methodology for developing intent specifications. The work presented here describes how those goals can be accomplished using as a test-bed a new air traffic control (ATC) Medium Term Conflict Detection (MTCD) function currently under evaluation at the EUROCONTROL Experimental Centre (EEC) in Bretigny, France. This work is part of a collaboration project between the Software Engineering Research Laboratory (SERL), M.I.T., and EUROCONTROL. The goals of this project are to demonstrate how to lead a safety analysis for an automated system using MTCD as a test-bed, and to recommend a safety methodology to apply when introducing or planning changes to the Air Traffic Management (ATM) system.

Chapter 1

Air Traffic Control and Human-Centered Design

Air traffic forecasts are normally based on passenger demand and economic factors, under the assumption that there is no limitation in the air traffic system itself; it is however obvious that there are constraints limiting the expansion of the number of flights. The Air Traffic Controller is confronted with the multiple, sometimes-conflicting goals of "maintaining the *safe* and *expeditious* flow of aircraft through the airspace" [**49**]. On the whole, the air traffic control (ATC) system is remarkably safe, given what it is asked to do, partly because of the redundancy that was built in it, the high level of professional ATC work force, the large error margins (e.g., in the separation criteria for aircraft), and the loose coupling (so that errors are contained and do not propagate rapidly throughout the various components). Yet, this safety, which was primarily based on the *human* control, has sacrificed efficiency and led to considerable delays and wider-than-necessary separations in the air.

1. Automation in Air Traffic Control

A typical automated system for Air Traffic Control can be divided into three principle subsystems whose functions involves sensing, planning, and controlling.

Sensing subsystem: All of the components, including ground radars, mode C/S transponders, and ground computers and tools, that contribute to generating aircraft position tracks on ATC monitors.

Planning subsystem: Planning the most efficient landing order and assigning optimally spaced landing times to all arrivals. The times are planned such that traffic approaching from all directions will merge on the final approach without conflicts and with optimal spacing. Planning tools also assist the Air Traffic Manager in rerouting traffic from an overloaded sector to a lightly loaded one, a process known as gate balancing, and efficiently rerouting and rescheduling traffic in response to a runway reconfiguration or a weather disturbance. Finally, planning tools help controllers re-plan

traffic quickly in response to several special situations, such as missed approached, runway changes, and unexpected conflicts.

Controlling subsystem: Providing the controllers handling descent traffic with tools to implement the traffic plan. The Air Traffic Control subsystem has essentially four functions:

- Traffic separation and surveillance
- Conflict detection
- Conflict resolution
- Emergency handling

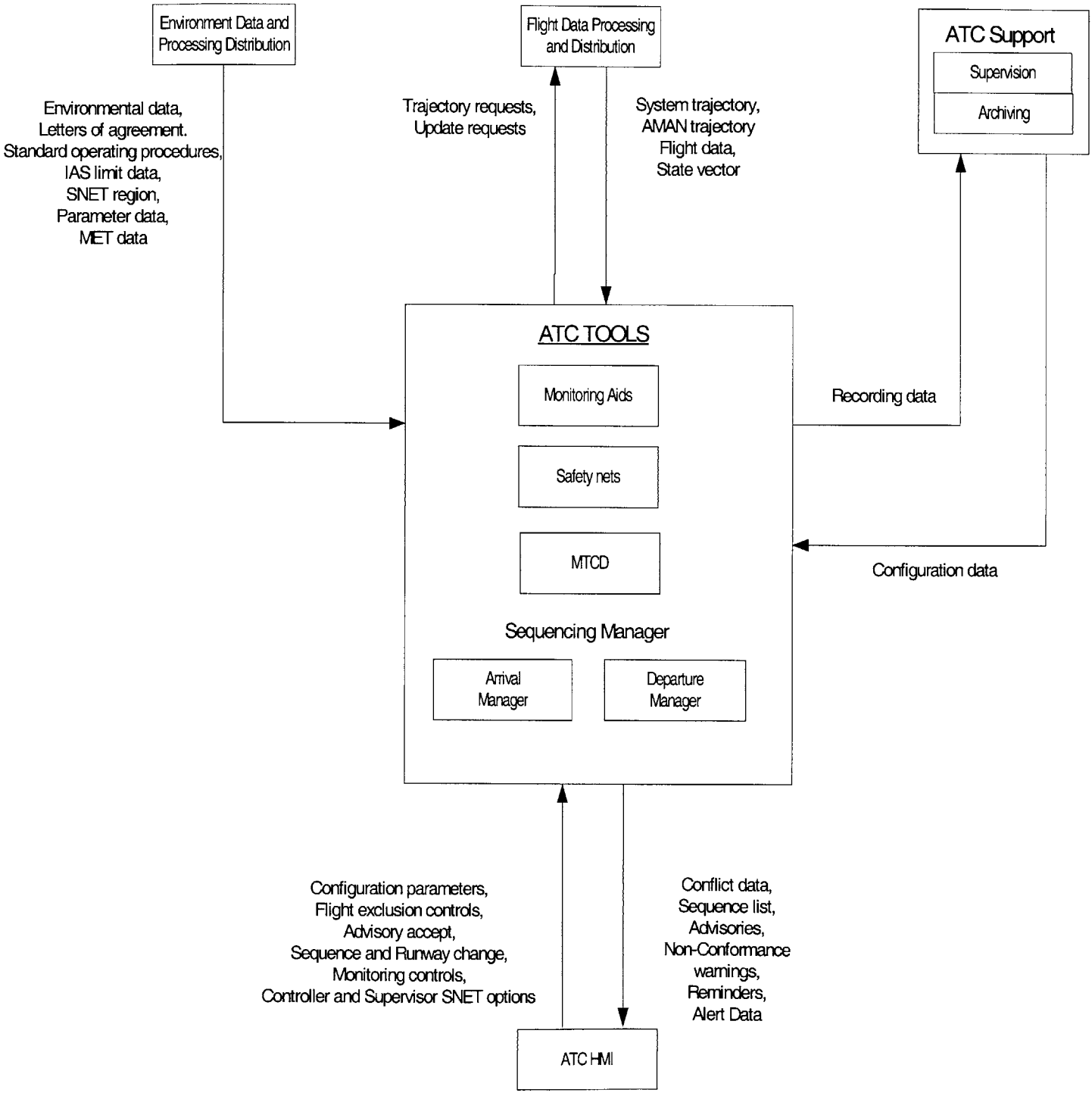
Figure 1 shows the different tools present in the EUROCONTROL- EATCHIP III (European Air Traffic Control Harmonization and Integration Programme, Phase III) system. EATCHIP is a cooperative program of the European Civil Aviation Conference (ECAC) Member States, coordinated and managed by EUROCONTROL in partnership with the national ATM providers of ECAC and other institutions. Its objective is the harmonization and the integration of European ATM services.

The Flight Data Processing and Distribution Function and the Environment Data Processing and Distribution Function are part of the sensing subsystem, and their outputs are used by the planning and the controlling subsystems. The planning subsystem will consist mainly of a Central Flow Management Unit. Finally, the controlling subsystem uses tools such as Mona, Safety nets, and MTCD. The Arrival and Departure Managers appear in both the planning and the controlling subsystems (MTCD and MONA as well, to a certain extent). MTCD is the first tool that is being developed, implemented and tested. The other tools will then be added progressively. By adding these new tools, there will certainly be a temptation by the developers to make changes to the system functions and components, and a sound safety methodology is therefore crucial to insure a safe and efficient transition from the current to the future ATM system. The boundaries of system studied in this paper are defined in **Appendix A**. More details on the EUROCONTROL system can be found in [**27**].

2. Why Air Traffic Control?

ATC/ATM is a very interesting field to demonstrate our human-centered, safety-driven approach because it requires both a high level of safety and strong human factors bases. ATC has in fact been

Figure 1: CNS/ATM Functional Architecture



growing in terms of demands and numbers, flying capabilities of aircraft, navigation facilities, computers in systems, and the controller's equipment and tasks, but the human capabilities and limitations (learning, attending, understanding, making decisions, resolving problems, predicting, etc.) have not. Modernization and automation seem therefore essential. New automated systems include straight-forward assistance or replacement of some vulnerable human functions (such as electronic data link to replace voice communication or the use of predictive displays to support the controller in his/her prediction tasks), but some (proposed or implemented) ATC tools may include replacement of key perceptual and cognitive functions, such as suggesting and perhaps implementing aircraft flight path changes to avoid conflict and maximize efficiency. This is partly the case in the new EUROCONTROL-EATCHIPIII system. The problems inherent in upgrading an airspace management system are further exacerbated by the fact that a completely new system is not being designed at one time but changes will be introduced in stages. There needs to be a way to add tools and new technology and evolve functionality over time without compromising the safety of the existing system and infrastructure. In particular, it is vital to make sure safety (and maybe efficiency) is not being compromised by marginalizing the human controller's ability to effectively monitor the process, intervene as spot errors or failures in the software or environmental disturbances require, or assume manual control if the automation becomes untrustworthy.

The Air Traffic Control system should be upgraded using a human-centered approach. The degree of involvement of human factors in any system depends on a number of factors, such as how critical human involvement is, the novelty of the design, and the degree to which the system requires human interaction (during normal operations). These three factors appear to be maximized for ATC/ATM systems, since ATC is a highly human-interactive system, depends on human intervention for safety of operations, and is currently evolving rapidly [**39**].

Air Traffic Control, despite what some have argued, is safety-critical. A "safety-critical" system or software is not defined in terms of catastrophic failure, but in terms of "hazards" and the ability to contribute to a hazard. A safety-critical system or subsystem is one that can contribute to the system getting into a hazardous state, eventually experiencing a significant loss. Accidents do not usually have single causes, since the simpler accident potentials are usually (and rather easily) eliminated from the systems. The system safety goal is then to eliminate or mitigate the occurrence of hazards from a system point of view.

3. The Safety and Human-Centered Methodology

The methodology presented in this document combines human factors and safety analyses, using both formal and informal methods. The approach aims at completing the current efforts made on the HMI side, but not replacing them.

Figure 2 shows the overall structure of the methodology. The steps in the middle column represent the general system engineering activities. The right column shows special safety engineering activities and those in the left column represent human factors engineering. This figure is notional only---the system engineering procedures (shown in the middle) integrate the human factors and safety analysis throughout the development and operations processes and also involve more iteration and feedback than shown. In addition, some of the analysis procedures in the right column, such as mode confusion and human error analyses, actually represent an overlap between safety and human factors engineering and their placement in the right column is arbitrary.

The methodology is supported by the Intent Specifications structuring approach mentioned in the Introduction. Intent Specifications organize system specifications not only in terms of "what" and "how" (using refinement and part-whole abstractions) but also in terms of "why" (using intent abstraction) and integrate traceability and design rationale into the basic specification structure. Each level of the Intent Specifications supports a different type of reasoning about the system and uses a different model of the system. Each level also includes information about the verification and validation of the system model at that level. By organizing the specification in this way and linking the information at each level to the relevant information at the adjacent levels, higher-level purpose or intent, i.e. the rationale for design decisions, can be determined. In addition, by integrating and linking the system, software, human task, and interface design and development into one specification framework, Intent Specifications can support an integrated approach to system design. Thus, Intent Specifications permit a level of traceability not normally found in system specifications, and allow changes to be made much more quickly and easily. Being able to trace a particular design feature or operator task for instance to a box in a fault tree or to a principle in the task allocation or usability analysis will allow decisions to be made about whether and how that feature or task can be changed.

There are six levels in an Intent Specification (see **Figure 3**). Level 1 supports reasoning about system-level properties such as goals, task allocation, operator goals and responsibilities, high-level requirements, design constraints, and hazards during the earliest stages of the system development

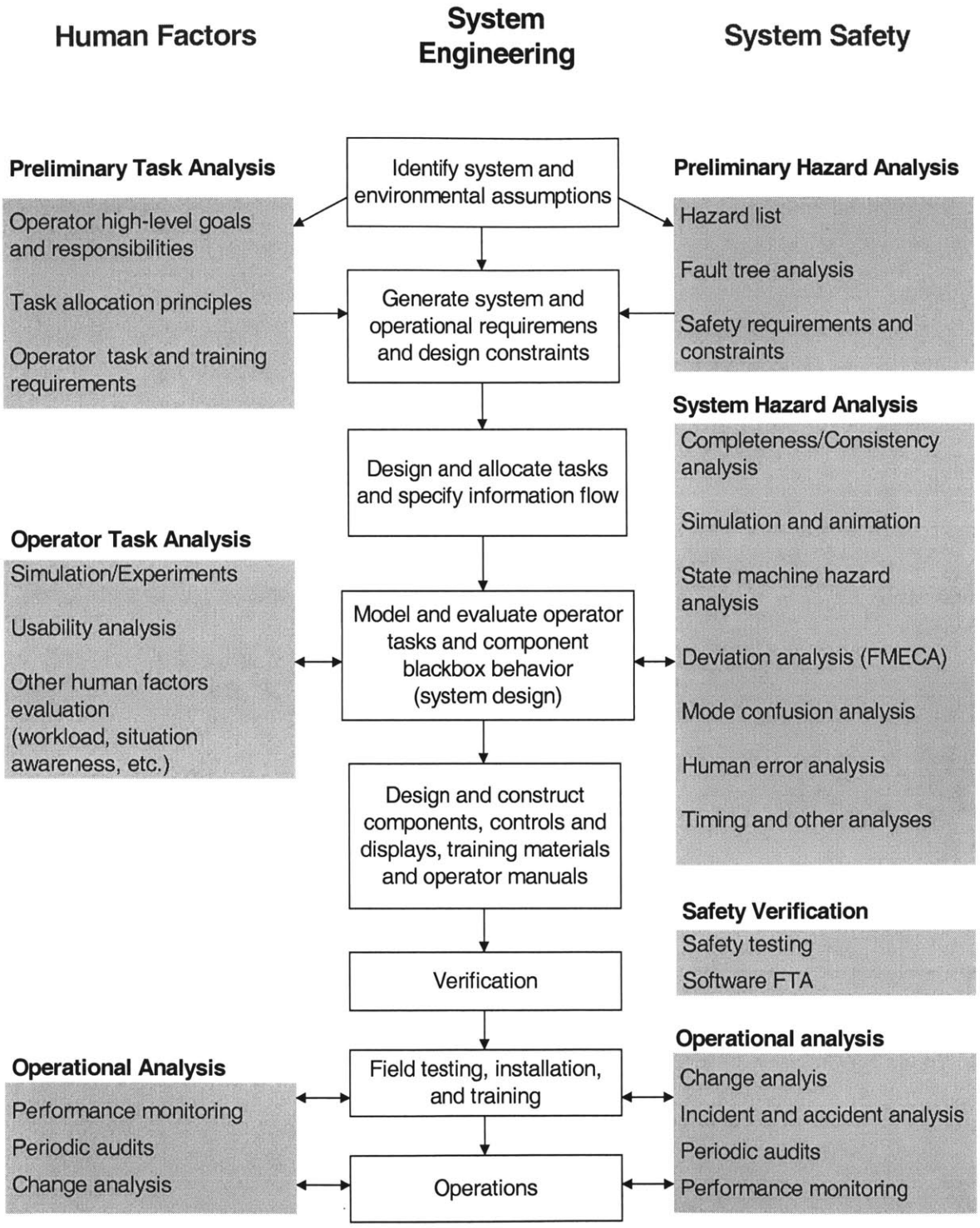


Figure 2: A Human-Centered, Safety-Driven Design Process

process and later acts as documentation of the overall system concept and requirements. A portion of the first level of the MTCD Intent Specification is presented in **Chapters 2-4**.

Level 2 includes the system design principles upon which the logical and physical design at the lower levels is based and through which the goals and constraints at the highest level are satisfied (see **Chapter 5**). The models and specifications at this level may be subjected to scientific and system analyses to evaluate design alternatives with respect to the higher-level goals, constraints, and identified hazards. Level 2 also includes principles and bases for a series of simulations and experiments aimed at verifying and refining the operator's tasks and the HMI design principles. All Level 1 requirements, constraints, task allocation principles, human factors and hazards are mapped to their corresponding design features at Level 2.

The third level contains formal models of the system components blackbox behavior, including the operator tasks, interfaces and communication paths between components, and transfer functions (blackbox behavior) for each new system component (see **Chapter 6**). The models at this level are executable and mathematically analyzable. The information at this third level is used to reason about the logical design of the system as a whole (the system architecture) and the interactions among components as well as the functional states without being distracted by implementation issues.

The fourth and fifth levels of an Intent Specification document the physical design and the physical implementation respectively. The sixth level includes the information necessary for and generated during operations. The following chapters will focus only on the first three levels of the Intent Specification.

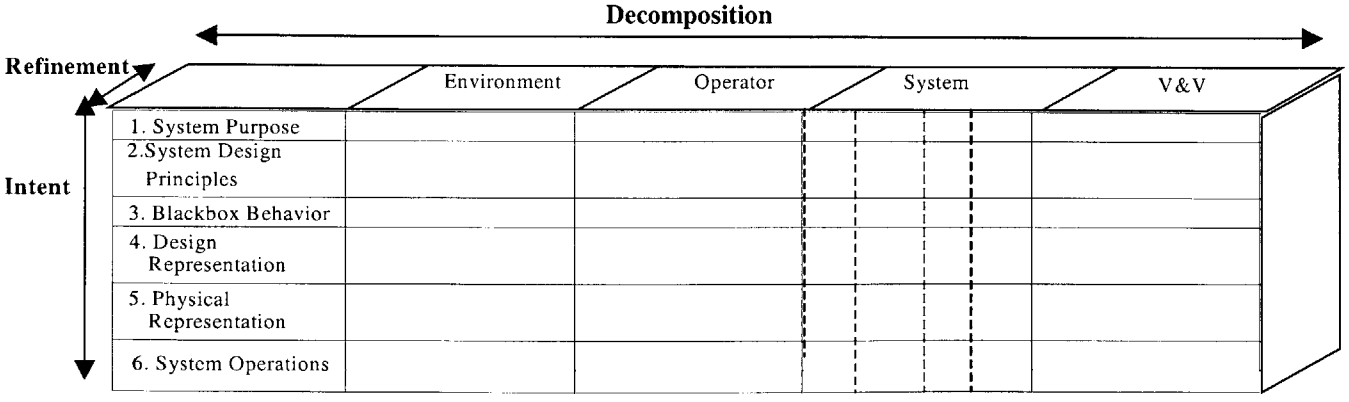


Figure 3: The Form of an Intent Specification

4. Medium Term Conflict Detection (MTCD)

MTCD is an EATCHIP III added function. The automated MTCD function will assist the controllers in monitoring the air situation continuously and providing conflict data to the controllers through HMI. Controllers monitor these operational data on situation displays. They will remain responsible for the assessment of conflicts, as well as reacting to them. MTCD will inform the controller of aircraft conflicts up to 20-60 minutes in advance, and of special use airspace penetrations and descents below lowest usable flight level. Controllers can influence MTCD operational behavior by excluding and re-including individual flights from conflict detection calculations. These interactions, and all interactions with respect to conflict display, conflict display acknowledgement, etc., will be governed by the HMI.

In the European ATC system, the controlling tasks are performed by two controllers, the planning/strategic controller (PC) and the executive/tactical controller (TC). The high-level goals of the PC and the TC are similar, and their areas of responsibility are the same, but their areas of interest are different: the PC handles the pre-sector and sector entry traffic and works on the in-sector and sector exit areas only when his/her workload allows it and the TC requests assistance. The TC is thus responsible for short-term management of in-sector and exit traffic (including radio communication with pilots), whereas the PC is more concerned by the medium-term issues. In reality, however, the two controllers work very closely together, sharing tasks spontaneously, and communicating with gestures as much as words. The main goal of this division of responsibilities is a better management of the controllers' workload. Although MTCD is available to both controllers, it is primarily of interest to the PC.

Chapter 2

Goals and Responsibilities

Level 1 assists system engineers in their reasoning about system-level properties during the earliest stages of the system design (concept formation) and later acts as documentation of the overall system concept and requirements. We are particularly concerned about safety-critical systems, so we start with a preliminary hazard analysis (PHA) to understand the potential system hazards. Such a PHA was performed for MTCD but is not presented in the present paper. **Appendix A** provides additional information on this topic.

A preliminary task analysis (PTA) is then performed. The PTA consists of cognitive engineers, human factors experts, and operators together specifying the goals and responsibilities of the users of the new tool or technology, the task allocation principles to be used, and operator task and training requirements. The involvement of the operators at this stage is very important, as they are the final users of the system and because they can provide a description of their needs for assistance to the system designers, engineers, or managers. The PTA and the PHA go hand in hand, and several iterations are necessary as the system designers acquire a better understanding of the operators' responsibilities and of the different human factors to take into consideration.

From the resulting high-level operator-centered requirements and user requirements, the system requirements, design constraints, and system interface requirements are then developed. The functional goals are given before the analysis, but they may be adjusted as we learn more about the hazards and the operators' tasks.

In this chapter, we cover the first part of the process, identifying the goals of the system and the goals and responsibilities of the human operators. We will also present some human factors that have to be taken into account when designing any system. **Chapters 3** and **4** address the remainder of the Level 1 analysis.

1 MTCD: Functional Goals

Our methodology begins with identifying the high-level functional goals for the new system or component(s) and the assumptions and constraints on the new tool or component design arising from the environment. Following are the high-level goals for the MTCD system. The goals are stated in very general terms at the early stages of the project, adjusted as the preliminary analyses are performed, and then refined into testable and achievable requirements and constraints.

- Provide a conflict detection capability to air traffic control for all flights in the Area of Operation.
- Take over part of the air traffic controllers' monitoring task to keep aircraft separated.
- Provide a planning tool to air traffic controllers.
- Provide controllers with enough time to assess and if necessary to resolve conflict by deliberate action.
- Be considered for implementation in all European Civil Aviation Community (ECAC) areas irrespective of their complexity and density.
- Allow use of different separation criteria between aircraft when required, and increase airspace capacity.
- Help keep the workload of the controllers within acceptable and safe limits despite the foreseen increase of traffic.

2 ATCO High-Level Goals and Responsibilities

The next step in the process consists in identifying the operators' high-level goals and responsibilities. Indeed, we consider any new or altered operator tasks related to the new tool to be within the system because such new tasks must be designed together with the other new parts of the system. In the case of MTCD, the controllers' high-level goals and responsibilities are similar in the current system and in the new system, as MTCD is only intended to be a decision support system.

The air traffic controllers are responsible for:

- Managing en-route flights, arrivals, departures

- Insuring traffic separation/resolving conflicts
- Managing pilot's requests (weather, conflicts, re-routing, etc.), managing adjacent controller's request (inter-sector communication) and providing pilots with relevant information (trajectory, weather, runway status, etc.)
- Managing aircraft emergencies
- Managing adjacent controller's requests (inter- and intra- sector coordination) and providing colleagues with relevant information (intra-sector communication; hierarchical communication)
- Carrying out the handoff from the previous controller to the next controller (inter-sector or PC/TC)

As mentioned in **Chapter 1**, the controlling tasks in the EUROCONTROL system are performed by two controllers, the planning controller (PC) and the executive/tactical controller (TC) (**Figure 4**), the main goal of this division of responsibilities is a better management of the controllers' workload.

- **Assumption**: Each controller, PC and TC, has his/her independent set of ATC tools (MTCD, MONA, Safety nets, etc.). The planning tools are shared by the whole system.

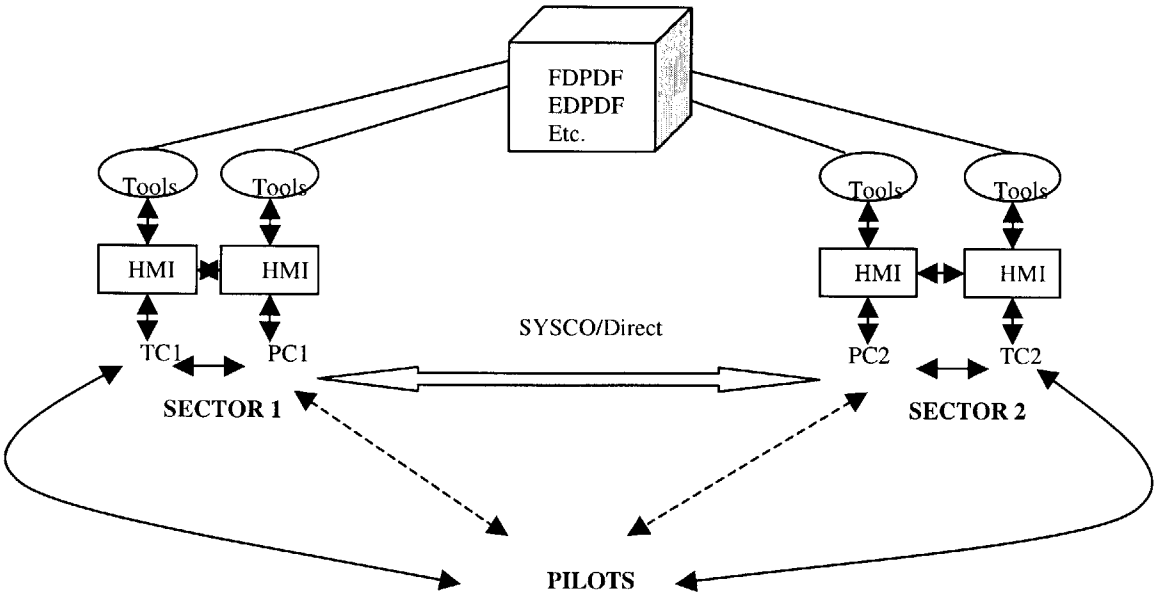


Figure 4: Elements of the controlling sub-system

(Acronyms are defined in **Appendix B**; "Tools" designates MTCD, Safety Nets, MONA)

Given this division of tasks between the PC and the TC, it is important to specify the responsibilities of each of the controllers to better understand how they will be affected by the introduction of the new system. We specify *all* of the TC and PC goals and responsibilities, not just those directly related to the new tool. We include all responsibilities because any safety or usability analysis will require showing that the tool does not negatively impact any of the controller activities.

The responsibilities of the PC are the following:

- Inter-sector coordination
    - Entry and Exit problem resolution
    - Handoff
    - Awareness and management of requests
- Coordination with TC
    - Allocate problems to the TC
    - Handoff
    - Advise the TC of problems
    - Monitor the TC workload
    - Awareness and management of requests
- System updating
    - Modify trajectories where required
    - Other information (runway, weather experiences by pilots, etc.)
- Detection of proposed traffic on the display
- Detection of problems
    - Conflict between two trajectories (violation of minimum separation standards)
    - A/C below minimum altitude
    - A/C entering a restricted airspace
    - A/C entering unsafe atmospheric region
    - A/C loses control
- Assessment of the impact of problem and proposed solution on the over-all traffic
- Assessment of the need for intervention
- Conflict resolution

    - Inter-sector coordination for a change of Entry conditions
    - Coordination with the TC: pass the problem
    - Tactical resolution of the problem by the PC when needed
        - Communication with pilot
        - Resolution by change of heading/speed/routing/flight level
        - Choice of optimal decision
- Insuring optimal status
    - Compliance with safety standards
    - Compliance of minimal change principle or optimal re-routing
    - Control of downstream delays
    - Detect change in traffic load
- Last minute intervention
    - Detection of imminent conflicts/system errors
        - Detection of traffic handled by TC on the display
    - Resolution of conflict
        - Fast situation assessment
        - Fast decision making
        - Fast and direct communication with pilot ?

In order to achieve these goals, the PC needs the following **information**:

- Current and future information on single A/C
    - Heading, trajectory, flight level, speed
    - Size, charge, etc.
    - Flight plan (destination, weigh points, etc.)
- Current and predicted weather information
- Runway status, restricted areas, separation/safety standards, etc.
- Status of the traffic handled by the TC; TC problems; TC workload
- On-board problems
- Status of traffic in surrounding sectors; problems near the boundaries
- Status of the system (automation/humans)

The responsibilities of the TC, on the other hand, can be defined as follows:

- Inter-sector coordination
    - Handoff (mostly exiting A/C)
    - Awareness and management of requests
- Coordination with PC
    - Ask for assistance for in-sector traffic
    - Handoff
    - Receive problems allocated by PC
    - Awareness and management of requests
- Coordination with pilot
    - Communicate advisory
    - Issue clearance for flight level change/take-off/landing
    - Communicate weather status
    - Receive pilot demands
        - Advisory
        - Weather report, Position
        - Clearance for flight level change/take-off/landing
- System updating
    - Modify trajectories where required
    - Other information? (runway, weather experiences by pilots, etc.)
- Detection of proposed traffic on the display
- Management of climb-descent situations
    - Assessment of impact on surrounding/over-all traffic
    - Assessment of safety and A/C capabilities
    - Assessment of efficiency (delays)
- Detection of problems
    - Conflict between two trajectories (violation of minimum separation standards)
    - A/C below minimum altitude
    - A/C entering a restricted airspace
    - A/C entering unsafe atmospheric region
    - A/C loses control, technical problems, abnormal situations

- Assessment of the impact of problem and/or proposed solution on the over-all traffic
- Assessment of the need for intervention
- Awareness of the TC's feelings of stress, fatigue, workload
- Conflict resolution
    - Coordination with the PC: ask for assistance
    - Tactical resolution of the problem
        - Communication with pilot
        - Resolution by change of heading/speed/routing/flight level
        - Choice of optimal decision
- Insuring optimal status
    - Compliance with safety standards
    - Compliance of minimal change principle or optimal re-routing
    - Control of downstream delays
    - Detect change in traffic load
- Last minute intervention
    - Detection of imminent conflicts/system errors
        - Detection of traffic handled by PC on the display
    - Resolution of conflict
        - Fast situation assessment
        - Fast decision making

In order to achieve these goals, the PC needs the following **information**:

- Current and future information on single A/C
    - Heading, trajectory, flight level, speed
    - Size, charge, etc.
    - Flight plan (destination, weigh points, etc.)
- Current and predicted weather information
- Runway status, restricted areas, separation/safety standards, etc.
- Status of the traffic handled by the PC; PC problems;
- On-board problems

- Status in down-stream sectors
- Status of the system (automation/humans), technical problems

3 Some Human Factors

Before the automation is specified and designed, a determination of the human factors to take into account in the system development process must be performed. In particular, when equipment failure occurs, safety relies on the skills of operators, and therefore the designers must make sure these critical failure-recovery aspects, which require an error-tolerant system, are not reduced by new procedures and technologies. The operators should be able to use these new procedures and technologies efficiently and integrate them successfully into their existing knowledge and experience. They should also have the information necessary to perform their tasks with no automated assistance when equipment failure occurs. The major human factors to consider are presented in the following sections.

3.1 Teamwork and Communication

Teamwork and communication are key issues as far as human factors are concerned. The air traffic controller is an integral part of two teams: one defined by the controller and the pilots of all the aircraft in his/her sector, and the other defined by all the controllers at a facility. Communication is vital to these team functions. Communication can often be analyzed from an information processing perspective, and much of it depends on both sender and receiver sharing the same mental model of the situation. Most automated aids are more suitable for individuals than for teams, and interaction between human and machine often takes the place of interactions between people.

Many believe that communication errors constitute the largest single category of errors in aviation. According to Billing [**6**], 73% of errors resulting in incident reports occur in transfer of information, 85% of which occur when the information is transmitted orally, because of the imprecision of the natural language [**13**], the speed at which some controllers speak, the language barriers, or simply the equipment. A solution to that problem in ATC was to use standardized terminology (ICAO), but controllers and pilots sometimes slip back to colloquial English. Moreover, communication errors can simply arise from people perceiving what they expect to hear (e.g. call-signs confusion) or having memory failures (e.g. controller forgetting to complete handoff, [**53**]). Automation (such as digital *datalink* systems) is thought to be the answer, but a potential problem is that it may eliminate

some valuable communication channels such as nonverbal cues or voice inflection, thereby reducing situation awareness and degrading efficiency of the decision making process. In addition, automation may not solve the problems related to *shared situation awareness* due to a discrepancy between the information given to the controller and to the pilot [**63**]. These same issues are raised with EUROCONTROL's SYSCO (System Supported Coordination), the new automated inter-sector coordination system.

3.2 Situation Awareness

A definition of situation awareness is given in [**17**] as "perception of the elements in the environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future." Situation awareness has a lot to do with the accuracy of one's mental model of events and objects in the real world. Both attention and memory are implicated in this process. Contrary to reports naming communication as the largest cause of errors, it was found in [**68**] that 60% of errors in the ATC system were caused by insufficient attention, non-recognition of problems, and judgment errors, thus leading to failures in monitoring altitude, heading, or flight path [**53**]. According to Redding [**58**], the second largest source of ATC errors after communication was the misidentification or misuse of radar data (37.6%), and 20% of errors in situation awareness result from a mismatch between the controllers expectations and what actually occurred. The mode confusion problem discussed in **Chapter 6** is a particular case of loss of situation awareness.

3.3 Perceived Automation Reliability

Another important issue that must be taken into account when introducing new automated features is that of *perceived* automation reliability (i.e. from the *operator*'s point of view, is automation doing what it is supposed to do?). This leads to situations of mistrust that could imply inefficiency or, more critically, to situations of overtrust and complacency. In the latter case, the capability of the controller to intervene in emergencies where automation fails are severely affected: detection capabilities can be significantly reduced, as well as situation awareness, especially if the system provides poor feedback about the ongoing state of the automated process. Finally, complacency can lead to skill loss since active participation in routine tasks aids memory and understanding. When computer assistance takes the form of automating routine manual tasks, the controller often finds it necessary to perform a new task to retrieve from the system information formerly obtained

incidentally. Without active participation in the control loop, maintaining knowledge of the system state to perform even residual functions (failure recovery) adequately can become tenuous.

3.4 Workload

Finally, one of the reasons why automation is being introduced into the ATC system is that the increase in air traffic density and complexity has led to substantial demands on mental workload of controllers, occasionally leading to incidents or accidents such as the 1991 accident in the Los Angeles Airport in which a departing commuter plane was placed on the runway in the path of a landing US Air 737. Very high workload can lower performance and set an upper limit on traffic-handling capacity. Conversely, very low workload may result in lack of concentration and reduced alertness, with subsequent implications for handling emergencies. Various performance aids in the form of computer assistance, designed to alleviate workload when the controller is very busy, may aggravate the problem of lack of concentration if they must also be used when the controller is lightly loaded, for then they reduce workload even further. In fact, Stager and Hameluck [**69**] found that 80% of the errors occurred in periods of average or below average workload. It is therefore not possible to draw conclusions about the relative likelihood of errors under varying workload conditions. In addition, Rodgers [**61**] hypothesized that different types of errors are associated with high and low workload. High workload errors would include computer entry, flight strip processing, or relief briefings. Low workload can lead to errors in the coordination between sectors and facilities.

As part of the first level of the Intent Specifications, this chapter presented the system goals, the operators' goals and responsibilities, and some human factors that should be considered in any human-centered, safety-driven system design. The users goals and the human factors will play an important role in the identification of the task allocation principles and of the different sets of requirements, as explained in the two following chapters.

Chapter 3

Task allocation: Human vs. Automation

In this chapter, we cover a very important part of the Preliminary Task Analysis introduced in **Chapter 2**, the Task Allocation. We first start by introducing the concept of Task Allocation and its importance in Air Traffic Control systems. We then provide some specific Task Allocation principles for the EUROCONTROL system, more precisely MTCD and the related tools.

1 Task Allocation: Why and How

The approach of compiling a listing of functions suitable for human beings and of functions suitable for machines dates from the 1951 famous report concerning human factors in air traffic control, by Fitts [**26**]. However, with this first classification, all tasks that a machine could possibly do were to be automated, regardless of the human factors involved (see **Table 1**). We call this kind of

Men are better at:
• Detecting small amounts of visual, auditory, or chemical energy
• Perceiving patterns of light or sound
• Improvising and using flexible procedures
• Storing information for long periods of time, and recalling appropriate parts
• Reasoning inductively
• Exercising judgment
Machines are better at:
• Responding quickly to control signals
• Applying great force smoothly and precisely
• Storing information briefly, erasing it completely
• Reasoning deductively

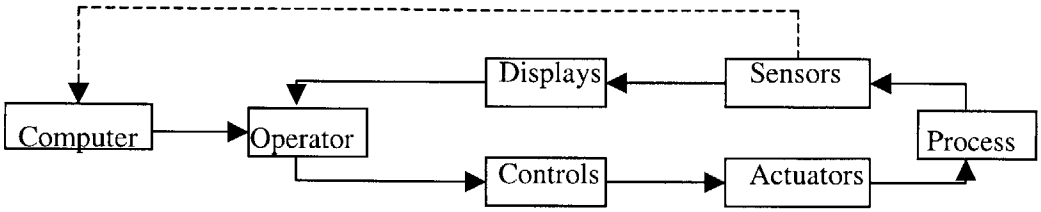
Table 1: The Fitts (1951) MABA-MABA List

allocation *comparison allocation*, which is a strategy where humans and machines are directly compared based on a particular sub-function. *Comparison allocation* is itself part of the *prescriptive methods*, that also include *leftover allocation* (a strategy that consists of automating what can be automated and allocating the remaining functions to the human), and *economic allocation* (a strategy where functions are divided between human and machine based on the maximization of some utility). Prescriptive methods have failed, because the word "best" in "best allocation" is itself hard to define, since the human operator is difficult to model and since the "system" implications of allocation are very complex. Present task allocation efforts try to avoid such a narrowness of view, as we shall see.

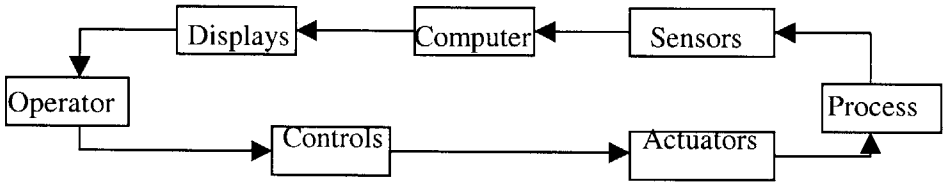
First, let us try to understand why automation is desirable in a system. Some processes are automated because it is dangerous or impossible for humans to perform the equivalent tasks, or because they are very challenging for the human operator. This is not the case in ATC, at least not for the moment. Sometimes functions are automated simply because the technology is there and inexpensive, even though it might provide little or no value to the human user. This must be avoided since it introduces unnecessary complexity and vulnerability to a system that could work perfectly without it. Finally, automated functions might sometimes not replace but may simply *aid* humans in doing things in otherwise difficult circumstances (tasks requiring much working memory, prediction, planning, etc.). This is the main and most desirable reason for use of automation in the ATC system, but the degree to which computers may *aid* the controller (i.e. the level at which desired intentions are stated by the human operator) makes a big difference, as we will see. Only control and cognition tasks will be considered in the remaining of this report, perception tasks being fairly well understood.

A 10-level scale of automation relating to decision and action selection can be identified. At the extreme of total manual operation, a particular function is continuously performed by the human operator, with no machine control. At the other extreme of total automation, all aspects of the function (including its monitoring) are delegated to a machine, so that only the end product and not its operation is made available to the human operator. In between these two extremes lie different degrees of participation in the function by the human and by the automation. Here is another classification proposed by Sheridan, in seven levels:

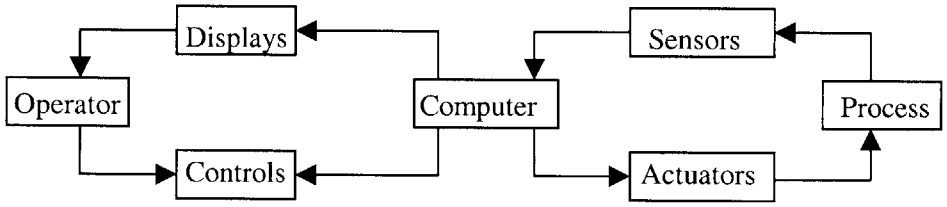
1. The automation offers no assistance: the human must do it all.
2. The automation suggests alternative ways to do the task.
3. The automation selects one way to do the task, and
4. ... executes that suggestion if the human approves, or



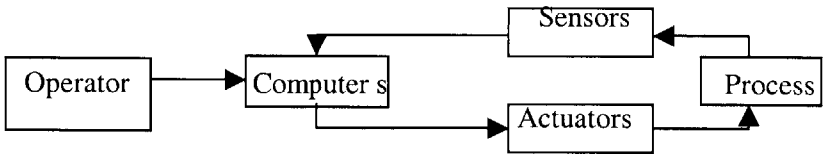
(a) Automation provides information and advice to controller, perhaps by reading sensors directly.



(b) Computer reads and interprets sensor data for operator.



(c) Computer interprets and displays data for operator and issues commands; operator makes varying levels of decisions.



(d) Computer assumes complete control of process with operator providing advice or high-level direction

Figure 5: Alternative uses of computers in control loops (Leveson, [42])

5. ... allows the human restricted time to veto before automatic execution, or
6. ... executes automatically, then informs the human only if asked
7. The automation selects, executes, and ignores the human.

An illustration of the relative position of automation vs. human operator in the control loop is also presented in **Figure 5**, taken from [**42**]. Note also that it has been suggested in some cases (like Free Flight, autonomous vehicles, UAVs) to remove humans all together from the decision systems- but in fact the suggestions were often simply about shifting responsibility from one human to the other, e.g. from the controller to the pilot . This issue will not be discussed further here.

Whatever the level of automation chosen, the locus of authority must be as unambiguous as possible in order to minimize opportunity for confusion between perceived and actual authority. As to how to determine what level of automation is most adequate for a given function, different factors may intervene and there is no single accepted strategy to determine this best mix of human and automation contributions. An example could be the concept of human-centered automation (Billings, 1996), i.e. an "automation designed to work cooperatively with human operators in the pursuit of stated common objectives". The idea, although very appealing, was criticized by some authors for not allowing a clear, satisfying allocation [**67**]. It is the purpose of this thesis to provide a well defined approach to *human-centered system design.*

2 Task Allocation and Air Traffic Control

The importance of modernization and automation in Air Traffic Control has been highlighted in **Chapter 1**. It was also mentioned that such automation could take the form of a straightforward assistance or replacement of some vulnerable human functions, but that some ATC automated tools might also replace some key perceptual and cognitive tasks. Given these major changes introduced to the system, it is vital to make sure safety (and maybe efficiency) is not being compromised by marginalizing the human controller's ability to effectively monitor the process, intervene as spot errors or failures in the software or environmental disturbances require, or assume manual control if the automation becomes untrustworthy. In particular, the question of what should (or can) be automated is raised.

Going back to **Figure 5**, the use of automation in most current Air Traffic Control systems would fit in the models (a) (the automation provides information and advice to controller, perhaps by reading

sensors directly) or (b) (computer reads and interprets sensor data for operator). The automation, indeed, gives well-defined information that the human uses, together with several other pieces of information, to "control" the airspace for which he/she is responsible. We could say that automation executes the low level tasks, such as computations, whereas the human executes the high level tasks, such as issuing the final advisory.

As people talk more and more about full automation, it is fundamental to understand why it is important to have a human as a supervisory controller, giving him or her the final authority. One important reason is that he/she can accept the responsibility, but in order to be responsible for decisions, the controller needs to have authority for those decisions. Moreover, a functional requirement for the air traffic control system being the adaptation to unanticipated situations, the human plays an essential role in the system: it is common for the human to successfully adapt to unanticipated conditions. In particular, experience and knowledge seem to be used in understanding how to adapt. On this subject, Billings comments in [**8**] that "...pilots and air traffic controllers are essential because they are able to make good decisions in difficult situations. We have not yet devised a computer that can cope with the variability inherent in the flight and air traffic environment".

3 Task Allocation in the EUROCONTROL System

EUROCONTROL has chosen to develop decision support tools for the controllers. Different tools will assist the controller for the different aspects of his/her tasks. Within this complex system, it is important to allocate functions based on sound principles. Such principles are developed using the results of the PHA, previous accidents and incidents, the analysis of the current system's efficiency and safety levels, the inputs of the actual end-users (controllers and pilots in this case), controller preferences and inputs, human factors considerations (including cognitive science and engineering; [**9**, **37**]), etc. An example of the controllers' cognitive model used by EUROCONTROL [**19**] is presented in **Figure 6**. This model was used in the referenced study to identify the main tasks performed and describe these tasks in terms of information required, actors and tools involved for both current procedures (in a broad context) and proposed future procedures. **Figure 6** shows the mental model (central part) as well as the identified tasks.

In addition to the inputs mentioned above, a high-level, human-centered hazard analysis can be very useful to determine the tasks where the human might need assistance, as well as those where the introduction of automation can contribute to new hazards. The task allocation and the preliminary

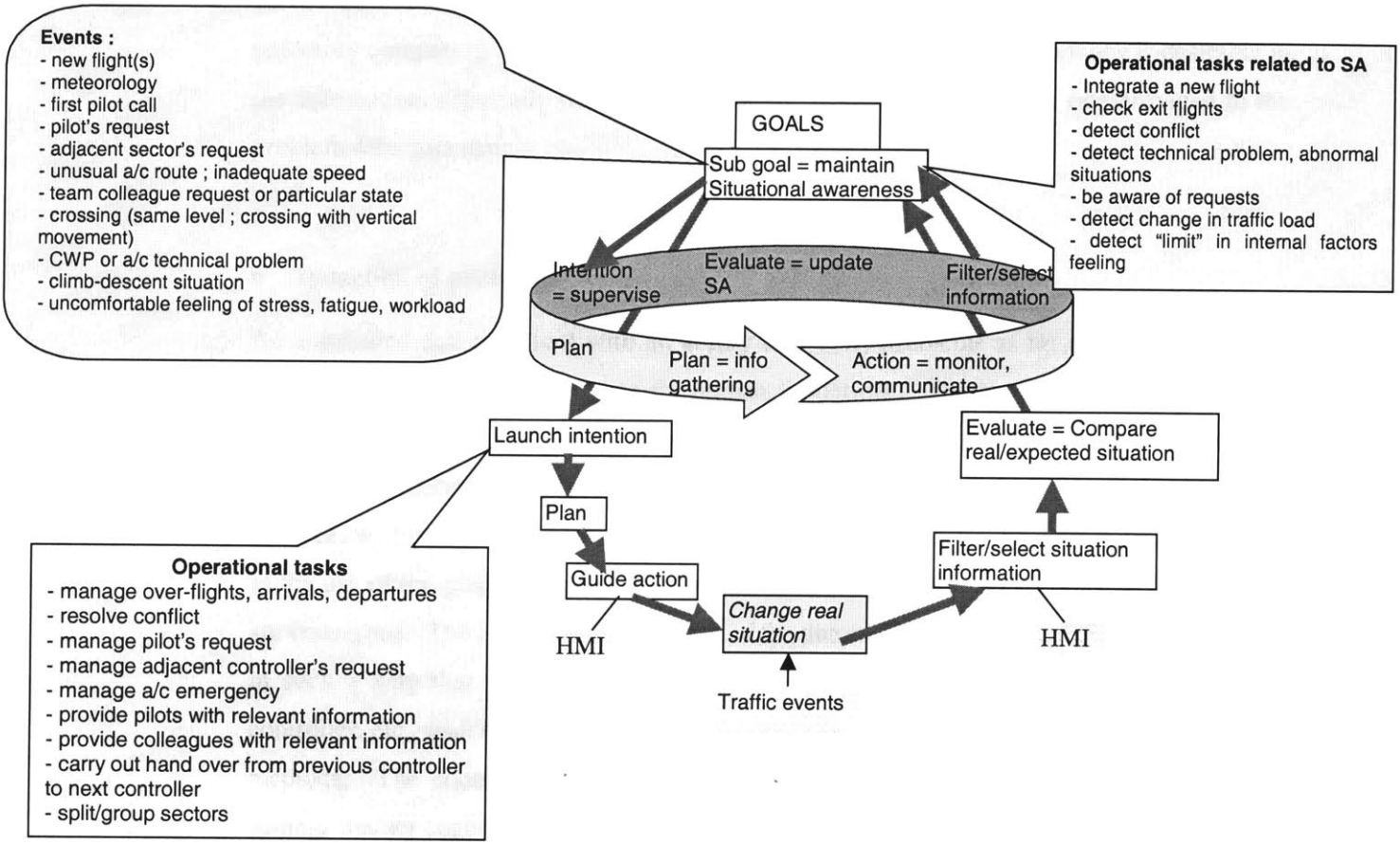


Figure 6: Air Traffic Controller's Cognitive Model, EUROCONTROL

hazard analysis should go hand in hand, in an iterative process (see **Appendix A** for more details on Preliminary Hazard Analysis). Several other human factors issues, such as those described in the next section, can also be considered during task allocation.

The following list is a high-level overview of the allocation principles that we believe should be respected for some of the high-level goals listed previously (*note*: with more information on the system, we would have given more details on some allocation issues specific to EUROCONTROL):

- **Detection of the proposed traffic**

  Both the human and the automation should be able to "see" (perceive) the traffic. The information should be accurate and up to date; appropriate feedback on the mode/state of the automation should be provided to the controller. The information should be displayed in such a way as to be easily and quickly accessible by the human and to match his/her working behavior, methods and decision-making. No additional perceptual or cognitive workload

should be introduced. The actual detection is performed through automated sensors and trajectory computers. The information is updated automatically. Redundancy is desirable in the automation, especially to make sure that the traffic sensed and the one provided to the problem detectors match.

- **Detection of problems**

An automated detection tool with an ability to predict problems as far as possible into the future is desirable, to compensate for human limitations. The automation shall notify the human in a clear and unambiguous way when a problem is detected. The information used should be accurate and up to date. Appropriate feedback on the mode/state of the automation should be provided to the controller. The human shall have final authority as far as the use of the prediction tool, the need for intervention and the criticality of the situation are concerned. The configuration/settings of the automation should be chosen by the human in such a way that the rate of false alarms vs. misses, the range of time given for the controller, etc., match the human's cognitive/workload capacity and his/her problem solving methods. The controller should be able to hide certain problems temporarily to better manage his/her cognitive workload, but the problem should be displayed again automatically after a certain amount of time. A last-minute automated conflict detection shall be provided, and the operator should only be able to turn it off after a well-thought, conscious decision is made. The automation shall be able to detect all the problems normally assigned to the controller: conflict between two trajectories (violation of minimum separation standards), A/C below minimum altitude, A/C entering a restricted airspace, A/C entering unsafe atmospheric region, A/C loses control, technical problems, abnormal situations.

- **Assessment of the impact of problem and/or proposed solution on the over-all traffic; Assessment of the need for intervention**

The human should have the final authority, but automation should be available to support situation awareness and compensate for limitations in human prediction. The information used should be accurate and up to date. Appropriate feedback on the mode/state of the automation should be provided to the controller. The automation should allow the human to experiment or test different solutions and see their future effects. The decision making can be supported by letting the automation assess the different solutions (safety + optimality) and

notify the controller of their relative quality. The experimentation with new solutions should not interfere with the achievement of the other goals-- including problem detection-- and the additional perceptual and cognitive workload should be minimal. The time of use of this function should not exceed a certain value so as to make sure the other traffic is not being neglected and no delays are being introduced in the system.

- **Awareness of the controller's feelings of stress, fatigue, workload**

This task should be performed mainly by the human operators. The automation can however give some assistance if a proper workload assessment technique is specified. In particular, the performance of the controller at his/her primary task could be assessed (e.g. by the number of conflicts handled per hour), but this measure is only valid in period of high workload, which is also the case where an automated workload assessment can be useful. In all cases, the PC, whose workload level is normally lower than the TC's, shall be able to monitor the TC's workload by observing his/her performance on the primary and secondary tasks. Appropriate feedback on the mode/state of the automation (if used) should be provided to the controller.

- **Conflict resolution**

The controller shall have the final authority, but automated decision-making support shall be available in the form of advisory tools. The automation should support the human in his/her assessment of the impact of the problem and/or the proposed solution on the over-all traffic. The automation should also suggest different solutions with an indication of their relative quality (safety + optimality), but it should not apply any decision without the human's approval. The best solution found by the automation should be highlighted, especially and most critically for last-minute intervention. This last-minute solution should be visible by both the TC and the PC for redundancy reasons. The resolution of a conflict should not interfere with the achievement of the other goals- including problem detection. The information used should be accurate and up to date. Appropriate feedback on the mode/state of the automation should be provided to the controller. The time of use of this function should not exceed a certain value so as to make sure the other traffic is not being neglected and no delays are introduced in the system-- a warning should be given to the controller after this lapse of time. The automation advisories should be clear and displayed in a way

consistent with the controller's decision-making behavior and his/her preferences; the choice could be given to the controller as far as the display method is concerned (lists vs. graphics, etc.); the display should add no additional perceptive or cognitive workload.

- **Ensuring optimal status**

Compliance with safety standards requires good problem-detection capabilities. Optimal re-routing or trajectory change requires the use of automation for a faster examination of all possible solutions (see "conflict resolution"). The automation should notify the human when better solutions exist, but this option should be easy to turn off to avoid additional workload and annoying cues in times of high workload. This functionality is more useful for the PC. The automation should be able to signal downstream delays to the controller periodically in order to avoid additional, human-induced delays. The automation should be able to signal the traffic load to the PC upon demand, especially the traffic handled by the TC. The information used should be accurate and up to date. Appropriate feedback on the mode/state of the automation should be provided to the controller.

- **Inter-sector coordination**

The inter-sector coordination should be automated as much as possible to avoid the introduction of delays or slips in the system. The automation should be able to give the controller an indication of the traffic/problems/delays in the surrounding sectors upon request. Special advisories should be provided for handoffs and for resolving boundary problems, taking into account the status of all the sectors involved. A warning should be given if the controller's decision is under-optimal or if it can lead to a short-term or a medium-term conflict. The handoff task should be automated, but the operation should be induced and controlled by the human: the human has the final authority and appropriate feedback on the mode/state of the automation should be provided to the controller. An automated communication channel should be available for requests, with a clear signal being given to the controller upon reception/emission of a message. The new communication methods should be easy to use and should not interfere with the execution of the other goals; the additional perceptive and cognitive workload should be minimal. Alternative (direct) communication channels should be available for emergencies and for informal communication in case the controller is not comfortable with the automation.

- **PC/TC coordination**

The PC/TC coordination should be automated as much as possible. The automation should be able to give the controller an indication of the traffic/problems/delays handled by the other controller. The handoff task should be automated, but the operation should be induced and controlled by the human operator: the human operator has the final authority and appropriate feedback on the mode/state of the automation should be provided to the controller. An automated communication channel should be available for requests, with a clear signal being given to the controller upon reception/emission of a message. The new communication methods should however be easy to use and should not interfere with the execution of the other goals; the additional perceptive and cognitive workload should be minimal. Alternative (direct) communication channels should be available for emergencies and for informal communication in case the controller is not comfortable with the automation. The PC, whose workload level is normally lower than the TC's, shall be able to monitor the TC's workload by observing his/her performance on the primary and secondary tasks. Emergency problems and solutions should be visible by both the TC and the PC for redundancy reasons. The automation should be able to signal the traffic load to the PC upon demand, especially the traffic handled by the TC.

- **Coordination with pilot**

The controller (TC)/pilot coordination should be automated when possible to reduce the occurrence of voice transmission errors and to limit the delays. The automation should be able to give the controller an indication of the traffic/technical problems handled by the pilot. Appropriate feedback on the mode/state of the automation should be provided to the controller. An automated communication channel should be available for requests and answers, with a clear signal being given to the two controllers and to the pilot upon reception/emission of a message by any of them. The new communication methods should be easy and fast to use and should not interfere with the execution of the other goals of both the pilot and the controller; the additional perceptive and cognitive workload should be minimal. Alternative communication channels should be available for emergencies and for informal communication in case the controller or the pilot are uncomfortable using the automated channel (because of the lack of the voice intonation feedback for instance). Both the PC and the TC should be able to communicate and coordinate with the pilot (in particular

to take the pilot's preferences into account), but the content of the communication as well as the resulting measures should be detectable by all parties upon request.

- **System updating**

The system should be updated automatically when possible for a better management of the controllers' workload and to reduce the occurrence of entry errors /slips. Feedback should be given to the controller when such an update occurs. The controller should also have the possibility to update the system. The manual update should be easy to execute, should not interfere with the other goals, and should add no workload to the controller. It is preferable that the update capabilities be mainly with the PC because his/her responsibilities are medium-term, giving him more flexibility to manage housekeeping tasks.

This chapter explained the need to perform a task allocation between the human and the automation, as well as between the different human controllers, when developing a human-centered, safe system. The task allocation principles obtained at the end of this phase of Level 1 in the Intent Specification will be used in writing requirements (**Chapter 4**) and in identifying the design principles and the human tasks in Level 2 (**Chapter 5**).

Chapter 4

Requirements

Using high-level operator-centered principles related to the different users of the system (the users here include the controllers, the airlines, the FAA, etc.), the system high-level requirements (including functionality, maintenance, and management), the operator requirements, the system interface requirements, and the operator training requirements are developed and later refined as the user tasks and the system design are refined. These principles are found in the preliminary hazard analysis, preliminary task analysis, human factors analyses, historical information, environment description, etc. Note that the automation requirements are derived from the operator task analysis, not vice versa (the more common design approach).

1 System Requirements

In the case of the EUROCONTROL tools, an important part of the system has already been built, so at this point we can only trace the task-allocation and human factors requirements to the software requirements. The following is a list (partial) of the automated tools that assist with attaining each of the high level goals described above:

- Detection of the proposed traffic: Radar sensors, HMI, FDPD, EDPD;
- Detection of problems: MTCD, MONA, Safety Nets, Arrival Manager, Departure Manager;
- Assessment [...]: MTCD, What-if Probe;
- Awareness of the controller's feelings of stress, fatigue, workload: none;
- Conflict resolution: What-if probe; conflict resolution aids will be added to the system in the future;

- Insuring optimal status: MTCD, MONA, Safety Nets, Arrival Manager, Departure Manager;
- Inter-sector coordination: SYSCO, phone;
- Intra-sector coordination: Direct, other;
- Coordination with pilot: radio, data link;
- System updating: supervision/archiving/recording;

Since the main focus of these intent specifications is MTCD, we will trace the task allocation principles to the MTCD environment assumptions and constraints, and the MTCD operational requirements: the requirements of the automated system should *at least* respect those principles. A complete list of the identified requirements can be found in [**27**]. The numbers used below are to be linked to the Level 1 requirements of the Intent Specifications document, and correspond to those used in [**27**]. They illustrate the tracing method used, but they are not intended to be complete.

Detection of problems

- **Principle**: The automation shall notify the human in a clear and unambiguous way when a problem is detected.
    - **1.HMI.01:** HMI shall handle all the interactions between the controllers and MTCD and between the MTCD supervisor and MTCD.
    - **1.45:** MTCD shall provide the HMI with conflict data for the display of conflicts after each conflict detection calculation. MTCD shall provide the HMI with an end-of-conflict notification for each conflict ended.

- **Principle**: The information used should be accurate and up to date. Appropriate feedback on the mode/state of the automation should be provided to the controller.
    - **1.28**: MTCD shall warn the controller and the MTCD supervisor of total losses, partial losses, and corruptions in the provision of conflict data.
    - **1.47**: On controller's request through the HMI, MTCD shall provide the HMI with feedback on the current configuration parameters values.

- **1.HMI.05**: HMI shall provide appropriate feedback on the state of the automation: either operational, configuration, or off state.

- **Principle**: An automated detection tool with an ability to predict problems as far as possible into the future is desirable, to compensate for the poor prediction capabilities of the humans.
  - **1.01:** MTCD shall detect within the area of operation, all aircraft conflicts in the aircraft prediction horizon. The default values of this prediction horizon are 0-20 minutes.

- **Principle**: The human shall have final authority as far as the use of the prediction tool, the need for intervention and the criticality of the situation are concerned.
  - **1.42:** MTCD shall be started/stopped by the HMI.
  - **L.**2 MTCD provides support, but the responsibility of separating aircraft remains, at all times with the controller.

- **Principle**: The configuration/settings of the automation should be chosen by the human in such a way that the rate of false alarms vs. misses, the range of time given for the controller, etc. match the human's cognitive/workload capacity and his/her problem solving methods.
  - **1.48**: MTCD shall be configurable by the MTCD supervisor during a configuration phase.
  - **1.05:** MTCD shall allow changes of the values of the prediction horizons during the configuration phase.

- **Principle**: The controller should be able to hide certain problems temporarily to better manage his/her cognitive workload, but the problem should be displayed again automatically after a certain amount of time.
  - **1.Cstr-HMI.04**: HMI must not allow optional displays when a conflict has been detected.

- **Principle**: A last-minute automated conflict detection shall be provided, and the operator should only be able to turn it off after a well-thought, conscious decision is made.
    - **Env-As-SafNet-01**: The safety Nets will provide the tactical and planning controllers with alerts for aircraft conflicts, minimum safe altitude and minimum area proximity warnings with a typical horizon of 0-2 minutes.
    - **Env-Cstr-SafNet-02**: MTCD must have no knowledge of the existence of the Safety Nets alert tool [...].

- **Principle**: The automation shall be able to detect all the problems normally assigned to the controller [...].
    - **1.01:** MTCD shall detect within the area of operation, all aircraft conflicts in the aircraft prediction horizon.
    - **1.02**: MTCD shall detect all special use airspace penetrations within the area of operation [...].
    - **1.03:** MTCD shall detect all descent below lowest usable flight level [...].
    - **1.04:** MTCD shall detect all nominal route overlaps [...].

More requirements, constraints and limitations can be found in [**27**].

2 Operator and Training Requirements

Assumptions, requirements, and constraints involving the users' behavior and training should then be defined using the results of the preliminary hazard analysis, task analysis and task allocation. This information will be used in the design of the human-computer interface, the system logic, operator tasks and procedures, operator documentation and training plans and programs. Since little data is available at this point as far as MTCD users' behavior is concerned, additional requirements are introduced. Note that these requirements will be further refined and clarified as the user tasks and system design are refined.

- **MIT-OP-R01**: While the ultimate responsibility for safety rests with the controller in charge, the need to use MTCD's planning and warning capabilities shall be emphasized in MTCD training.

- **MIT-OP-R02**: The PC shall plan traffic based on MTCD output, and where a problem persists shall notify its existence and nature to the TC.

- **MIT-OP-R03**: If incorrect or inconvenient behavior (e.g. high rate of false alarms) is observed, the controller shall turn MTCD off.

- **MIT-OP-R04**: MTCD shall be used as a complement to the normal controller conflict detection vigilance.

- **MIT-OP-R05**: Controllers using MTCD shall be rated for both Planning and Radar licenses.

    - **MIT-OP-R05.1**: Air traffic controller ability to read and analyze conflict data displayed on the screen shall be assessed in simulated environment

- **MIT-OP-R06**: Controllers shall be trained to use MTCD.

    - **MIT-OP-R06.1**: Controllers shall be trained on the new procedures in a simulated environment
    - **MIT-OP-R06.2**: Controllers shall be trained for a stripless environment
    - **MIT-OP-R06.3**: Controllers shall be trained for the new intra-sector handoff procedures
    - **MIT-OP-R06.4**: Controllers shall be trained for the new inter-sector handoff procedures

- **MIT-OP-R07**: Controllers shall provide update data to the system.

- **MIT-OP-R08**: The PC shall separate traffic strategically and assign/co-ordinate semi-tactical instructions where workload permits.

  Rationale: The PC is responsible for detecting medium-term entry/exit conflicts and solving them by formulating plans with the TC of his/her sector and the PC of the other concerned sector. The TC is then responsible for implementing the changes decided. Also, one the main goals of MTCD being a better management of the controller's workload, the PC helps the TC with his/her semi-tactical (short-term) tasks when possible.

We also identified some operator constraints:

- **MIT-OP-C01**: In the event of any unforeseen occurrence, like an emergency, the controllers must not use MTCD. (*Note*: the non-compliance or unpredictability of the traffic makes it almost impossible to use the decision support tools based on trajectory prediction)

    - **MIR-OP-R08**: Controller shall use Safety Nets conflict data rather than MTCD in case of conflicting displayed data or unforeseen occurrence, like an emergency.
      Rationale: Safety Nets detects conflict based on radar data only, and therefore does not depend on the intermediary tools such as the TP or MONA. In addition, Safety Nets detects conflicts on a much shorter time range (0-2min), providing a more accurate conflict detection.

The information derived from the operator requirements and from the training requirements will be used in the design of the human-computer interface, the system logic, operator tasks and procedures, operator documentation and training plans and programs.

3 System Interface Requirements

The system interface separates this system from the different elements of its environment. The human-centered requirements on the automated parts of the environment have been included in the previous section and in [**27**]. We will consider here the human-machine interface, across which all the interaction between the system and the users occurs.

The requirements of the interface should be developed using the preliminary hazard analysis, task analysis and task allocation, as well as a large spectrum of human factors. It is important to ensure that the information is easily and quickly accessible by the human user and that it is displayed in a way that matches his/her working behavior, methods and decision-making; no additional perceptual or cognitive workload should be introduced. For this purpose, the interface requirements should take into account different perceptual principles, mental model principles, principles based on attention, memory principles, etc. A particular attention should be given to alerting displays, labels, monitoring displays, displays/controls layouts, input devices, etc. More details can be found in [**73**].

The detailed study of EUROCONTROL's HMI will not be undertaken here, in particular because of a lack of the information needed for such a study. HMI requirements, however, should normally be included at this level, and a list of human factors should be developed in parallel and traced down to the latter requirements. It is interesting to note that the actual HMI will vary across the ECAC and that it has not been clearly defined yet since is not completely developed. This is the kind of things we want to avoid by using a *human-centered design.*

Even if we are not studying HMI in detail in this document, it is important to ensure that the necessary information and feedback are provided to the controller and that the interface does not introduce any additional cognitive workload. The Preliminary Hazard Analysis, Preliminary Task Analysis, lessons learned from former accidents and incidents, and controllers' preferences will be used again here to determine what information is needed and what part of this information is supposed to be provided by the automated tool. Also, it is important that the information essential for the controller to do his/her task be available in a secondary form in case the automation goes down and the controllers need to take over full control. The result of this analysis should then be traced to the requirements of both MTCD (what should MTCD be able to give to HMI?) and HMI (what should HMI be able to give to the controller?). As far as the workload is concerned, the human-centered requirements pertain to the amount of information given and the form in which this information is given, the level of detail/abstraction for instance. The concept of Ecological Interface Design [**71**] could be considered as a framework for the development of such interfaces.

Now what information does the PC need for conflict detection, for example? (We are not including PC-TC coordination, although availability of information on this interaction is essential.)

- Current and future information on single A/C in the sector
    - Heading, trajectory, flight level, speed;
    - Size, charge, etc;
    - Flight plan (destination, weigh points, etc.);
- Restricted areas, separation/safety standards, etc;
- Traffic coming to or going from the sector;
- Status of traffic in surrounding sectors; problems near the boundaries;
- Status of the system (automation/humans);

The HMI should show, independently of the status of the automation, the traffic in and around the sector as sensed by radars, the flight plans and the restricted areas. It should be able to give the controller upon demand the separation and safety standards, the status of traffic in surrounding sectors, the problems near the boundaries and the status of the system. It was confirmed in fact that controllers only consider the data needed to make decisions [**23**]. Position and altitude are considered as two key data. It is only when information on position and altitude are not sufficient for conflict detection that the controllers look for other sources of information [**23**]. The results of an experiment held by Mogford [**52**] with ATC trainees suggest, on the other hand, that "... certain situation awareness aircraft data are more critical than others... Although it might be expected that all aircraft information is critical for adequate air traffic controller situation awareness, this experiment demonstrates that some elements (e.g. aircraft altitude and heading) may play a key role, whereas others (e.g. speed, position and identifier) may not be as important as expected... There may be three kinds of data in ATC environment: a) that which must be remembered and updated; b) that which can be searched for when needed and forgotten; and c) that which can be ignored. Only the first type of data is retained in situation awareness" [**19**]. Obviously, there is no clear agreement on what information is most critical, but it is clear that some items play a more important role than others, and those items will be required to be present on the HMI. The following requirement on the HMI may then be deduced from the analyses presented above:

- **HMI-R01**: HMI shall provide an indication of the identification number, position, altitude and heading of the aircraft in a conflict. Identifications of speed, aircraft type and size, and trajectory shall be available upon request by the operator.

If we add to the previous features a conflict detection tool (MTCD or Safety Nets), the following information should be available to the controller for the tool to be useful:

- Future information on the A/C in the sector through trajectory prediction;
- Aircraft affected by the conflict (name, heading, trajectory, flight level, speed, Size, Flight plan) with a clear (representational) distinction between the different conflicts;
- Time to conflict and time to loss of separation, with a listing of conflicts by chronological order and a different representation for last minute emergencies (possibly aural);

Rationale: *It has been shown that the controllers deal with conflicts by chronological order, and seldom think about the severity levels (e.g. distance of aircraft when conflict occurs) as we might imagine.*

The minimal information needed by the controller shall always be displayed (at worst, through an iconified window). Additional information should be accessible quickly and simply, as requested by the controller. HMI shall display conflict detection data in a clear and unambiguous way. Graphical displays should be used when possible, and specific items of interest should be highlighted in color. On the other hand, during the phase of introduction of the tool, it should be possible to chose an interface similar to the one the controllers were familiar with in case they are not comfortable with the new one. In particular, the controllers should be able to hide the newly introduced windows/menus/lists, and physical flight strips should be available upon request. In all cases, the HMI must give a warning to the user if he/she attempts to open a TBD number of windows (to avoid clutter, high perceptual or cognitive workload). HMI shall display conflict detection data in a clear and unambiguous way. HMI shall concentrate conflict data into a limited number of windows. HMI shall clearly show what controller is in charge of the conflict (inter-sector and intra-sector).

This chapter completes the steps required in the first level of the Intent Specifications. This document presented the steps involving the human aspects of the system; the actual Intent Specifications for MTCD can be found in [**27**]. The following chapters will now cover the second (**Chapter 5**) and the third (**Chapter 6**) levels of the Intent Specifications.

Chapter 5

Task Analysis and System Interface Design

The second level of an intent specification contains the system design principles (such as the control laws or surveillance principles and formulae) upon which the designed design at the lower levels is based and through which the goals and constraints at the highest level are satisfied. Models and specifications at the second level may be subjected to scientific and system analyses to evaluate alternatives with respect to the higher-level goals and constraints. The information at the second level allows engineers to reason about the system in terms of the physical principles and laws upon which the design is based and to use the information derived from the hazard analysis in any tradeoff decisions.

Using the system requirements and design constraints as well as the other information that has been generated to this point, a system design (or alternative system designs) is generated and tasks are allocated to the system components (including the human operators) to satisfy the requirements, Task Allocation principles, and operational goals. Note that this process will involve several iterations as the results of analysis, experimentation, review, etc., become available.

More precisely, for our purposes, the results of the analyses made in Level 1 are traced down in Level 2 into an operator or user task specification and user interface design principles. The system design principles (basic principles and assumptions upon which the system design depends) are derived from the Level 1 requirements and the Level 2 user task specifications. These principles can be found in [**27**]. Simulations, experiments, engineering analyses, and other verification and validation (V&V) procedures are used. The findings are then used to refine and improve the automation and the HMI designs and the operators' tasks and procedures, eventually allowing the development of a complete user manual. As in Level 1, the operator input is an essential part of the process. It is also very important for the designers to go through a process of "enculturation" in order to acquire an instinctive understanding of the culture and recognition of the importance of the context [**32**].

1 Task Analysis for the Existing System

Because the automation is being introduced into an existing system, it is important to define the new tasks or procedures based on the old one to ensure a smooth and safe transition. We start by performing a task analysis on the existing system, determining the tasks or actions that will be affected by the new tool, and deduce some principles for the definition of the new tasks and procedures.

The task analysis of the old system can be performed using the formal procedures (ICAO procedures for ATC). However, it should be kept in mind that in complex systems like air traffic control, the users seldom follow the procedures to the word. This aspect of ATC, as well as the different task analysis methods commonly used in the literature, is presented in Level 3. We believe that in a human-centered design the operators should be implicated in the process as much as possible. Therefore, it is essential to combine both procedures, published literature, verbal comments from controllers, and observations (video and audio recordings). In this study, we were not able to apply these recommendations thoroughly, but we did consult an Air Traffic Controller (currently involved with the development of simulations at EUROCONTROL) as we performed the analyses. In addition, the expertise of the analyst and his/her degree of familiarity with the system can make a big difference; again, in our case, no such experts were available. It is also important to note that this process is iterative: as we collect data and perform verification and validation on our analysis, we will have to modify or complete our results. The task analysis presented here is partly inspired from the PUMA (Performance and Usability Modeling technique in ATM) method and toolset [**31**] and from EUROCONTROL's methodology for predicting performance and workload [**18**]. The logic behind the analysis is, however, different: we are not studying the workload at this point.

A simple, hierarchical task analysis was performed for the European Air Traffic Control system. For a given goal or responsibility, designated here by "*function*" (e.g. conflict detection), we identify the related "*tasks*" (e.g. gather info on traffic) and "*actions*" (e.g. read call-signs on the traffic display). A task or function can be divided into subtasks, but the action is the lower level in the task hierarchy. An action can be visual, auditory, cognitive, psychomotor, etc. A task analysis for the air traffic control system based on this classification of actions can be found in [**18**]. Tasks and actions can be performed in sequential or parallel order, they can be optional or compulsory, they can be interrupted or not, they can have initiation/termination conditions, etc. The MAD* approach takes all these characteristics into account while defining single tasks, then combines the tasks in trees [**66**].

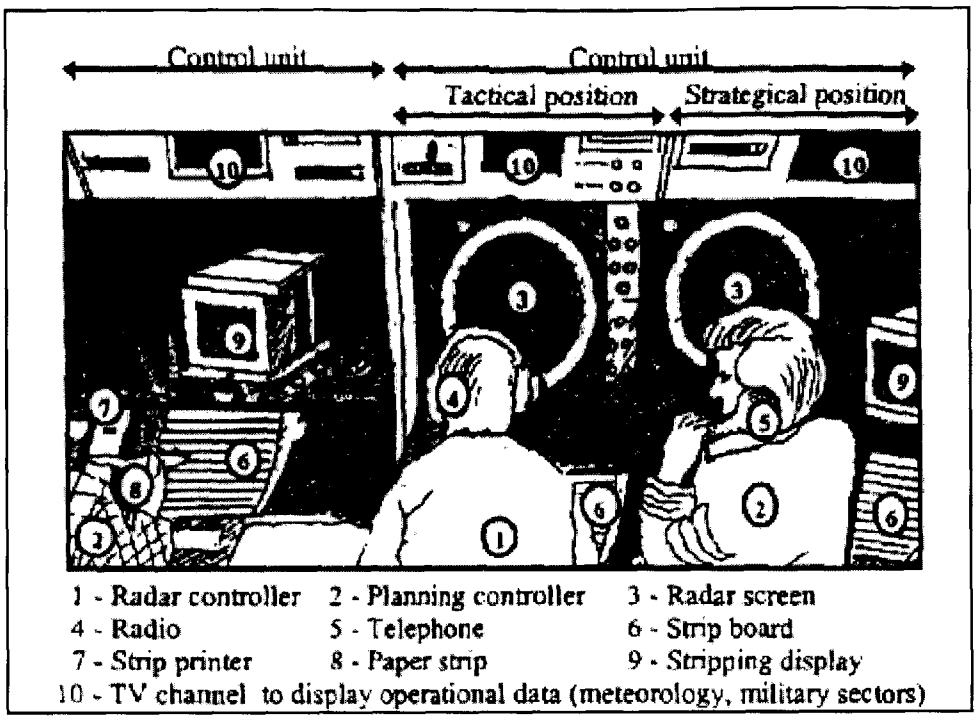


Figure 7: Example of control-unit organization [70]

Given the scope of the analysis performed here, we will not include the characteristics of the tasks mentioned above. Also, the analysis is partial, and is only meant to illustrate the kind of information needed at this level. We are concentrating on conflict detection and coordination/communication in en-route sectors, since these are the most relevant tasks for MTCD[1]. Finally, it is clear that the distinctions between task and action (or even function) are in part influenced by experience, since expertise can change the way we perceive the changes we can effect in the world. Expertise can also make tasks become simple actions. Therefore, the analysis would need to be performed with other assumptions, such as the skill-level of users in mind. The analysis presented here reflects only the normal behavior of an expert operator, no additional information being needed at this level.

[1] In Europe, the Upper Area Control Centers (UAC) and Area Control Centers (ACC) handle high and medium altitude flights that would be handled by the American en route centers

1.1 Overview

Each en-route sector (UAC or ACC)[2] is controlled by a Planning Controller who has 'Flight Strips' saying when aircraft intend to pass over specific points on their routes and at what level [**14**]. From these strips, the planner checks that aircraft will not pass over the same beacon at too close a time interval at the same level (the aircraft must normally be 5nmi apart). If they will, the planner usually decides to alter the level of one or the other (controllers make these decisions based on a vast repertoire of experience, and different controllers may make different decisions in the same context.) He/she also checks the acceptability of his/her decision with the surrounding sectors. A second controller, the Tactical/Executive Controller, watches the radar and talks to the aircraft, greeting them, changing their levels as the planner has planned, sometimes turning them, waiting until a problem is past and heading them back to their track, and passing them on to the next sector. In reality, the two controllers work very closely together, sharing tasks spontaneously, and communicating with gestures as much as words. The basic training for controllers takes three years, and it takes a further six months for a controller to reach a level of familiarity to be able to work a sector.

The Tactical Controller mainly uses a radar screen that depicts the position of all the aircraft in the sector. Each aircraft shown on the screen is accompanied by a data block (or label) that, under the present system, contains the call sign, altitude, ground speed, and weight class of the aircraft. Both the Tactical and the Planning controllers sometimes use a T.V. display giving information about weather (using METAR standards) or military airspaces for instance. The Planning controller, in addition to the radar and the TV displays, uses Flight Strips. Flight Strips (printed paper slips in plastic holders, a technology developed half a century ago) are printed approximately 30min before the aircraft is anticipated to arrive at the sector. They provide information about the kind and volume of aircraft as well as their arrival routes.

Using the information provided by the different displays and artifacts, the controllers can, in particular, detect conflicts, assess their criticality, formulate plans to solve the conflict, and assess this plan. For that purpose, the controller collects data about the different characteristics of the aircraft (speed, heading, altitude, weight, etc.). He/she also considers other factors such as current wind conditions (from TV display or by requesting a pilot's airspeed and comparing it with the

[2] The en-route systems use VOR radials to provide airways between 1,200 AGL up to FL450. Above FL450 aircraft fly direct between navaids or use another means of navigation (LORAN, GPS, celestial, etc.) [**74**]

ground speed shown on the data tag). The controller must also consider variations in piloting: control instructions can have different outcomes depending on how the pilot controls the aircraft. For example, controllers issue only the most routine instructions to foreign pilots lest they be misunderstood. Aircraft performance is another consideration.

1.2 Task Analysis for the TC (partial)

Following is the task analysis performed for the TC in the current system:

FUNCTION A: Monitoring/Detect possible conflicts

TASK A1: Gather information on the traffic

TASK A1.1: Read traffic display

ACTION A1.1a: Locate A/C on display

[Task A1.1.1: Read A/C label]

TASK A1.2: Read TV Channel

ACTION A1.2a: Locate restricted airspaces

TASK A1.2.1: Review weather conditions

ACTION A1.2.1a: Read meteorology information on display

ACTION A1.2.1b: Review mental model

ACTION A1a: Review (mentally) information obtained from pilots

ACTION A1b: Associate information from pilots, TD and TV

TASK A1.3: Determine if there are any possible conflicts

ACTION A1.3a: Compare Flight Levels of A/C

ACTION A1.3b: Compare times to reach a certain point of A/C

ACTION A1.3c: Compare directions of A/C

ACTION A1.3d: Compare Flight Level of A/C and minimum flight level

ACTION A1.3e: Compare location of A/C and of restricted airspaces

ACTION A1.3f: Compare location of A/C and weather conditions

ACTION A1.3f: Review mental model

TASK A2: Determine required action

ACTION A1a: Compare (mentally) different strategies

ACTION A2b: Choose the most adequate strategy

FUNCTION B: Assess of the need for intervention

TASK B1: Gather information on conflict

TASK B1.1: Read traffic display → A1.1

TASK B1.2: Read TV display → A1.2

ACTION B1a: Review (mentally) information obtained from pilots

ACTION B1b: Associate information from pilots, TD and TV

TASK B2: Determine if conflict is true

ACTION B2a: Review mental model

ACTION B2b: Decide if conflict is true.

TASK B3: Assess criticality of conflict

ACTION B3a: Review mental model

ACTION B3b: Assess criticality of conflict

TASK B4: Determine required action → A2

FUNCTION C: Conflict resolution/Formulate plan with PC

TASK C1: Gather information on conflict

TASK C1.1: Read traffic displays (PC and TC) → A1.1

TASK C1.2: Read TV display → A1.2

ACTION C1a: Review (mentally) information obtained from pilots

PC-TASK C1.3: Read flight strips (PC) → PC-A1

[TASK C1.4: Discuss traffic with PC]

ACTION C1.4a: Review appropriate phraseology

ACTION C1b: Associate information from pilots, TD, TV and PC

TASK C2: Formulate plans with PC

ACTION C2a: Propose line of action

ACTION C2b: Listen and assess line of action proposed by PC

TASK C3: Assess feasibility of options

TASK C3.1: Read traffic displays → A1.1

TASK C3.2: Read TV channel → A1.2

ACTION C3a: Review (mentally) information obtained from pilots

ACTION C3b: Associate information from pilots, TD and TV

TASK C3.3: Predict future behavior of traffic with proposed option

ACTION C3.3a: Review mental model

ACTION C3.3b: Assess possibility of future conflicts

[TASK C3.4: Discuss feasibility with PC]

ACTION C3.4a: Review appropriate phraseology

TASK C4: Determine required action

ACTION C4a: Compare different options

[TASK C4.1: Discuss comparison with PC]

ACTION C4.1a: Review appropriate phraseology

ACTION C4b: Choose best option

FUNCTION D: Conflict resolution/Formulate plan by self

TASK D1: Gather information on conflict

TASK D1.1: Read traffic display → A1.1

TASK D1.2: Read TV display → A1.2

ACTION D1a: Review information obtained from pilots

ACTION D1b: Associate information from pilots, TD and TV

TASK D2: Formulate plans

TASK D3: Assess feasibility of options

TASK D3.1: Read traffic displays → A1.1

TASK D3.2: Read TV Display → A1.2

ACTION D3a: Review (mentally) information obtained from pilots

ACTION D3b: Associate information from pilots, TD and TV

TASK D3.3: Predict future behavior of traffic with proposed option

ACTION D3.3a: Review mental model

ACTION D3.3b: Assess possibility of future conflicts

TASK D4: Determine required action

ACTION D4a: Compare different options

ACTION D4b: Choose best option

FUNCTION E: Assess implications of changing A/C parameter

TASK E1: Assess implications of changing FL

TASK E1.1: Assess degree of clutter

ACTION E1.1a: Look at traffic display

ACTION E1.1b: Assess degree of clutter

[TASK E1.2: Determine requirements to housekeeping]

TASK E1.3: Gather information on traffic

TASK E1.3.1: Read traffic display → A1.1

TASK E1.3.2: Read TV display → A1.2

ACTION E1.3a: Review (mentally) information obtained from pilots

ACTION E1.3b: Associate information from pilots, TV and TD

ACTION E1.3b: Detect other A/c at requested (or suggested) and interim FLs

ACTION E1.3c: Review mental model

ACTION E1.3d: Assess possibility of future conflicts

TASK E1.4: Determine required action

1.3 Task Analysis for the PC (partial)

Following is the task analysis performed for the PC in the current system:

FUNCTION A: Monitoring/Detect possible conflicts

TASK A1: Gather information on the traffic

TASK A1.1: Read Flight Strips/Stripping display

ACTION A1.1a: Read A/C call-sign

ACTION A1.1b: Read time to waypoint

ACTION A1.1c: Flight Level on waypoint

TASK A1.2: Read traffic display

ACTION A1.2a: Locate A/C on display

[TASK A1.2.1: Read A/C label]

TASK A1.3: Read TV Channel

ACTION A1.3a: Locate restricted airspaces

TASK A1.3.1: Review weather conditions

ACTION A1.3.1a: Read meteorology information on display

ACTION A1.3.1b: Review mental model

ACTION A1a: Associate information from strips, traffic display, and TV display

TASK A1.4: Determine if there are any possible conflicts

ACTION A1.4a: Compare Flight Levels of A/C

ACTION A1.4b: Compare times to reach a certain point of A/C

ACTION A1.4c: Compare directions of A/C

ACTION A1.4d: Compare Flight Level of A/C and minimum flight level

ACTION A1.4e: Compare location of A/C and of restricted airspaces

ACTION A1.4f: Compare location of A/C and weather conditions

ACTION A1.4f: Review mental model

TASK A2: Determine required action

ACTION A1a: Compare (mentally) different strategies

ACTION A2b: Choose the most adequate strategy

FUNCTION B: Assess of the need for intervention → TC-B

FUNCTION C: Conflict resolution/Formulate plan with TC → TC-C

FUNCTION D: Conflict resolution/Formulate plan with PC in next sector

ACTION Da: Activate phone/intercom; Call sector

TASK D1: Gather information on conflict

TASK D1.1: Read traffic displays → A1.2

TASK D1.2: Read TV display → A1.3

TASK D1.3: Read flight strips → A1.1

[TASK D1.4: Discuss conflict with second PC]

ACTION D1.4a: Review appropriate phraseology

ACTION D1a: Associate information from TD, TV, strips and second PC

TASK D2: Formulate plans with second PC

ACTION D2a: Propose line of action

ACTION D2b: Listen and assess line of action proposed by second PC

TASK D3: Assess feasibility of options

TASK D3.1: Read flight strips → A1.1

TASK D3.2: Read traffic displays → A1.2

TASK D3.3: Read TV display → A1.3

ACTION D3a: Associate information from TD, TV, strips and second PC

TASK D3.4: Predict future behavior of traffic with proposed option

ACTION D3.4a: Review mental model

ACTION D3.4b: Assess possibility of future conflicts

[TASK D3.5: Discuss feasibility with second PC]

ACTION D3.5a: Review appropriate phraseology

TASK D4: Determine required action

ACTION D4a: Compare different options

[TASK D4.1: Discuss comparison with PC]

ACTION D4.1a: Review appropriate phraseology

ACTION D4c: Choose best option

2 Identifying Tasks Affected by the New System

Once a task analysis has been performed on the existing system, one proceeds to identifying those functions, tasks or actions that will be affected by the introduction of the new system.

With the introduction of the new EUROCONTROL system, most of the tasks listed above will be affected. Since our system includes MTCD, SYSCO, and the conflict detection/communication functionalities of HMI (as a blackbox), we can say, based on the goals and requirements of these systems, that the performance of the following tasks will be affected:

- For the TC: A1 (includes A1.1, A1.2, A1b, A1.3); B1 (includes B1.1, B1.2, B1b), B3; C1 (includes C1.1, C1.2, C1.4, C1b), C3 (includes C3.1, C3.2, C3b, C3.3, C3.4); D1 (includes D1.1, D1.2, D1b), D3 (includes D3.1, D3.2, D3b, D3.3); E1 (includes E1.1, E1.2, E1.3,);
- For the PC: A1 (includes A1.1, A1.2, A1a, A1.3, A1.4); B1 (includes B1.1, B1.2, B1b), B3; C1 (includes C1.1, C1.2, C1.3, C1.4, C1b), C3 (includes C3.1, C3.2, C3b, C3.3, C3.4); Da, D1 (includes D1.1, D1.2, D1.3, D1.4, D1a), D3 (includes D3.1, D3.2, D3.3, D3a, D3.4, D3.5)

Obviously, most of the tasks will have to be redefined with the introduction of the new tools.

3 New Task Definition: Simulations and Experiments

At this point, it is essential to carefully define the principles underlying the future task definition using the previous task analysis, the automation's requirements and design principles, human factors considerations (like situation awareness, workload, communication, etc.), hazard analysis, and an understanding of the schemes of human errors. This definition of the tasks should be performed in parallel and iteratively with the definition of the human interface design principles, the definition of the automation design principles, and the refinement of the hazard analysis.

In order to identify a set of acceptable human task principles and system and HMI design principles, a series of simulations, experiments, engineering analyses, and other verification and validation

(V&V) procedures is needed. This observation implies a strong involvement of the operators as well as the engineers. The simulations allow a parallel definition of the user tasks, the interface design and the tool's design. For this purpose, an incremental development, starting from the current system, is preferred in order to explore larger parts of the design space and avoid introducing artificial design constraints (e.g. choosing to assess the criticality of a conflict based on time to loss of separation vs. distance between aircraft upon loss of separation, choosing the actions needed from the PC to initiate a System Supported Coordination with another PC, etc.). Using a baseline as close as possible to the current system and adding or changing features progressively is very important. This process is currently used for the HMI design at EUROCONTROL, though the simulations are not used to define the design principles for the tool itself. The experimental approach at the EUROCONTROL Experimental Centre (Bretigny, France) is an "Incremental Evaluation and Development Approach", in which prototypes are evaluated then improved. References [**19-22**] give some descriptions of such simulations. Obviously, this assumes that several design decisions have been made prior to the simulations and experiments. Ideally, however, the simulation would start earlier to simply assess the different design options individually. This process is nonetheless very costly, and in some cases a very simplified simulation or a survey of the end users' opinions is preferred.

This problem can be solved by using a blackbox model of the automation as a prototype that can be run on actual HMIs. Our blackbox models, presented in **Chapter 6**, are indeed all formal and executable, and do not need any physical implementation. Those models are constructed in the third level of our methodology, but because Intent Specifications abstract on intent rather than chronology, they are relevant in Level 2 as well. Thus, one would conceive a preliminary design for the automation and the HMI, and a set of controller tasks, then build the appropriate blackbox model of the automation and execute it with the HMI. The results of the simulation are then used to review the automation and the HMI design, and the controller's tasks and procedures.

The different designs/task descriptions and simulations/experiments should be based on the principles and requirements identified in Level 1 and on a large spectrum of human factors/cognitive engineering principles, such as those found in the human factors databases (references and standards as used by human factors professionals worldwide, e.g. [**55**, **64**, **12**]). The design development and evaluation should also go hand in hand with the Hazard Analysis, the fault tree being revisited to

check for new hazards that can rise from the proposed designs. The actual goals of the simulations and experiments may be defined as follows for each configuration of the system[3] [**39**]:

- Determine the adequacy of controllers' workload and performance,
- Understand the user-acceptance of a new configuration,
- Investigate error potential, and
- Evaluate the usability of the system.

In order to achieve these goals, several different analyses can be undertaken. These analyses may include:

- Workload analysis, e.g. the Instantaneous Self Assessment (ISA) technique, developed to assess mental workload in real time; the PUMA method and toolset [**31**] or EUROCONTROL's methodology for predicting performance and workload [**18**]. Other workload assessment techniques, such as the NASA-Task Load index [**29**], the Subjective Workload Assessment Technique [**60**], or some secondary task techniques [**50**], can also be used. Human factors experts should be responsible for selecting the most adequate technique for the considered simulation or experiment.
- Performance assessment.
- Subjective questionnaires, interviews, or debriefings. The content of these forms/interviews should be determined very carefully to reflect the desirable properties of the system. An example of a questionnaire used for one of EUROCONTROL's simulations at the EEC can be found in [**19**].
- Real-time tests to assess situation awareness and automation trust or mistrust.

Note that the experiments and simulations described in this section were not performed for MTCD at the Software Engineering Research Laboratory because of a lack of appropriate facilities and specialized personnel. The results of the EUROCONTROL simulations and experiments, as described in [**19-22**], will be used in the remaining of this document.

[3] A configuration includes a set of automation and HMI design principles and a set of human tasks.

4 HMI Design Principles

As mentioned previously, several simulations and experiments have been undertaken at the EUROCONTROL Experimental Centre to define MTCD's Human-Machine Interface. The principles used throughout the design and evaluation process were the following:

- The intention is to design a graphical interface without paper- or electronic- strips.
- The controller interaction with the system functionality is mainly envisaged through the radar labels.
- HMI displays permanently only the minimum information needed by the controller.
- The controller should be able to access additional information simply and quickly.
- Color should be used to highlight items of specific interest.
- The data display should not be split over too many devices.
- HMI should include graphical displays coupled with filtering of the information.

Obviously these principles are too high level to guide the design process, not only for the physical design of the interface but also for the informational content. EUROCONTROL used an incremental approach, developing and evaluating three baselines and then adding the specific functionalities. References [**19-22**] provide more detailed descriptions of the HMI baselines and functionalities used, as well as on the operator tasks evaluated.

Figure 8 below, taken from [**24**], shows one of the designs evaluated for the MTCD HMI. Note that at this level in the analysis, we are not concerned with the specifics of the physical representation (such as colors or dimensions), but more with the concepts and the informational content. The figure shows that three displays on the HMI are dedicated to MTCD: the *Dynamic Flight Leg* (DFL), *Vertical Alert Window* (VAW), and the *Potential Problem Display* (PPD), also referred to by CRT (*Conflict and Risk Display*). The DFL provides a graphical display of the selected flight's currently planned path through the sector and an indication of conflicting flights along the planned route. The VAW allows the user to view the vertical profile of a single flight through the sector, as well as other aircraft classified as risks or conflicts. The PPD allows the visualization of the conflicts predicted by MTCD for the sector through text and graphics, and displays all conflicts detected together with their associated risk. The VAW and the PPT can be iconified but not closed. The background radar display window is referred to by *Air Situation Window* (ASW).

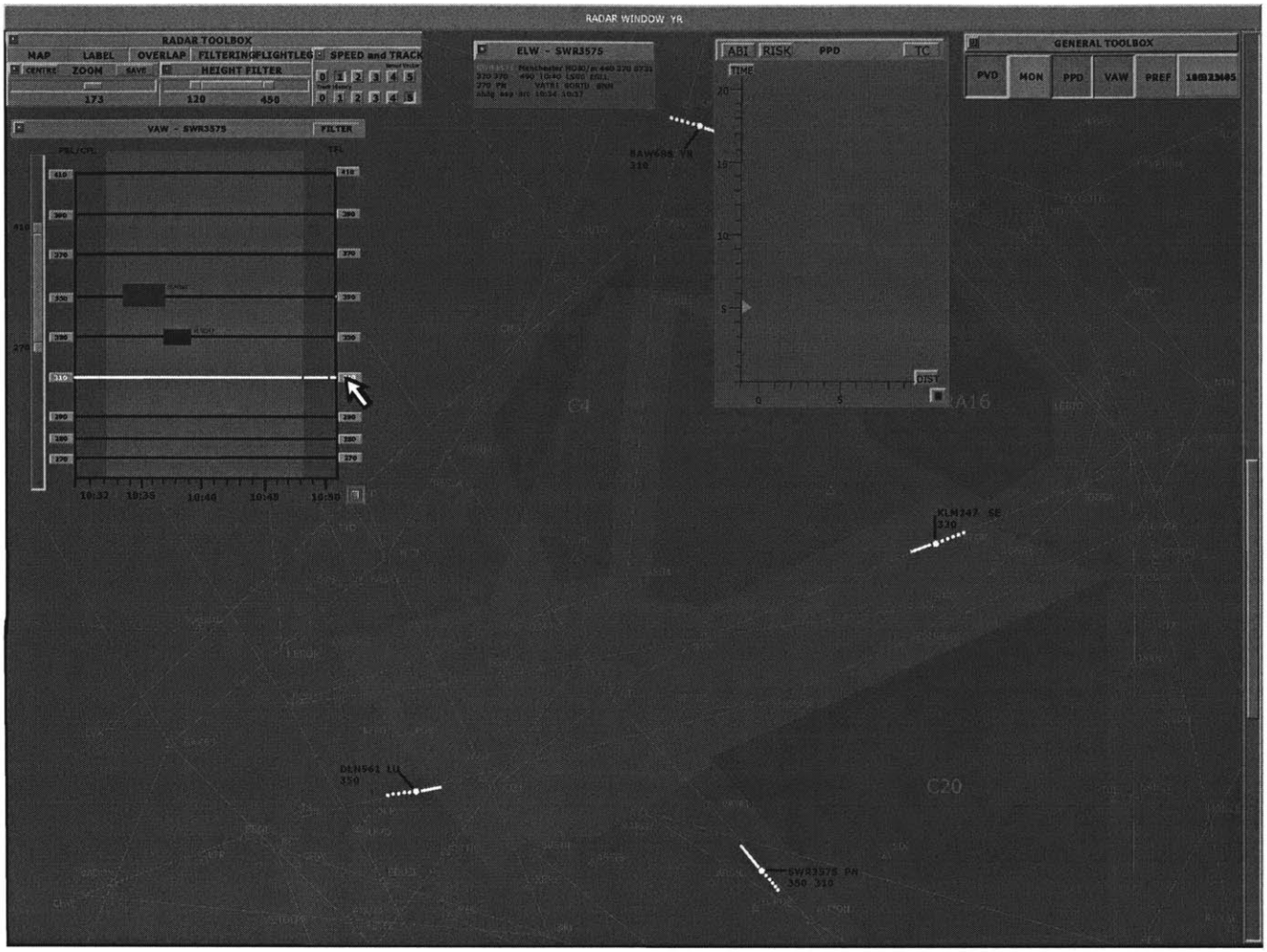


Figure 8: EUROCONTROL proposed Human-Machine Interface

It is not the purpose of this section to analyze the interface proposed by EUROCONTROL, but rather to illustrate what can be included at this level concerning the HMI. The experiments, simulations, and analyses required at this level necessitate the design of an HMI prototype. Ideally, these analyses shouldn't test the physical interface itself, but rather the design concepts and functionalities (e.g. using the DFL, VAW, and PPT for MTCD rather than other display options that could be considered as well). It is clear however that workload, performance, and operator trust/mistrust are closely dependent on the physical HMI (colors, location of windows, relative sizes, readability, etc.). It is therefore important to keep in mind that one can work on more than one level of the Intent Specification at a time (in this case levels 2, 4 and 5), in an iterative process. This has already been noted when we suggested the use of the Level 3 state machine as a prototype of the automation for the Level 2 analyses.

At the end of this level, we should have safe, functional, and usable design principles for the automation, the human-machine interface and the operator tasks. These three components are analyzed together, in a systemic approach. The next level of the Intent Specification, Level 3, will include a more formal analysis of these components. The actual physical layout of the human-machine interface would normally be included in Levels 4 and 5, but these levels are not covered in the present document.

Chapter 6

User Model and Blackbox Analyses

The third level contains formal, blackbox behavior models of the system components, including the operators, the interface or communication between the components, and the basic human-machine interface design. These models are executable and formally analyzable. The information and languages used at this third level enhance reasoning about the logical design of the system as a whole (the system architecture) and the interactions between components as well as the functional states without being distracted by implementation issues.

The next step in the process involves validating the system design and requirements and performing any trade studies that may be required to select from among a set of design alternatives. This validation is accomplished using a formal specification language called SpecTRM-RL [**46**]. Using SpecTRM-RL, designers build formal blackbox models of the required component behavior and operator tasks. An executable task modeling language has also been defined that is based on the same underlying formal model as that used for the automation modeling (see description below).

In this paper, we focus on the operator task analysis and show how the operator tasks are modeled using our state-machine language. A draft of the automation state-machine model can be found in [**27**]. We also show how this model can be formally analyzed and executed with the specification of the automation behavior simultaneously. Using a combination of simulation and formal analysis, the combined operator task model and blackbox automation behavior models can be evaluated for potential problems, such as task overload and automation design features that could lead to operator mode confusion [**62**].

The analysis performed at this stage of system design and development is not meant to replace standard simulator analysis, but to augment it. The formal analysis and automated execution of

specifications focus on safety, rather than usability. Formal analysis is indeed better able to handle safety while usability is better tested in simulators. The analysis can also assist in designing the simulator scenarios needed to evaluate safety in the more standard simulator testing. Because our models are executable, they can themselves be used in the simulators or used to automatically generate code for the simulators, as mentioned in the previous chapter. Simulators for the environment will, however, still need to be handwritten.

1 Why Build Task and User Models

A basic tenet of the linear control theory is that every controller contains a model of the general behavior of the controlled process. This model is updated and kept consistent with the actual system state through various forms of feedback from the system to the controller. In addition, the human controller has a model (i.e. some understanding) of the blackbox automation behavior. When these models diverge from the actual state of the controlled process or when the tasks of the controller are inconsistent with his understanding of the automation or the process, erroneous control commands can lead to an accident [**42**]. The situation becomes more complicated when there are multiple controllers because the models of the various controllers must also be kept consistent. Note that the different models can diverge because they were incorrect or incomplete to begin with or because they are improperly updated due to incorrect feedback about the state of the modeled system. Explicitly specifying and validating these models during system design allows, however, engineers to identify and eliminate error-prone features of automation and interfaces and to assist in task analysis and development of operator training and reference manuals.

2 Limitations of the Models

Air traffic control is often considered a complex task in terms of cognition. According to Woods [**75**], its main aspects fall into three categories: environmental characteristics, characteristics of the operators, and those of the interfaces. The complexity due to the characteristics of the control task is generally represented by dynamism, indetermination, and the large amount of data to be processed at any time. These characteristics induce data that is inaccurate, either because of its fuzziness (if the controller or automation has access only to an estimate of its value) or because of its uncertainty (if its value is likely, but needs confirmation). Other difficulties emerge also from the multiple sources of data (pilots, interfaces, teammates, procedure manuals, etc.) and from the perceptive and cognitive limitations of the human controller [**10**].

In fact controllers in all types of air traffic control facilities deal with the complexity of the system by developing strategic plans for traffic flow, monitoring plans with visual inputs to update their "big picture" of the traffic flow, and communicating heavily with pilots and other controllers to ensure continued safety and efficiency-- which can make control rooms sound sometimes like trading floors [**40**]. They appear to trade off their cognitive workload and their tactical goals, keeping their risk-taking constant to perform a task at an acceptable cognitive cost [**1**], and adapting their task-management to the context. It is explained in [**41**] that controllers manage to treat complex situations although they logically lack information to do so by acquiring, through experience, knowledge about the customary evolutions of the airspace, and using this knowledge to anticipate future evolutions.

The controller will usually classify the problem simply and initiate the appropriate procedure (set of clear rules or actions) to resolve it: he is using *rule-based behavior*, where, as Rasmussen says, "the rules can be reported by the person, although the cues releasing a rule may be difficult to describe" [**56**]. In some unusual cases, however, (like an emergency that the controller cannot resolve using a well-defined procedure), the human control might rather be ill-defined: the controller here uses *knowledge-based behavior*. Among the contributing factors to this unstructured response is the possible lack of trust in the automation (due to previous false alarms for instance), a mismatch between the automation and the pilot's current expectancy, or just the unavailability of the appropriate structured response (procedure) in the controller's short or long term memory.

This shows that simply examining the tasks and the cognitive, perceptual and motor demands on the controllers with respect to the normative tasks can be restrictive. Aspects of non-determinism in the controller's behavior are already included in the models described below, through an expansion of the baseline state machine model (SpecTRM-RL) that still allows performing math analyses like consistency and completeness checks, forward and backward simulations or deviation analysis. What is needed is a way to capture the *knowledge-based* behavior of the controllers, without sacrificing the rigor of the formal methods. Attempts have been made to integrate knowledge into formal task analysis, but the models were often too formal for the purposes of our study.

3 Task Analysis and User Model: Background

Brown and Leveson proposed an executable task modeling language based on an underlying formal model using state machines [**11**]. The components of this modeling language are shown in **Figure 9.**

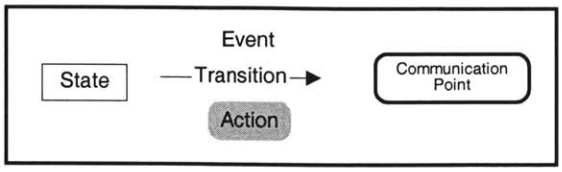


Figure 9: Elements of the Brown and Leveson task analysis model

The steps required to complete a task are represented by *states* (square box). A *transition* (arrow from one state to the next) is defined as the process of changing from one state to the next. Conditions that trigger transitions are called *events* (text above the transition), and an *action* (text with gray shade beneath the transition) describes a result or output associated with the transition. A *communication point* links different models together. It is represented by a round box with an arrow attached to it indicating the direction of the communication. The *destination* of the communication point is indicated at the end of the arrow, and the *means of communication* is noted under the round box. In our case, the human task models communicate their outputs (or actions) via communication points to the system model and to each other. In this way we can model human-computer interaction. The models are limited to the *expected* controller behavior as defined in the operational procedures. Deviation analyses are then performed to account for other controller behaviors. In the model, the normative actions are seen on the main horizontal axis of each controller's task model, whereas non-normative behaviors diverge from that main axis.

Perhaps the closest work to ours is that of Degani and that of Fields, Wright, and Harrison. Degani's work is also based on state-machine models, Statecharts, with a task-modeling framework known as OFAN. These models were found to be difficult to apply on complex systems and inadequate for our goals. Fields, Wright, and Harrison [**25**] use CSP models to attempt to achieve similar purposes, but this language is very formal and probably not usable without extensive training in discrete mathematics. In addition, the use of state-machine models seems most natural and promising for modeling real-time control.

Javaux [**33**] and Javaux and Polson [**34**] take a more psychological approach. They use a finite state machine based on Anderson's ACT-R [**2**] to describe a cognitive mental model (called a *spreading activation network*) and identify potential instances of mode confusion. However, these models do not scale up to the levels of complexity of modern control systems. In addition, we try not to use models of human cognition or human mental models because such models imply some assumptions as to the human cognition. These assumptions could indeed be wrong, given the complexity of the poorly understood human cognition, and therefore the models would have weak bases. Instead, we

try to model the blackbox behavior of the automation that the user expects and depends upon and the required steps needed to complete a given task.

There have been numerous models for human task analysis developed to identify the knowledge and steps required to perform each human task. In general, however, these models do not include models of the automation or the other components of the system. Examples of such models include the Hierarchical Task Analysis [**3**], the Goals, Operators, Methods, and Selection Rules (GOMS) framework [**38**], the Task Analysis for Knowledge Descriptions (TAKD), the Task Knowledge Structures (TKS), etc. Some executable task models have also been developed and used for modeling human-computer interaction, such as the Soar/TAQL used by Yost [**77**] or the Programmable User Model (PUM) used by Young *et al* [**76**]. Finally, the Operator Function Model (OFM) describes the required operator behavior by representing the operator functions, sub-functions, and information needed by the operator for each activity [**51**].

4 User and Task Model for MTCD

A user model and a task model were constructed for MTCD using the results of several simulations and experiment performed at the EUROCONTROL Experimental Centre and direct inputs from an Air Traffic Controller. The models are shown in **Figure 10**. They reflect not only the role of MTCD in the conflict detection process, but also the working method that the controllers will adopt when using MTCD. It is clear, for instance, that communication will still be the most important factor in the controller's tasks after the introduction of MTCD.

Four sub-models are included in this model: the Automation Expected Behavior, the Current Planning Controller, the Current Tactical Controller and the Next Planning Controller. We could have included the Next Tactical Controller or the Previous Planning/Tactical Controller(s) as well, but we chose to take the Current Planning Controller as the basis of the analysis and to only include those systems components (human or automation) that he/she interacts with.

The model is surprisingly simple, and includes very few human-automation interaction instances, as compared to the complex Flight Management System (FMS) model described in [**62**], to the helicopter hydraulic leak detection system model [**4**] or simply to the hand-off procedure model analyzed by Brown and Leveson in [**11**], as the latter included a model of the pilot tasks, which are particularly complex. The reason for that is that MTCD only provides decision support for the controller, and has no direct control on the process. In addition, MTCD is simply displaying

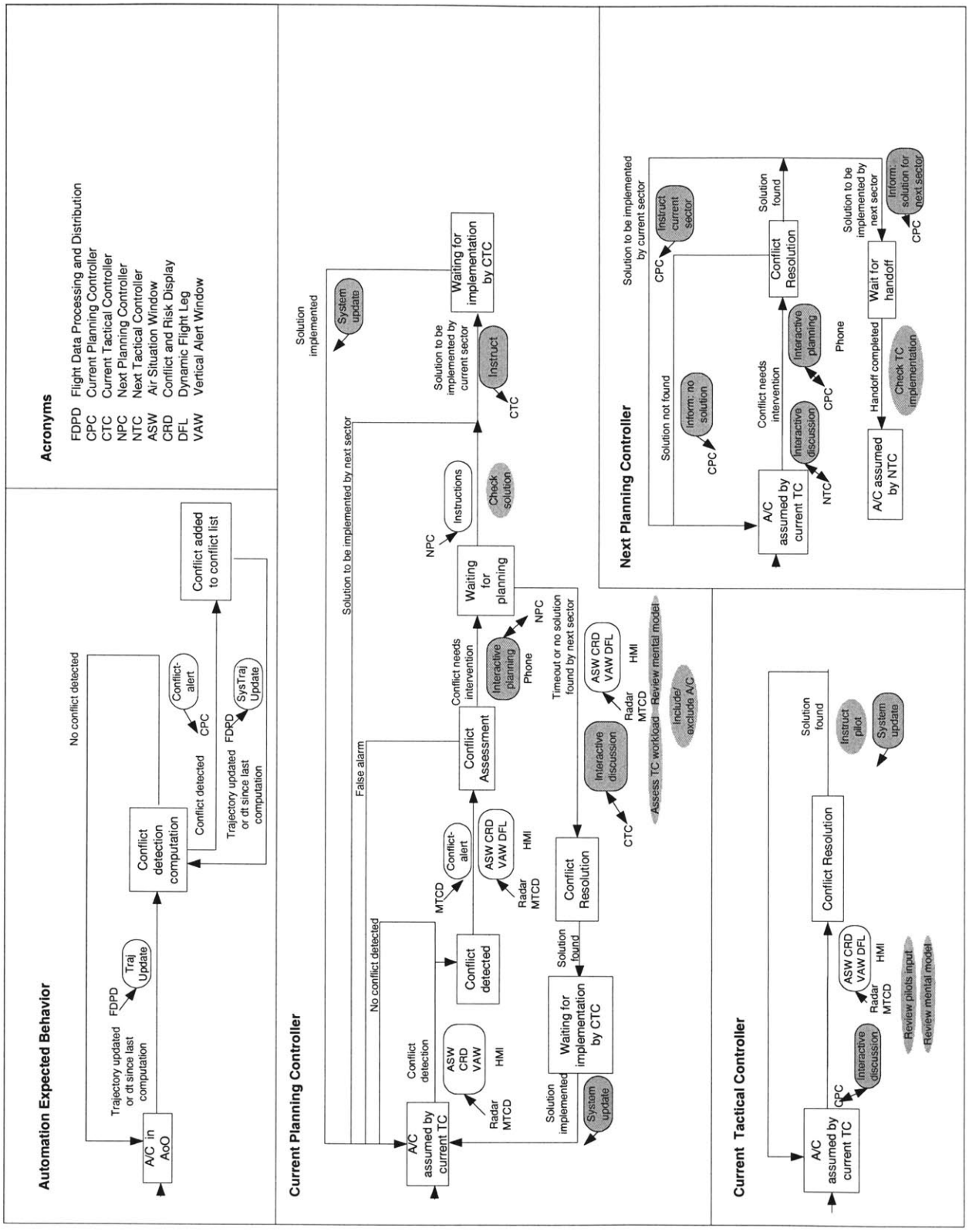


Figure 10: User and Task Models for MTCD

information to the controller, who, in return, only interacts with MTCD through excluding and re-including aircraft from the calculations. Finally, the tasks we are trying to describe in this model are cognitively complex, but include few well-identified or prescribed sub-tasks. The modeling technique seems therefore to be more valuable for systems where the automation has a higher level of authority and interacts more with the human operator, and where the operator relies more an a prescribed behavior or procedures that on his/her own analysis and judgment: as mentioned earlier, knowledge-based behavior is difficult to capture in user and task models.

The model built for MTCD does, however, add some value. Indeed, it represents in a simple and unambiguous way the working methods of the controllers, as well as their interaction. It puts MTCD in the context of the complex interactions going on in the system, and would help identify some limitations for the system or some undesirable side effects, both for the automation and the HMI. The model can also assist the operators themselves in understanding the required tasks and procedures. This idea will be addressed again later in this chapter.

5 Model Analyses

Once the user and task models have been built, several analyses can be performed to determine the safety, usability and performance of the system. The analysis techniques include the execution of the models, the use of animation and visualization tools that can be linked to the SpecTRM-RL model, the use of several safety analysis tools, or simple reviews by human factors, aviation and air traffic control experts. These analyses have not yet been applied to our MTCD blackbox model. It is our intention to analyze the automation SpecTRM-RL model, but the user and task model might not be analyzed with the automation because of the simplicity of the model, as explained above.

Our blackbox models are amenable to several types of analyses, as discussed before. Completeness and consistency analyses identify inconsistencies in the procedures or in the specification, or conditions not accounted for in the procedural and automation specifications [**30**, **48**]. Deviation Analysis provides a way to evaluate the specification for robustness against incorrect inputs, allowing the designer for instance to account for unexpected behaviors of the controllers [**59**]. This analysis is particularly important, as human behaviors are situation dependent and can change over time. Backwards Hazard Analysis starts from a hazardous state and works backward to determine if and how that state could be reached. If the analyses reveal the existence of a hazardous feature in the design, the designers can change the specification as the evaluation proceeds. Backwards analysis

techniques and hybrid modeling (necessary for the case of MTCD) are covered in [**54**], together with an application to MTCD.

6 Model Application: Identifying Mode Confusion Potential

A mode is defined in [**16**] as a "common architecture for grouping several machine configurations under one label". However, the understanding of modes and their potential contribution to confusion and error is still far from complete, and Degani *et al* give the example of Johnson and Engelbeck's demonstration of the widespread disagreement about what modes are, independently of how they affect users [**36**].

Leveson *et al.* identified six categories of system design errors or features that can lead to mode confusion errors [**44**]:

- Ambiguous interfaces
- Inconsistent system behavior
- Indirect mode transitions
- Lack of appropriate feedback
- Operator authority limits
- Unintended side effects

Degani has also identified some characteristics of Statechart models that he suggested could lead to mode confusion errors [**15**]. Leveson's error categories are different from Degani's, although there is an overlap, and Degani's Statechart features can be mapped into the Leveson categories. The Statechart models were, however, found to be difficult to apply to complex systems, whereas Leveson's specification language (SpecTRM-RL) was specifically designed to enable mode confusion analysis.

Leveson and Palmer made an initial attempt to learn more about how to identify such design flaws using a pilot error that has been the subject of many ASRS reports [**45**]. One result of this case study was the recognition that such mode confusion errors could only be identified if the software (automation) model was augmented by a simple model of the controller's view of the software's behavior (a user model)- the formal software specification was not enough. Later, Rodriguez *et al.* [**62**] investigated the utility of comparing user and pilot task models for detecting potential mode

confusion in a MD-11 Flight Management System (FMS) case study and demonstrated how several of the error-prone design features mentioned above can be identified using the Brown and Leveson formal models.

Because MTCD has a very simple behavior, no mode confusion instances have been detected in our analyses, although a problem may appear when the What-If Probing functionality is added. The designers should indeed make sure the controller knows at all time whether MTCD is in normal operation mode or in what-if probing mode: in situations of high workload, it is not uncommon for Air Traffic Controllers to begin a procedure and not finish it, so they might forget that they have started a what-if probing session.

In addition to MTCD, we also looked at SYSCO (System Supported Coordination), the automated communication tool proposed by EUROCONTROL that will probably be used with MTCD. SYSCO was only analyzed from a human-machine interaction perspective and some instances of Mode Confusion were detected. The higher level of complexity of SYSCO and the stronger involvement of the operators with SYSCO make this system a better test-bed for the Level 3 analyses than MTCD. SYSCO will not be discussed further in this document.

7 Model Application: Learning and Training

In addition to minimizing human error such as those related to mode confusion and situation awareness, task and user models can also be used to enhance learnability and simplify the training of human operators to interact with the automation.

In their application of the user model to an airliner's vertical descent guidance logic, Bachelder & Leveson [5] observed that the model not only enhanced detection of potential mode confusion features, but could also be useful as an operator display that showed current, previous, and anticipated system states. The model can also be used to accelerate and broaden the student operator system knowledge, as a better understanding of the systemic interactions and an easier approach to complexity is possible using the model than what is currently available for students and instructors. The student can for instance zoom-in and zoom-out of the model to view the system with different levels of details. Also, the instructor can hide certain states and ask the student to reason based on some given states, system modes, and inputs to find what the missing states should be. Several similar exercises can be conceived for training and testing purposes.

The model and the results of the simulations and analyses performed using the model can also be the basis for developing the training and operator manuals, traditionally included in Level 6 of the Intent Specification. Having a formal model, and the possibility to "zoom-in" and "zoom-out" to better deal with the complexity and the details of the system, adds indeed a lot of value. One can be easily convinced of this statement by looking at the manuals currently available, where just enough information is given for routine tasks and the rest is left to the operator to learn "on the job".

Our MTCD user/task model was not applied for these purposes. We did observe, however, that the model helped the people involved in the project better understand the tasks, the procedures, the interactions between the different system components, etc. The model was very helpful in providing us with a good understanding of the operational aspects of the system as a whole.

The models presented in this chapter are very important, as they allow a more formal approach to system analysis and design. This concept is at the heart of our methodology: combining formal and informal methods. The utility of the models, however, goes beyond the third level of the Intent Specification: the automation blackbox model could be used in Level 2 as a prototype for the simulations (see **Chapter 5**), and the user/task model can be used in Level 6 to define training procedures and manuals, or can even be used as a display to support the operator's situation awareness. Other applications of the Level 3 formal models are being studied at the Software Engineering Research Laboratory as well.

Conclusion and Future Work

We have presented in this document a human-centered approach to designing and developing safe systems. The methodology considers the human operators as a part of the system, and bases the automation design on the human needs and limitations- rather than trying to adapt the human to the automation and performing safety and human factors analyses *after* the fact. Starting with an Intent Specification structure, we achieved our goal by integrating methods such as Preliminary Task Analysis, Task Allocation, Task Analysis, User and Task Modeling, etc. The methodology covers the whole system life cycle and combines formal and informal methods. It also supports documentation and traceability.

The document also included an application of the methodology to a new Air Traffic Control tool, MTCD. The Level 1 steps (**Chapters 2-4**) were fully applied, including the Preliminary Task Analysis and the development of the User Requirements. The Level 2 (**Chapter 5**) and Level 3 (**Chapter 6**) analyses and models were described, but no simulations were performed on the system, as we believe that MTCD does not present the desirable features (such as control complexity) to illustrate the advantages of our formal analyses. The methodology will be applied in the future to some carefully selected systems.

As part of the future work related to our methodology, we are further developing the individual parts of the process and building procedures and tools to assist the designer. For example, designers need more structure and assistance in going from the requirements and the design principles derived in the process to a specific system design. There is also a need to build automated tools to assist with the analyses, especially for Level 3, as the formality of the models make them good candidates for automation. A better understanding of the analyses needed in Level 2 is also desirable, and would be achieved through experimentation on different systems. Finally, we are working on completing the task and user modeling language and on expanding its applications. Part of the possible applications were presented in **Chapter 6**.

References

1. R. Amalberti. La conduite de systemes a risques. *Le travail Humain*. PUF, Paris, 1996.
2. J.R. Anderson. Rules of the Mind. Hillsdale, NJ: Lawrence Erlbaum Associates, 1993.
3. J. Annett, K.D. Duncan. Task analysis and training design. Ocupational Psychology, 41:211-221, 1967.
4. E. Bachelder, N. Leveson. Describing and probing complex system behavior: a graphical approach. Society of Automotive Engineers Aviation Safety Conference, Seattle, WA, September 2001.
5. E. Bachelder, N. Leveson. A graphical language for describing complex system behavior: applications to design, training and user operation. Digital Avionics and Systems Conference, Daytona Beach, FL, October 2001.
6. C.E. Billings. Information transfer problems in tech aviation systems. Technical report, NASA, 1981.
7. C.E. Billings. Aviation automation: the search for a human centered approach. Lawrence Erlbaum, NJ, 1996.
8. C.E. Billings. Human-centered aviation automation: Principles and guidelines. NASA Technical Memorandum 110381, 1996.
9. G.A. Boy. Cognitive Function Analysis. Ablex Publishing. Westport, Conn. 1998.
10. M.C. Bressolle, R. Benhacene, N. Boudes, R. Parise. Advanced decision aids for Air Traffic Controllers: Understanding different working methods from a cognitive point of view. ATM Seminar 2000, Napoli, 2000.
11. M. Brown, N. Leveson- Modeling controller tasks for safety analysis- Workshop on Human Error and System Development, Seattle, April 1998.
12. K.M. Cardosi and E.D. Murphy. Human Factors Checklist for the Design and Evaluation of Air Traffic Control, 1995.
13. S. Cushing. Fatal words: communication clashes and airplane crashes. University of Chicago press, 1994.
14. H. David. Design of an interface for air traffic control. Workshop on Computational Semiotics for New Media, June 2000.
15. A. Degani. Modeling Human-Machine Errors: on Modes, Error and Patterns of Interaction. PhD thesis, Atlanta, GA: Georgia Institute of Technology, 1996.
16. A. Degani. Modes in human-machine systems: constructs, representation, and classification. The international Journal of Aviation Psychology, 1998.
17. M. Endsley, M. Rodgers. Attention distribution and situation awareness in air traffic control. In the 40th Annual Meeting of Human Factors Society, 1996.
18. Eurocontrol Experimental Centre. Validation of a methodology for predicting performance and workload. EEC note No. 7/99, June 1999.

19. Eurocontrol Experimental Centre. EATCHIP III Evaluation and Demonstration, PHASE 3 Project, Experiment 3Abis: MTCD, Final Report. EEC Report No. ### October 2000.
20. Eurocontrol Experimental Centre. Romania 99 Real-Time Simulation. EEC Report No. 346. May 2000.
21. Eurocontrol Experimental Centre, G. Glynn. Romania 99 Real-Time Simulation- CONTROLLER INFORMATION. EEC/RTO, May 1999.
22. Eurocontrol Experimental Centre, M. Bonnier, G. Glynn. Romania 99 Real-Time Simulation- SYSTEM HANDBOOK VERSION 5.0.EEC/ OPS, March 2001.
23. Eurocontrol Experimental Centre. Situation Awareness- Synthesis of Literature Search. EEC Note No. 16/00, Project ASA-Z-EC, December 2000.
24. Eurocontrol Experimental Centre. Eurocontrol EATCHIP Phase III HMI Catalogue, An Example of a Consistent HMI Application for EATCHIP Phase III Functions, January 1998.
25. R. Fields, P. Wright, M. Harrison. A task centered approach to analyzing human error tolerance requirements. Int. Symp. on Requirements Eng., 1995.
26. P.M. Fitts. Human engineering for an effective air-navigation and traffic-control system. NRC, Committee on Aviation Psychology, Washington, D.C., 1951.
27. M. Galouzeau de Villepin. A Safety-Centered Approach to Developing New Air Traffic Management Tools. MIT Thesis, Department of Aeronautics and Astronautics, May 2001.
28. F. Gamboa Rodriguez. Spécification et implémentation d'ALACIE: Atelier Logiciel d'Aide à la Conception d'Interfaces Ergonomiques. PhD thesis, Paris XI University, October, 1998.
29. S.G. Hart, L.E. Staveland. Development of NASA-TLX (Task Load Index): Results of experimental and theoretical research. In P.A. Hancock and N. Meshkati (eds.), *Human mental workload*, Amsterdam, North Holland, 1988.
30. M.P.E. Heimdahl, N.G. Leveson. Completeness and consistency in hierarchical state-based requirements. IEEE Transactions on Software Engineering, SE-22, No.6, June 1996.
31. M. Hook. A Description of the PUMA Method and Toolset for Modelling Air Traffic Controller Workload. Object Plus Limited Chandler's Ford, UK, 1996.
32. A. Jackson. HMI- Requirements to Implementation: Learning from Experience. FAA/EEC Workshop on "Controller Centered HMI", Toulouse, France, April 1999.
33. J. Javaux. The prediction of pilot-mode interaction difficulties: spreading activation networks as an explanation for frequential and inferential simplifications. 10th Int. Sym. On Aviation Psychology, May, 1999.
34. D. Javaux, P. Polson. A method for predicting errors when interacting with finite state machines: the impact of implicit learning on the user's model of the system. Human Error, Safety, and System Development, Liege, June 1999.
35. Johnson, W.G. *MORT Safety Assurance Systems*, Marcel Dekker, Inc., 1980.
36. J. Johnson, G. Engelbeck. Modes survey results. SIGCHI Bulletin, 20(4), 38-50, 1989.

37. W.N. Kaliardos, R.J. Hansman. Semi-structured decision processes: a conceptual framework for understanding human-automation decision systems. PhD thesis, Cambridge, MA: Massachusetts Institute of Technology, 1999.
38. D.E. Kieras. Towards a practical GOMS model methodology for user interface design. In M. Hellander (ed.) The Handbook of Human-Computer Interaction, North Holland, Amsterdam, pp. 135-158, 1988.
39. B. Kirwan *et al.* Human Factors in the ATM System Design Life Cycle. FAA/Eurocontrol ATM R&D Seminar, Paris, France, 16 - 20 June, 1997.
40. W. Langewiesche. Slam and Jam. The Atlantic Monthly; Volume 280, No. 4, pages 87-100, October 1997.
41. M. Leroux. The role of verification and validation in the design process of knowledge based components of air traffic control systems. In J.A. Wise, V.D. Hopkin, P. Stager, *Verification and validation of complex systems: human factors issues.* Springer-Verlag, Nato Asi Series, Berlin, 1993.
42. N.G. Leveson. Safeware, system safety and computers. Addison-Wesley Publishing Company, 1995.
43. N. Leveson *et al.* Demonstration of a Safety Analysis on a Complex System. Presented at the Software Engineering Laboratory Workshop, NASA Goddard, December 1997.
44. N. Leveson et al. Analyzing Software Specifications for Mode Confusion Potential. Presented at the Workshop on Human Error and System Development, Glasgow, March 1997.
45. N. Leveson and E. Palmer. Designing Automation to Reduce Operator Errors. In the Proceedings of Systems, Man, and Cybernetics Conference, Oct. 1997.
46. N.G. Leveson, J.D. Reese, M. Heimdahl. SpecTRM: A CAD System for Digital Automation. Digital Aviation Systems Conference, Seattle, November 1998.
47. N.G. Leveson. Intent Specifications: An Approach to Building Human-Centered Specifications. IEEE Trans. on Software Engineering, January 2000.
48. N.G. Leveson. Completeness in formal specification language design for process-control systems. ACM Formal Methods in Software Practice, Portland, August 2000.
49. W.S. Luffsey. How to become an FAA Air Traffic Controller. Random House, NY, 1990.
50. R.J. Lysaght, S.G. Hill, A.O. Dick, B.D. Plamondon, P.M. Linton, W.W. Wierwille, A.L. Zaklad, A.C. Jr. Bittner, R.J. Jr. Wherry. Operator workload: Comprehensive review and evaluation of workload methodologies (ARI Technical Report 851). Alexandria, VA, U.S. Army Research Institute for the Behavioral and Social Sciences, 1989.
51. C.M. Mitchell. Task-analytic models of human operators: designing operator-machine interaction. Technical report, Georgia Institute of Technology, 1996.
52. R. Mogford. Mental models and situation awareness in air traffic control. The International Journal of Aviation Psychology, 7 (4), 1994.
53. R. Morrison, R.H. Wright. ATC control and communication problems: an overview of the recent ASRS data. In the Fifth Int. Symp. in Aviation Psychology, 1989.
54. N. Neogi. Hazard Elimination Using Backwards Reachability Techniques in Discrete and Hybrid Models. MIT Thesis, Department of Aeronautics and Astronautics, January 2002.

55. S. Pheasant. Bodyspace - Anthropometry, Ergonomics and Design. Taylor and Francis, 1998.
56. J. Rasmussen. Information-processing and human-machine interaction: an approach to cognitive engineering. Elsevier, NY, 1986.
57. J. Rasmussen. Cognitive control and Human error mechanisms. In J. Rasmussen, K. Duncan, J. Leplat, editors, New Technology and Human error, pp. 53-61, John Wiley & Sons, New York, 1987.
58. R.E. Redding. Analysis of operational error and workload in air traffic control. In Human Factors Society, 1992.
59. J.D. Reese, N.G. Leveson. Software Deviation Analysis. International Conference on Software Engineering, Boston, May 1997.
60. G.B. Reid, T.E. Nygren. The subjective workload assessment technique: A scaling procedure for measuring mental workload. In P.A. Hancock and N. Meshkati (eds.), *Human mental workload*, Amsterdam, North Holland, 1988.
61. M. Rodgers. An examination of operational error database for en-route air traffic control centers. Technical report, FAA Civil Aeromedical Institute, 1993.
62. M. Rodriguez, M. Zimmerman, M. Katahira, M. de Villepin, B. Ingram, N. Leveson- Identifying mode confusion potential in software design- DASC, Philadelphia, PA, October 7-13 2000.
63. E. Salas, D.P. Prince, L. Shrestha. Situation awareness in team performance: implications for measurement and training. *Human Factors*, 1995.
64. M.S. Sanders and E.J. McCormick. Human Factors in Engineering and Design (7th Edition). London: McGraw-Hill, 1992.
65. N.B. Sarter, D.D. Woods- How in the world did we get into that mode? Mode error and awareness in supervisory control- Human Factors, 1995, 37(1), 5-19.
66. D.L. Scapin, C. Pierret-Golbreich. Toward a method for task description: MAD. In Work with Display Units 89, L. Berlinguet and D. Berthelette, Elsevier Science Publishers B.V. (North Holland), 1990.
67. T.B. Sheridan. Human centered-automation: oxymoron or common sense? Keynote address, IEEE Systems, Man, and Cybernetics Conference, Vancouver, BC, Oct 23-25, 1995.
68. Stager, Hameluck. Factor in air traffic control operating irregularities. Technical report TP 9324E, Transport Canada, 1990.
69. Stager, Hameluck, Jubis. Underlying factors in air traffic control incidents. In The 33rd annual meeting of human factors society, 1989.
70. F. Vanderhaegen. Multilevel organization design: The case of the air traffic control. Control Eng Practice, 1997.
71. K.J. Vincente, J. Rasmussen. Ecological interface design: theoretical foundations. IEEE Transactions on Systems, Man and Cybernetics, SMC-22, 589-606, 1992.
72. Wiener, E.L., Chidester, T.R., Kanki, B.G., Palmer E.A., Curry, R.E., and Gregorich, S.E. The Impact of Cockpit Automation on Crew Coordination and Communications. NASA Ames Research Center, 1991.

73. C.D. Wickens, A.S. Mavor, R. Parasuraman, J.P. McGee. The future of air traffic control: human operators and automation. National Academy Press, Washington, D.C., 1998.

74. G. Widup, M. Hayden, B.Benaway. Basic ATC study guide. SATUSA, issue No. 003, December 1999.

75. D.D. Woods. Coping with complexity: the psychology of the human behavior in complex systems. In L.P. Goodstein, H.B. Andersen, S.E. Olsen, *Tasks, errors and mental models*. Taylor & Francis, London, 1988

76. R.M. Young, T.R.G. Green, T. Simon. Programmable user models for predictive evaluation of interface designs. In CHI '89, pages 15-19, May 1989.

77. G.R. Yost. Implementing the Sisyphus-93 task using Soar/TAQL. International Journal of Human-Computer Studies, 44(3-4), 1996.

Appendix A

Preliminary Hazard Analysis

A PHA is used in the early life cycle stages (Level 1 of the Intent Specifications) to identify critical system functions and broad system hazards; it is the heart of any system safety program. It should be started early so that the information can be used in tradeoff studies and selection among design alternatives. This process, however, is not done once: it is iterative, with the PHA being updated as more information about the design is obtained and as changes are made in the system. A PHA was performed for the system defined below, and the results were used in the different levels of the design methodology: developing requirements, preparing design specifications, operational procedures, test and management planning, etc. We present in this appendix the methods used to perform this PHA, as well as some examples of the results obtained.

1 System and Subsystems

The EUROCONTROL system (**Figure 1**) is complex and involves several interacting subsystems. It is crucial therefore to define the system and the environment clearly since the beginning of the analysis process. We opted for the following boundaries:

- The system includes:
  - MTCD
  - SYSCO
  - The controllers, PC and TC, in the sector at hand
  - HMI as a blackbox
- The environment includes:
  - FDPD, EDPD, HMI, AMAN, MONA, Safety Nets
  - The controllers in the adjacent sectors
  - The pilots

From our system's point of view, the elements of the environment will only be represented as flows of information, inputs and outputs coming into or going from the system, like an information network. Therefore, it is mostly the interactions between the elements of the system and the environment that matter, while inside the system both the individual behavior of the elements and their interaction have to be taken into account. It is very important to keep this notion of network and interactions in mind, since accidents are seldom due to a single factor: they are generally related to a flaw in the interaction of two factors or more. Note that HMI is considered as part of the environment because of the lack of available information on its behavior. The flows of information through HMI are, however, part of the system, and therefore will appear in our PHA.

2 Hazard Identification

2.1 Accidents, Hazards and Causes

It is important to distinguish a hazard from an accident/incident and from a cause. Accidents or incidents are what a hazardous state can lead to, like two aircraft colliding. Causes, which can be divided into systemic, contributing and direct factors, lead to the hazardous states. In order to determine these hazards and causes, it is often very useful to use existing accident investigations and identify what interactions in the system or the system and its environment are most likely to fail and how this happens. The process of finding these problems is rather random, and we felt that having an idea about the factors that play a role, however important this role is, can be very helpful.

This led us to develop a checklist of the contributing factors in the Air Traffic Control system in general (**Figure 11**). Judgment will probably stay the discriminating factor, but knowing what criteria can intervene in a fault tree for instance makes the analyst's task easier in that he or she will have a backup material to rely on if he/she cannot see how to go further. It will also be of a big value in some cases where the completeness of the decomposition might be crucial: if the analyst clearly identifies the criterion used and the different values that the output can have (e.g. workload high, adequate or low), he/she can make sure no important option was overseen that, eventually, could lead to the hazard under study.

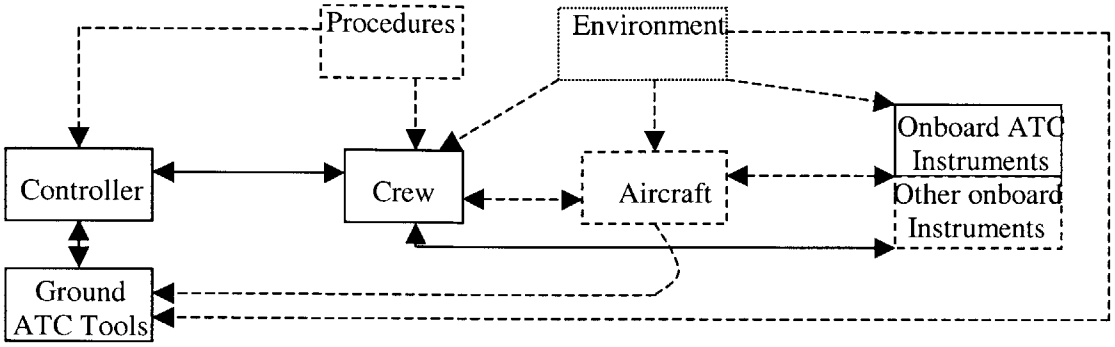


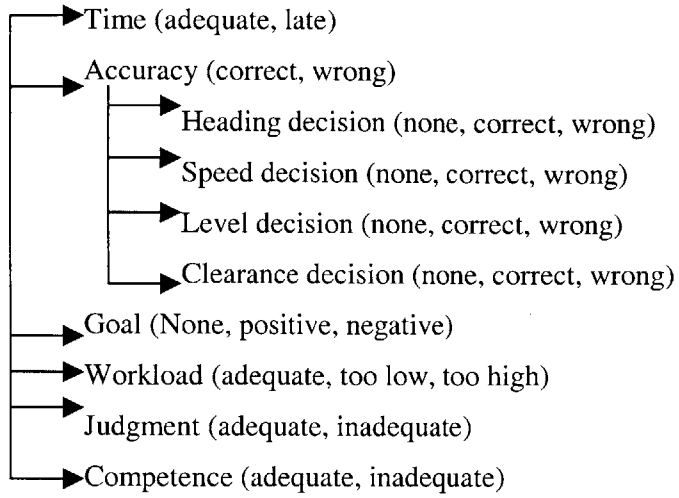
Figure 11: System, subsystems (full lines) and environment (dashed lines)

2.2 Checklist for Single Point Errors/Failures

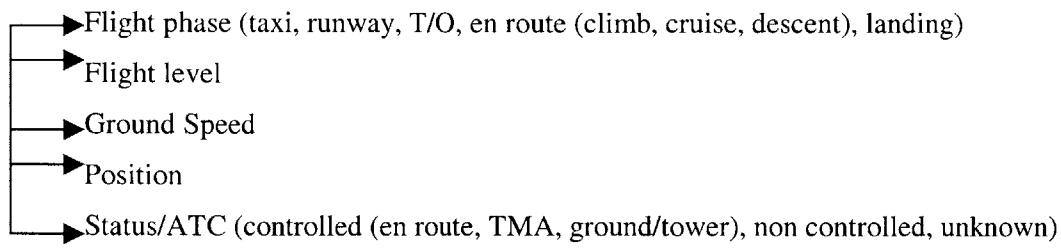
For each of the subsystems represented in **Figure 11**, we give a list of the related criteria we identified (i.e. possible reasons for malfunction), as well as the different values these criteria can take. The lists are not exhaustive; a closer study of previous accident investigations and fault trees should be done. Note that we only give the criteria related to a single subsystem here. Interactions will be described in the following section. The ground ATC tools listed are those of the EUROCONTROL EATCHIP Phase III.

1. The controller
   - Time (adequate, late)
   - Accuracy (correct, wrong)
     - Heading advisory (none, correct, wrong)
     - Speed advisory (none, correct, wrong)
     - Level advisory (none, correct, wrong)
     - Clearance advisory (none, correct, wrong)
   - Goal (None, positive, negative)
   - Workload (adequate, too low, too high)
   - Judgment (adequate, inadequate)
   - Competence (adequate, inadequate)

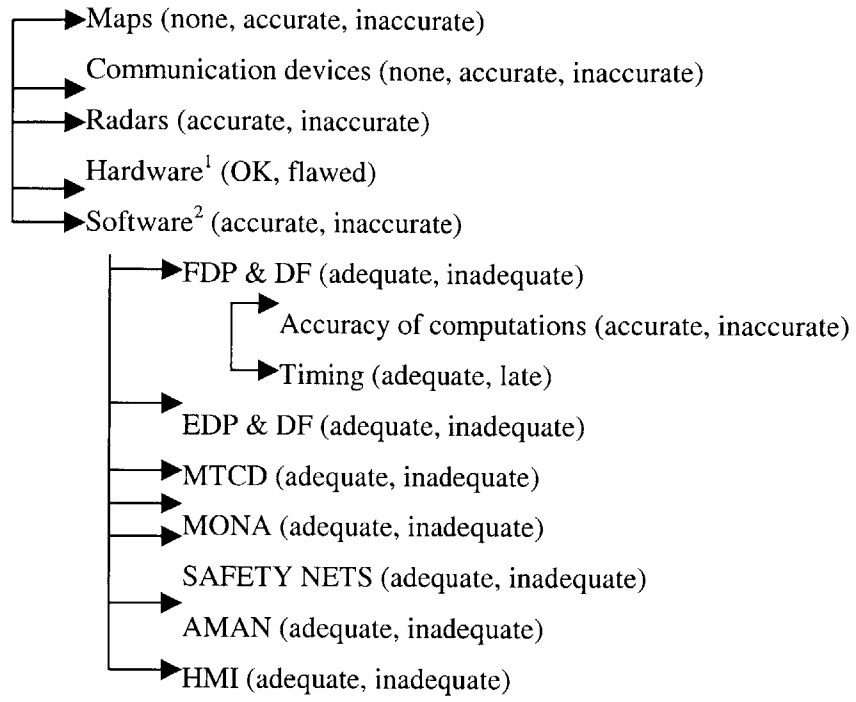
2. The crew (pilot, co-pilot, etc.)



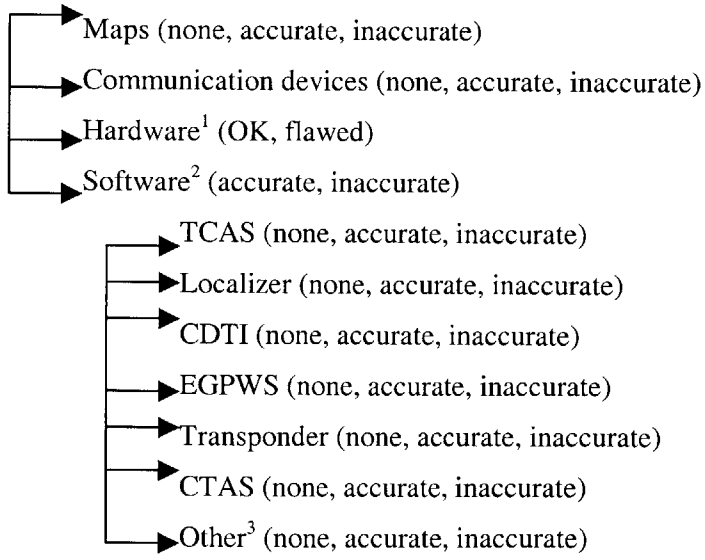
3. The aircraft



4. ATC tools (ground)



5. On-board ATC-related instruments



[1] Includes displays, computers, wiring, etc.

[2] Each of the following tools should be decomposed depending on its requirements and specifications. This is obviously not done here.

[3] To be identified.

6. Non ATC-related on-board instruments

   Accuracy (accurate, inaccurate)

7. Environment

   - Weather (unknown, favorable, unfavorable)
   - Static/moving object[4] (none, hazardous, non-hazardous)
   - Other (none, favorable, unfavorable)

8. Procedures

   (None, accurate, inaccurate)

2.3 Checklist for the Interaction Between Two Factors

We give here a partial checklist to illustrate the ways in which the interaction between two elements of the system can malfunction.

1. Controller/Crew

   - Detectability of message[5] (none[6], detectable, undetectable, unknown)
   - Intelligibility of message (none, intelligible, unintelligible, unknown)
   - Understandability of advise/feedback (none, understandable, non-understandable, unknown)
   - Advise/feedback taken into account (none, yes, no, unknown)

2. Controller/Rules-Procedures

   - Procedures considered (none, yes, no, unknown)
   - Procedures understood (none, yes, no, unknown)
   - Procedures followed (none, yes, no, unknown)

---

[4] Includes buildings, birds, etc.

[5] e.g. wrong end-user, bad links, etc.

[6] The value "None" means that there was no message.

3. Crew/Rules-Procedures
   - Procedures known/considered (none, yes, no, unknown)
   - Procedures understood (none, yes, no, unknown)
   - Procedures followed (none, yes, no, unknown)

4. Rules-Procedures/Aircraft
   - Airspace (none, authorized, special)
   - Flight envelope (none, respected, not respected)
   - Flight level (none, authorized, unauthorized)
   - Separation standards (none, respected, not respected)

5. Crew/Environment

   Effect[7] (none, positive, negative)

6. Crew/Aircraft

   Control of the aircraft (adequate, inadequate, unknown)

7. Crew/On-board ATC-related instruments
   - Use of maps (none, adequate, inadequate)
     - Readability (readable, unreadable, unknown)
     - Understandability (understandable, non-understandable)
   - Use of communication devices/hardware (adequate, inadequate)
   - Use of software (adequate, inadequate)
     - Use of TCAS (none, adequate, inadequate)
       - Clarity of display (clear, not clear)
       - Detectability of warning (none, detectable, undetectable)
       - Warning taken into account (none, yes, no)
     - Other (none, adequate, inadequate)

8. Crew/On-board non ATC-related instruments

   Use (adequate, inadequate)

---

[7] Effect of weather on crew: psychological, stress, workload, heath, etc.

9. Ground ATC tools/On-board ATC-related tools
   - Compatibility of advisory (compatible, incompatible, unknown)
   - Data exchange[8] (adequate, inadequate, unknown)

10. Controller/Ground ATC tools
    - Use of maps (none, adequate, inadequate)
    - Use of radars/hardware/communication tools (adequate, inadequate)
    - Use of software (adequate, inadequate)
      - Use of HMI (none, adequate, inadequate)
        - Clarity (clear, not clear)
        - Understandability of displayed data (understandable, non-understandable)
        - Commands entered (correctly, not correctly)
        - Mode (adequate, inadequate)

11. Ground ATC tools-Ground ATC tools

    See interactions between EATCHIP components.

2.4 Hazards for the MTCD System

Using the previous checklist, as well as several references, such as N. Leveson's hazard analysis for CTAS in [**43**], we identified a list of hazards and causes that will be used in the second phase of the PHA, the Fault Tree Analysis. The high-level hazards identified for the system defines above are the following:

1. A pair of controlled aircraft violate minimum separation standards.
2. A controlled aircraft enters restricted airspace without authorization.
3. A controlled aircraft descends below the standard minimum flight level.

3 Fault Tree Analysis

Fault trees are perhaps the most widely used system hazard analysis technique. We created a partial fault tree for our system, stressing the hazards included in our system more than those in the

[8] Through transponder, Data Link connections, etc.

environment and privileging the software and human related hazards. Remember that several iterations along the whole life cycle of the system are necessary before fault tree can be considered as sufficient. Therefore, the fault trees should be considered as temporary and fairly high-level trees that are meant to be refined as we build the system. Finally, we adapted the fault tree representation used in [**43**] instead of the standard representation, because it allows more information in each "box" and more "boxes" per page, thus increasing readability and understandability. These fault trees developed for MTCD can be found in [**27**]. A more extensive fault tree was built for EUROCONTROL.

Appendix B

Glossary of Terms and Acronyms

A/C	Aircraft
ASW	Air Situation Window
ATC	Air Traffic Control
ATCO	Air Traffic Controller
ATM	Air Traffic Management
CRD	Conflict and Risk Display
DFL	Dynamic Flight Leg
EATCHIP	European Air Traffic Control Harmonization and Integration Programme
ECAC	European Civil Aviation Conference
EDPD	Environment Data Processing and Distribution
EEC	EUROCONTROL Experimental Centre
FAA	Federal Aviation Agency
FDPD	Flight Data Processing and Distribution
FL	Flight Level
GUI	Graphical User Interface
HMI	Human Machine Interface
ICAO	International Civil Aviation Organization
MONA	Monitoring Aids
MTCD	Medium Term Conflict Detection

PC	Planning Controller
PHA	Preliminary Hazard Analysis
PPT	Potential Problem Display
PTA	Preliminary Task Analysis
SERL	Software Engineering Research Laboratory (M.I.T.)
STCA	Short Term Conflict Alert
SYSCO	System Supported Coordination
TC	Tactical Controller
TCAS	Traffic Alert and Collision Avoidance System
TP	Trajectory Probe/ Predictor
VAW	Vertical Alert Window