A Safety-Centered Approach to Developing New Air Traffic Management Tools

by

Maxime Galouzeau de Villepin

Engineering Degree (1999)
École Nationale Supérieure de l'Aéronautique et de l'Espace
SUPAERO
Toulouse, France

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of

Master of Science in Aeronautics and Astronautics

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2001

Author .....
Department of Aeronautics and Astronautics
May 11, 2001

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 Safety-Centered Approach to Developing New Air Traffic Management Tools

by

Maxime Galouzeau de Villepin

Submitted to the Department of Aeronautics and Astronautics
on May 11, 2001, in partial fulfillment of the
requirements for the degree of
Master of Science in Aeronautics and Astronautics

Abstract

This thesis describes a safety-centered approach for developing new air traffic management tools as the growing demand for increased system capacity leads to the introduction of new automation.

We use a Medium Term Conflict Detection tool, developed by Eurocontrol in Europe, as a baseline to demonstrate how to safely specify the requirements and implement the design.

The methodology we use is supported by a new specification structuring approach called "Intent Specifications", that supports traceability and documentation of design rationale as the development process proceeds. The Intent Specifications method has been developed and used for several years in the Software Engineering Research Laboratory at M.I.T under the supervision of Prof Leveson.

In this thesis we achieved the first three levels of this methodology applied on the Medium Term Conflict Detection tool. The first level gives the system goals, defines the environment, and the system requirements and constraints. The second level contains the scientific and engineering system design principles as well as the tradeoffs and the rationale for these design decisions. The third level describes a formal model of the blackbox behavior of the system components using a specification language.

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

Acknowledgments

I would like to thank all the people who helped me to complete this project:

My advisor, Prof Leveson, for her support for this thesis which allowed me to discover a new field: Software Engineering. I believe your approach will help engineers and scientists deal with increasingly complex and sophisticated software systems for the good of all.

All my labmates: Israel, Natasha, Marc, Anand, Mirna, John, Ed, JK, Kristina and Masa for the work achieved together and for the great atmosphere. It has been a pleasure working with you in such a nice and international lab.

My professors in SUPAERO, my university in Toulouse France. Pr Colongo, Pr Roques and Mr Vimont thanks to whom I spent two great years in Boston studying at M.I.T.

My friend, Antoine Deux for his support during these two last months writing this thesis in time for the deadline.

Finally, a very special thanks to my future wife, Anne, for accepting to follow me to this side of the Atlantic, in the wild New World. Thank you also for Frostbite sailing with me during the Boston-be-damned-winter by -22 degrees Fahrnerheit.

Contents

List of Figures

Problem and Approach

0.1 Introduction

In the introduction of our paper [29], we wrote:

> The current Air Traffic Control systems have proven over time to be very safe. This high safety level can be attributed to a variety of factors: a high level of professionalism in the ATC work force, designed-in redundancy and mutual checking, large error margins (e.g. in the separation criteria for aircraft), and loose coupling (so that errors are contained and do not propagate rapidly throughout the various components).The limits of such a conservative system, however, along with growing demand for increased system capacity are leading to the introduction of new automation.
>
> Automation has the potential to overcome human perceptual and cognitive limits and to reduce and eliminate specific common human errors in the current system, such as those that arise in human voice communication. At the same time, computer automation and assistance has led to new types of human errors.

The Medium Term Conflict Detection (MTCD) is an automated tool developed in the Eurocontrol Air Traffic Control organization in Europe to help controllers detect and predict aircraft involved in conflicts.

We use MTCD in this study to demonstrate how to safely specify the requirements and implement the design of such an automated tool highly interacting with a human controller.

0.2 Previous work

In [28], Pr Leveson and her students have developed a theoretical foundation for safety in complex systems and building a methodology upon that foundation. They gave a demonstration of this methodology on the Center-TRACON Automation System (CTAS) portion of the air traffic control (ATC) system and procedures currently employed at the Dallas/ Fort Worth TRACON (Terminal Radar Approach CONtrol). CTAS is an automated system to assist controllers in handling arrival traffic in the Dallas/ Fort Worth area.

> The methodology (as described in her book *Safeware* [22]) includes special management structures and procedures, system hazard analyses, software hazard analysis, requirements modeling and analysis for completeness and safety, special software design techniques including the design of human machine interaction, verification, operational feedback and change analysis.
> The *Safeware* methodology is based on system safety techniques that are extended to deal with software and human error. Automation is used to enhance our ability to cope with complex systems. Identification, classification and evaluation of hazards is done using modeling and analysis. To be effective, the models and analysis tools must consider the hardware, software and human components in these systems.[28]

Pr Leveson and her studends showed that safety is a system property, not a component property, so our safety analysis for MTCD considers the entire system and not simply the automated components. They also showed that because safety analysis of a complex system is an interdisciplinary effort, teams shall include system engineers, software engineers, human factors experts, and cognitive psychologists.

In this thesis, we use the experience gained during this previous work on CTAS to give a safety-centered demonstration to developing the MTCD new Air Traffic Management (ATM) tool.

The methodology we use is supported by a new specification structuring approach, described in the CTAS work and extended in this study. It is called "Intent Specifications"

and supports traceability and documentation of design rationale as the development process proceeds.

0.3 Intent Specifications or "Why" abstraction

As described in [28]:

> The design of intent specifications is based on the fundamental principles of problem-solving and abstraction that humans use to make complex tasks intellectually manageable. The problems in performing system and software engineering activities such as safety evaluations are rooted in complexity and intellectual manageability. Psychologists have found that complexity itself is not a problem if humans are presented with meaningful information in a coherent, structured context. [...]
>
> The approach found to be effective in dealing with complexity is hierarchical abstraction, that is, structuring the situation such that the problem solver can transfer the problem to a different level of abstraction. In general systems theory, models of complex systems can be expressed in terms of a hierarchy of levels of organization, each more complex than the one below, where a level is characterized by having emerging properties. [...]
>
> The goal of hierarchical abstraction is to allow both top-down and bottom-up reasoning about complex system. A higher level of the usual software specifications can be thought as providing "what" information while the next lower level describes "how". Such hierarchies, however, do not provide information about "why". [...]
>
> Each level of an intent abstraction contains the goals or purpose for the level below and describes the system in terms of a different set of attributes or language. Higher level goals are not constructed by integrating information from lower levels; instead each level provides different, emergent information with respect to the lower levels. [28]

System and software specifications are then organized using a specifications modelling language called SpecTRM-RL (Specification Tools and Requirements Methodology-Requirements Language) along three abstractions dimensions (as shown in figure 0-1): intent, refinement and part-whole decomposition.

The vertical dimension specifies the level of intent at which the problem is being considered, i.e. the language or model that is currently being used. The decomposition and refinement dimensions allow users to change their focus of attention to more or less detailed views within each level or model. The information at each level is fully linked to related information at the levels above and below it.

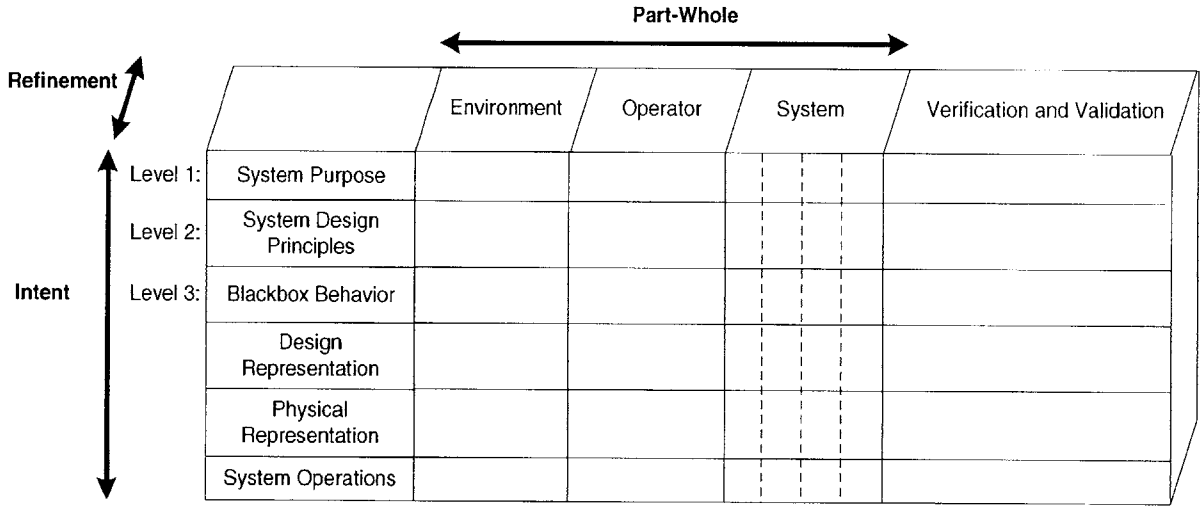

Figure 0-1: The form of an Intent Specification

The intent dimension has five levels of abstraction. The highest level of the specification contains the overall goals and safety constraints. Some of the information here is generated through the preliminary hazard analysis process. The next lower level contains the underlying scientific principles upon which the design at the lower levels is based and through which the goals and constraints ar the highest levels are satisfied. The third level contains the blackbox functional behavior model.This level is the most appropriate one for formal methods. The fourth and fifth levels contain design and implementation information.

Because each level is mapped to the appropriate parts of the intent levels above and below it, traceability of design rationale and design decision is pro-

vided from high-level system requirements and constraints down to code and vice versa.[28]

0.4 Our work

In this thesis we will focus on the first three levels of this approach. We apply them to the Medium Term Conflict Detection tool currently developed in Europe by the European Air Traffic Control agency Eurocontrol.

In the first level, we focus on the historical context, the system functional goals, and the description of the environment, and of the assumptions and constraints of this environment on the system. This level also contains our results of a preliminary hazard and task analyses that conduct to the system requirements, constraints and limitations.

In the second level, we describe the basic system scientific and engineering design principles needed to achieve the behavior specified in the first level. We also highlight the tradeoffs and the rationale for the design decisions that have been made for MTCD.

Finally in the third level, we will describe the blackbox behavior of the system components, including humans, and the logical aspects of the interfaces between the components. This model will have an underlying formal model to be later executable and subject to analysis.

Chapter 1

Level 1, System purpose and properties

As described by Leveson [23], this level of the intent specification contains the historical context, the system functional goals, requirements, constraints and limitations.
It also contains the description of the environment, and the assumptions and constraints of this environment on the system.
Results of preliminary hazard and task analyses for system-level qualities are also given in this level.

Most of the information at this level is generated during the conceptual design phase, but information will probably be added or changed during later design phases. This level will be the starting place for those unfamiliar with the system when trying to learn about it and to understand why it was designed the way it was.

1.1 Description of our Methodology

We give an outline of the methodology that links the different sections we develop in this level.

The historical context section highlights the reasons why the decision to build an automated conflict detection tool has been made. Basically, traffic is increasing, controller's workload is increasing so we want automation to take over part of the monitoring task.

The decision making process is described using Rasmussen socio-technical risk model [31]. We bring the historical information and context into the Rasmussen model framework to better understand how the decision to build MTCD (and how to build it) has been taken. This part of our Intent Specs is supposed to explain "why" this automated conflict detection tool is now being developed and implemented.

Based on that historical context, we write in general terms the main functional goals of MTCD.

Once we have the functional goals issued from the historical context, we describe the environment in which MTCD is to be designed.
We will also give the related environment's assumptions and constraints.

Once that we know the MTCD functional goals and the environment in which it is supposed to be built, we perform in parallel a Preliminary Hazard Analysis (PHA) and a Preliminary Task Analysis (PTA).

The Preliminary Hazard Analysis makes sure we identify early enough the hazards in the design process. It includes:

- A hazard list
- A fault tree analysis
- Safety requirements

The Preliminary Task Analysis makes sure that we adopt a Human centered approach when designing our system. For the Preliminary Task Analysis, we:

- Describe the Air traffic controller high-level goals,
- Describe the current responsibilities the controller has to fulfill to achieve these high-level goals,
- Describe the human automation task allocation principles, and give the rationale for these principles,
- Trace these principles down to the MTCD system requirements.

These two preliminary analyses are conducted in parallel and allow us to write and to classify our system requirements and constraints.

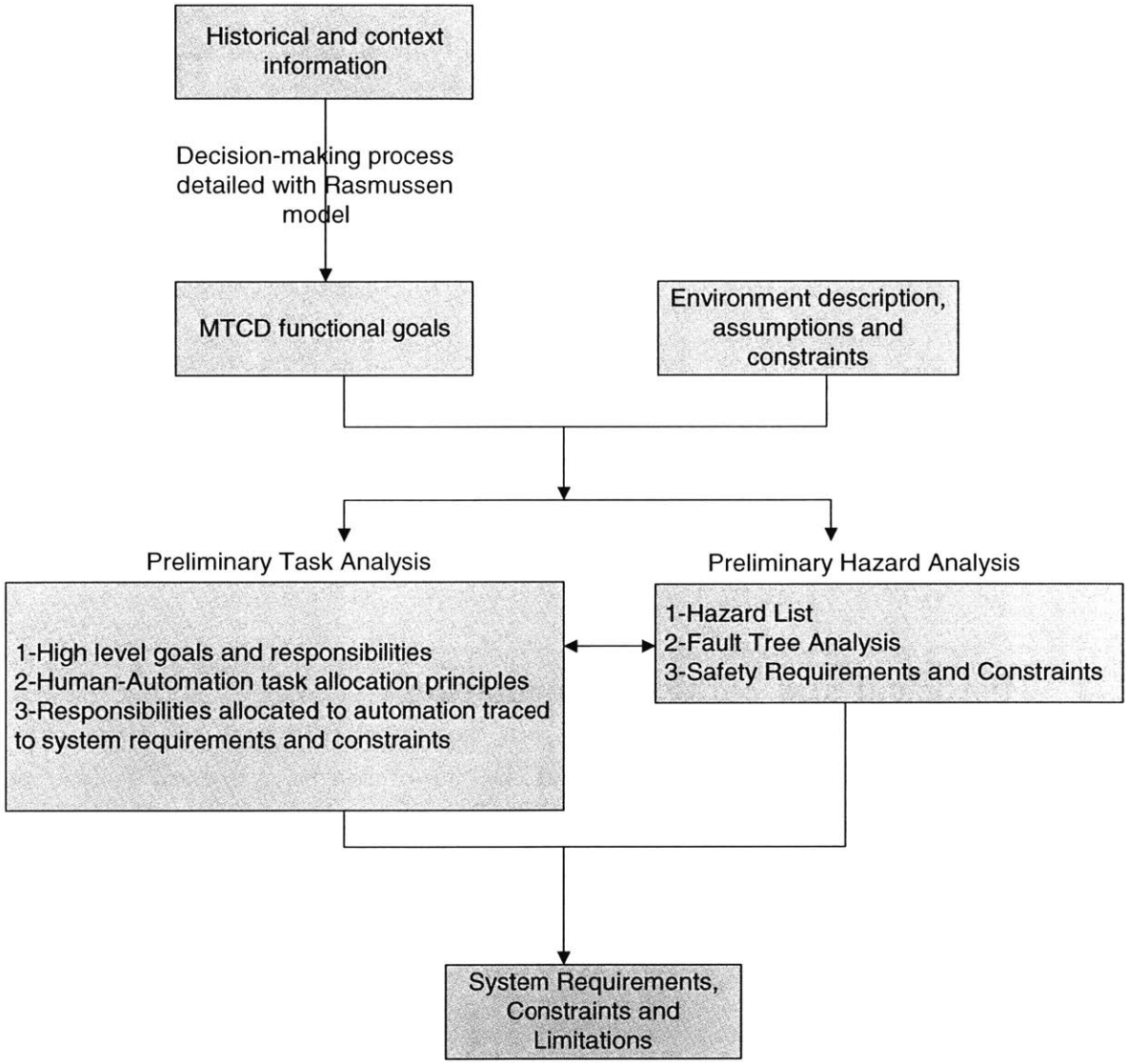

Figure 1-1: Level 1

1.2 Introduction

This introduction gives a brief overview of the system for those who have no information about it.

MTCD will support controllers in monitoring the future air traffic situation and identifying potential conflict situations on a medium-term time scale.

Because of the timely notification of potential conflict situations, MTCD will offer the controller time to assess the conflict, and, if necessary, to undertake deliberate actions to resolve the conflict. Timely notification should result in earlier conflict resolution and better planning.

Because MTCD will take over part of the monitoring tasks from the controller, the controllers should be able to handle more traffic. MTCD should therefore enable an increase of sector capacity.

Furthermore, MTCD will support the controller in handling more complex traffic, such as traffic on random point routings, without a decrease in safety levels.

1.3 Historical and context information

This section is based on previous work done at the Eurocontrol Experimental Center, in France [10]. We attempt to describe and highlight some relevant historical and context information about the development of the system (MTCD) or of related systems.

1.3.1 Today

In today's ATC environment, controllers monitor flights by scanning flight progress strips and radar displays, in order to predict future air situations. Simultaneously, controllers are monitoring one aircraft and relating its movement to the total air situation at all moments in the near future through the sector airspace. The controllers are also responsible for resolving any conflicts.

Traffic is now increasing, causing delays and airspace congestion. Both FAA and Eurocontrol are exploring new solutions to cope with the increasing traffic and to maintain the same level of safety for Air Traffic Control. The introduction of automated tools to support the controller's task is mainly considered as a solution to this problem.

1.3.2 Fixed and random point routings

At the moment, the majority of flights use fixed point routings along fixed ATS routes with limited capacity. One possible way to increase the capacity of the airspace is to allow random point routings, i.e. routings that do not follow the fixed ATS routes (this is also called free flight).
Free flight, along with the increase of air traffic, will however increase controller's workload. MTCD is viewed as a solution to help controllers with increased complexity of dealing with random point routing.

1.3.3 MTCD into operations

MTCD is foreseen to become operational between 1998 and 2005. Within this time frame, aircraft equipment and ground equipment will improve. This improvement may allow change of separation criteria, and reduction of the deviation of aircraft from predicted trajectories.

1.3.4 Future other functions

At the moment the Human Machine Interface (HMI) and AMAN (Arrival Manager) are the only functions currently identified to be provided with conflict data. In the future, other functions (e.g. conflict resolution advisory, what-if-probing) might use MTCD.

1.3.5 Responsibility for separation

The advent of new technology has started to reverse the trend of relying on separation assurance by a ground system. The successful introduction of Airborne Collision Avoidance (ACAS) Resolution Advice has introduced the first changes in pilot/controller relationships. In addition, Airborne Separation Assurance (ASAS) is conceivable through the introduction of Cockpit Displays of Traffic Information.

However MTCD is still based on the continuation of a ground-based service-providing ATC system. ACAS may already be in operation, ASAS has not left the research laboratories and has a long way to go yet before implementation. Until then, ground ATC systems need to develop enhancements to increase their level of service in the interest of safety and capacity.

1.3.6 Previous work on conflict detection

In two European ATC centers, Amsterdam and Maastricht, one may observe the first applications of medium term conflict detection. In fact, their development began before the European Air Traffic Control Harmonization and Integration Program (EATCHIP). The Amsterdam conflict detection tool has been the basis for the development of the European MTCD.

The MTCD Operational Requirements Document used for this study has been developed for Eurocontrol under contract by the Dutch Aerospace Laboratory, the NLR.

High tribute should also be paid to the French ERATO development, of which conflict detection is a basic element.

Similarly, in the United States the URET tool (User Request Evaluation Tool) is being implemented as an operational trial in the Indianapolis ATC center and extended to the Memphis center.

1.3.7 Future conflict resolution tools

Beyond the conflict detection performed by MTCD, EATCHIP will develop operational requirements for a Conflict Resolution Assistance tool (CORA), with the aim of providing resolution advice to the controller.

A first example of a conflict resolution tool has already been implemented in the Maastricht UAC under the name VERA (Verification and Resolution Advice). It produces advised headings to be given by the controller to aircraft for which a conflict has been detected.

Another success story aimed at giving advice to controllers is HIPS (the Highly Active Problem Solver). Started as an EEC (Eurocontrol Experimental Center) Bretigny research product, HIPS will be used in operational trial for oceanic traffic in the United-Kingdom.

1.3.8 Historical context -> Understanding of the decision-making process

Now that we know more about the historical context, we take this information and bring it into a model to better understand the decision making process that will lead to the development of MTCD.

1.4 Description of the decision-making process leading to the implementation and development of MTCD

1.4.1 Why it is important to understand this decision-making process: modeling accident

Accident models

According to Leveson [24]:

> Accident models are used to understand past accidents, to predict and thus prevent future accidents and to understand performance monitoring data. These models assume common patterns in accidents and are as a filter in collection of data about accidents and a basis for organizing data, setting priorities and designing and implementing countermeasures.
> Finally, these models are very influential in the cause ascribed to accident in preventative measures taken.

One of the most common systemic factor that can strongly contribute to an accident is the flawed decision-making process due to a lack of communication.
Our idea in this section is to better understand this decision making process.
It is, for instance interesting to analyze the decision-making process that led to the decision to develop and implement an automated conflict detection tool: MTCD.

1.4.2 The Rasmussen model for decision-making analysis

According to Rasmussen [31], an accident path is created by several decision makers subject to local pressure and criteria. Decision-making modeling requires a study of all involved actors. There are many nested levels of decision-making and regulatory rule-making involved in risk management. We can decompose them in: government, regulators, companies, management, development and operations.
This social organization is subject to severe environmental pressure in a dynamic and competitive society.
In the following section, we apply this to the ATC social organization. We attempt to show how this social organization comes out with the decision to develop MTCD and with the functional goals.

1.4.3 The model applied to "Eurocontrol deciding MTCD and setting its functional goals"

Government = Ministers and European commission.

Environmental stressors = Public concern over delays in air transportation.

Goals and motivation = To be reelected, ask regulators and companies to increase capacity.

System functional goals = G.6

Regulators = Eurocontrol, ICAO

Environmental stressors = Government concern

Goals and motivations = Harmonize and insure safety of air traffic control activities.

System functional goals = G.5, G.7

Companies = Eurocontrol.

Environmental stressors = Regulator's concern + Changing and competitive market conditions.

Their clients: the airlines, have concern over increasing traffic, and ask for a better Air Traffic Management (ATM) system to allow an increase in capacity. They push toward a random-point-routing, or "free flight", system.

Goals and motivation = Develop a better Air Traffic Management system to meet capacity issues.

System functional goals = G.6

Development = Engineers

Environment stressors = Project success

Goals and motivations = Realistic and feasible functional goals for MTCD as a first step toward conflict resolution automated tools

System functional goals = G.2, G.3, G.4

Operations = Air Traffic Controllers

Environmental stressors = Keep their jobs, keep ATC on ground.

Goals and motivations = Insure aircraft separation, manage over-flights, make a better ATC job than pilots

System functional goals = G.1

1.4.4 Understanding of the decision-making process–> MTCD functional goals

Based on that historical context and on our understanding of the decision-making process, we now appreciate better the need for a conflict detection tool that will help controllers deal with the increase of traffic and keep their workload into safe and acceptable limits. This allows us to write in general terms the main functional goals of MTCD.

1.5 MTCD system functional goals

This section includes the high-level goals for the system.
Usually, in the early stages of a project, goals are stated in very general terms. One of the first steps in defining system requirements (section 1.10) is to refine the goals into testable and achievable high-level requirements (the assumption is made here that requirements must be measurable and testable to be called a requirement).

- G.1

  Provide a conflict detection capability to air traffic controllers for all flights in the area of operation.

- G.2

  Take over part of air traffic controllers monitoring task to keep aircraft separated.

- G.3

  Provide a planning tool to air traffic controllers.

- G.4

  Provide controllers with enough time to assess and if necessary to resolve conflict by deliberate action.

- G.5

  Be appropriate for implementation in all European Civil Aviation Community (ECAC) areas irrespective of their complexity and density.

- G.6

  Allow use of different separation criteria between aircraft when required, and increase airspace capacity.

- G.7

  Help to keep the workload of the controllers within acceptable and safe limits despite the foreseen increase of traffic and balance workload between tactical and planning controllers

1.5.1 MTCD functional goals–>Environment description, assumptions and constraints

Now that we have the functional goals issued from the historical context, we describe the environment in which MTCD is to be designed and integrated.

We also give the related environment's assumptions and constraints.

1.6 Environment

In this section, we describe the "givens" or environment in which the system being specified will function. They include the components that already exist or at least are not being designed as part of this system development.
Basically, for the design of our MTCD system, we don't really have the choice. MTCD has to fit in the more global ATM architecture defined by Eurocontrol for its future ATM system as it has been described in Eurocontrol technical reports [10].
This is indeed interesting and relevant for our safety methodology demonstration, since most of the time, new software has to be integrated in already existing complex environment.

1.6.1 Definition of our system and of its boundaries

MTCD is not a stand-alone function, but is part of a larger ATC system that includes functions MTCD has to interact with or depends upon. Air traffic control system as a whole provides an ATC environment to the controllers in which they can perform all tasks necessary to ensure an efficient and safe handling of traffic. All functions in the system must cooperate and be tuned to each other to create this environment.
The elements of the environment of MTCD are:

- The controller's automated tools:
  - the Monitoring Aids: MONA,
  - the Safety Nets,
  - the sequencing managers: AMAN and DMAN;
- The basic functions:
  - the Real Time Flight data processing and distribution function (FDPS),
  - the Environment data processing and distribution function (EDPS),
  - the Surveillance track data function,
  - the Recording data function;
- And the Human Machine Interface.

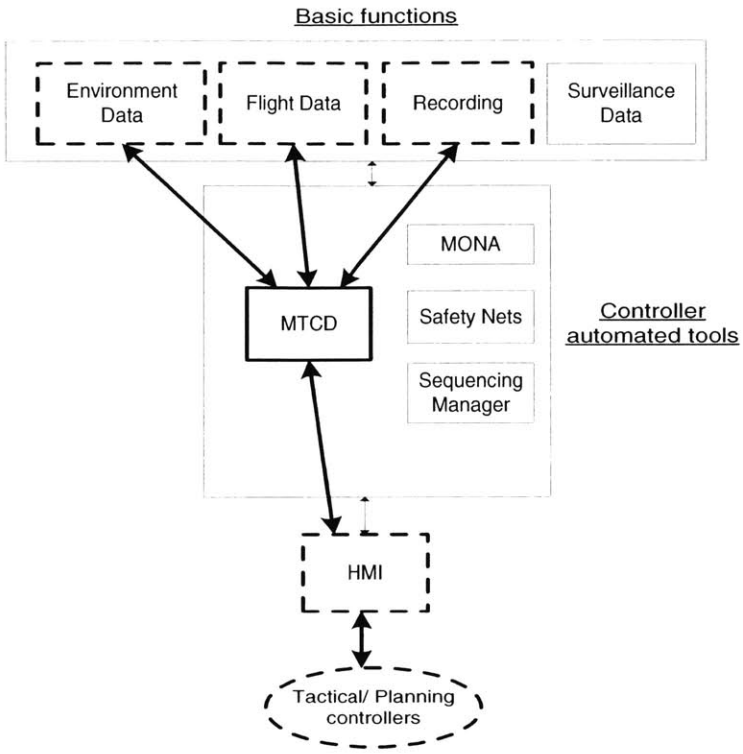

Figure 1-2: System Definition, Boundaries and Interfaces

Based on that description of the environment, we now can describe our system as being composed of:

- MTCD itself, that is the conflict detection computation,
- Its interactions with
  - The Real-Time Flight data processing and distribution function,
  - The Environment data processing and distribution function,
  - And the Recording data function;
- The Human Machine Interface as a blackbox (i.e., the flows of information through the HMI but not the physical design) and the interaction between MTCD and the HMI,
- The Planning (PC) and the Tactical controller (TC) as they use MTCD conflict data through the HMI. This part of our system is not detailed in the present study but by Daouk in a companion master's thesis [8].

1.6.2 System's Environment Description, Assumptions and Constraints

In the following sections, we describe the functions that directly interact with MTCD: FDPS, EDPS, the Recording function and the HMI. These interactions are part of our system.

We also describe the functions that have no interactions with our system but whose existence have an indirect role on MTCD.

These environment assumptions have to be traceable for the design and the implementations of the environment (HMI, FDPS, EDPS...). They are expressed with the "will" statement.

We decide so to make a difference with the requirements, which are expressed with "shall" and constraints which use "must " or "must not" statements.

The Real-Time Flight Data Processing and Distribution function

MTCD calculations are based on system trajectories of flights, flight plan data, and aircraft data. This data is provided by the Real-Time data processing and distribution function.

The trajectories can be either system or tentative. Tentative trajectories are new trajectories the controllers submit to MTCD calculations to assess if an advisory will cause any conflict.

Assumptions

- Env-As-FDPS-01

  FDPS will provide MTCD for all eligible flights at every surveillance data update (TBD) with:

  - System and tentative trajectories. The trajectory data consist of:

    * a trajectory,

      Description:

      According to the design adopted for FDPS: A trajectory is a set of points (x, y and flight level) with associated times.

      It for example includes the two important points associated to the entry/ exit times in the area of operations,

It also includes flight plan data information: areodrome of departure, destination, and planned flight levels,

  * a system/tentative indicator,
  * the phase-of-flight,

- Aircraft and flight data which consists of:
  * aircraft identification,
  * aircraft type,
  * aircraft equipment or navigational capabilities,
  * the class of flight,

    Rationale:

    For the MTCD we describe in this study, the class-of-flight is related to the size of an aircraft: heavy, B757, large, small. (This might not correspond to MTCD as it is developed in Eurocontrol).

  * the current 3D position,

    Rationale:

    Aircraft and flight data are constant for all trajectories (system or tentative) of each flight. For example, even if the controller is assessing several tentative trajectories for the same flight, the aircraft identification, or current position is the same for all these trajectories.

- A specification whether or not aircraft use parallel ATS routes

- Env-As-FDPS-02

  Current system trajectories for eligible flights will be available until the flight leaves the defined area of operation.

- Env-As-FDPS-03

  System trajectories will be built using best available flight plan data updated by last surveillance data or coordination data when surveillance data is not yet available, updated by data link, if available, and updated by latest controller inputs.
  Furthermore FDPS will inform MTCD if the flight data changes. Rationale:
  This assumption highlights the fact that trajectory data provided by FDPS comes from three main entities: the Control Flow Management Unit providing flight plan, the Radars and Data link providing current position, altimeter...

- Env-As-FDPS-04

  FDPS will inform MTCD when a system trajectory is deleted in order to end conflicts in which the system trajectory is involved.

  FDPS will notify MTCD when a system trajectory is created or modified.

  FDPS will provide MTCD with the re-calculated system trajectory after each trajectory re-calculation.

  Rationale:

  Trajectory re-calculation occurs when an aircraft deviates from its planned trajectory by more than the specified MONA deviation threshold parameter (Env-As-MONA-02).

- Env-As-FDPS-05

  FDPS will inform MTCD when a tentative trajectory is deleted (either because it is no longer valid or it becomes the system trajectory) in order to end conflicts in which the tentative trajectory is involved.

  FDPS will notify MTCD when a tentative trajectory is created or modified.

  Rationale:

  Tentative trajectories will be used in future versions of MTCD by the controllers as a "what-if' probing functionality. To resolve a conflict, the controller will submit a new or tentative trajectory for a flight to MTCD conflict detection calculation before to actually communicate advisories to the pilots. If this tentative trajectory is satisfactory for the controller, he would then make this tentative trajectory become the system trajectory for the flight. This functionality is not yet implemented, but the current version of MTCD we are working on shall support tentative trajectories. This is further detailed in our requirements (1.24).

Environment Data Processing and Distribution function

In addition to trajectory data, MTCD requires environment data. This data is provided by the Environment Data Processing and Distribution function.

Assumptions

- Env-As-EDPS-01
  - Env-As-EDPS-01.1

    The EDPS will provide MTCD with the following environment data:
    * The Area of operation, plus:
      - the default lowest usable flight level,
      - the default separation criteria,
      - the default uncertainty area parameters,
    * A list of airspaces in which different separation criteria, different uncertainty area parameters, different lowest usable flight levels apply,
    * A list of phases-of-flight for which different separation criteria, different uncertainty areas apply,
    * A list of classes-of-flight for which different separation criteria apply,
    * A list of navigational capabilities for which uncertainty areas apply,
    * The list of parallel ATS routes associated with the separation criteria to apply, item
    * A list of permanently special use airspaces,
    * A list of temporarily special use airspaces, with the time of restriction (start and end times),
    * Parallel ATS routes,
  - Env-As-EDPS-01.2

    This environment data will be set during the EDPS configuration phase by the MTCD supervisor. To configure EDPS, the MTCD supervisor shall stop MTCD.
- Env-As-EDPS-02

  The EDPS function will notify MTCD of any changes in the environment data.

- Env-As-EDPS-03

  The EDPS function will be available continuously to provide environment data to MTCD.

  Rationale:

  MTCD must not be used by controllers if the EDPS is not available, i.e. if environment data becomes obsolete. The obsolescence criteria and MTCD's behavior if EDPS fails are defined in our requirements (1.27, 1.28).

- Env-As-EDPS-04

  The EDPS function will provide MTCD with system-level parameters before it can be started, during the configuration phase.

Recording data function

This function will receive the recording data from MTCD.

Assumptions

- Env-As-Rcdg-01

  The recording data function will receive recording data identified by management from MTCD.

- Env-As-Rcdg-02

  The recording data function will not send any data to MTCD.

Constraints

- Env-Cstr-Rcdg-01

  The requirements for the recording function must not degrade the capability of MTCD to satisfy its functional and performance requirements.

HMI

The automated ATC tools: MTCD, the sequencing managers (AMAN, DMAN), the Safety Nets, and the Monitoring aids all share the same Human Machine interface. For our study we focus on the aspects of the HMI related to the use of MTCD. All interaction between the controller and MTCD is handled by the HMI. We use the Eurocontrol HMI as a baseline for our study [6].

The interaction with the controller is one of the most important issues in the ATC system. Without controller confidence, the system cannot be used efficiently.
The assumptions described in the next section are related to the part of the HMI that are not directly linked to the use of MTCD.

For our safety methodology demonstration, we only study the parts of the HMI (displays, controls) dedicated to MTCD and the interactions between MTCD and the HMI.
Furthermore, because the HMI will be used as an unique interface for all the automated tools, we need to write the following assumptions on the characteristics of the Human Machine Interface that are not related to MTCD.

Assumptions

- Env-As-HMI-01

  The baseline human machine interface will include a Radar Plan View Display (RPVD) of the airspace and of the traffic situation.

- Env-As-HMI-02

  The baseline human machine interface will include the aircraft tracks with interactive Radar Labels presenting flight data.

- Env-As-HMI-03

  The information displayed and the color of the Radar Label will be dependent upon the aircraft planning state.

- Env-As-HMI-04

  The baseline human machine interface will include the Extended Radar label providing additional flight plan data, not available within the Radar Label.

- Env-As-HMI-05

  The baseline human machine interface will include the Radar Toolbox containing a set of tools that allows the controller to change the display characteristics of the Radar Plan View Display: map, label, overlap, filtering, flight leg, zoom, height filter, speed and track (speed vector, track history).

- Env-As-HMI-06

  The baseline human machine interface will include the Sector Inbound List (SIL) displaying advanced information for aircraft planned to enter the controlled sector.

- Env-As-HMI-07

  The baseline human machine interface will include the Message-IN and the Message-OUT windows providing in-coming and out-going coordination messages exchanged with neighboring sectors.

- Env-As-HMI-08

  The HMI will include a short term conflict alert display.

- Env-As-HMI-09

  The HMI will warn controllers of any trajectory deviations detected by the Monitoring Aids (MONA).

- Env-As-HMI-10

  The HMI will provide controllers for each airspace within the area of operation with feedback on the set values of the separation criteria, uncertainty areas, lowest usable flight levels provided by the EDPS to MTCD. The HMI will also provide feedback on the parameters identified in (Env-As-EDPS-01) used by MTCD for conflict calculations.

  Rationale: EDPS parameters have a direct influence on MTCD and shall be displayed.

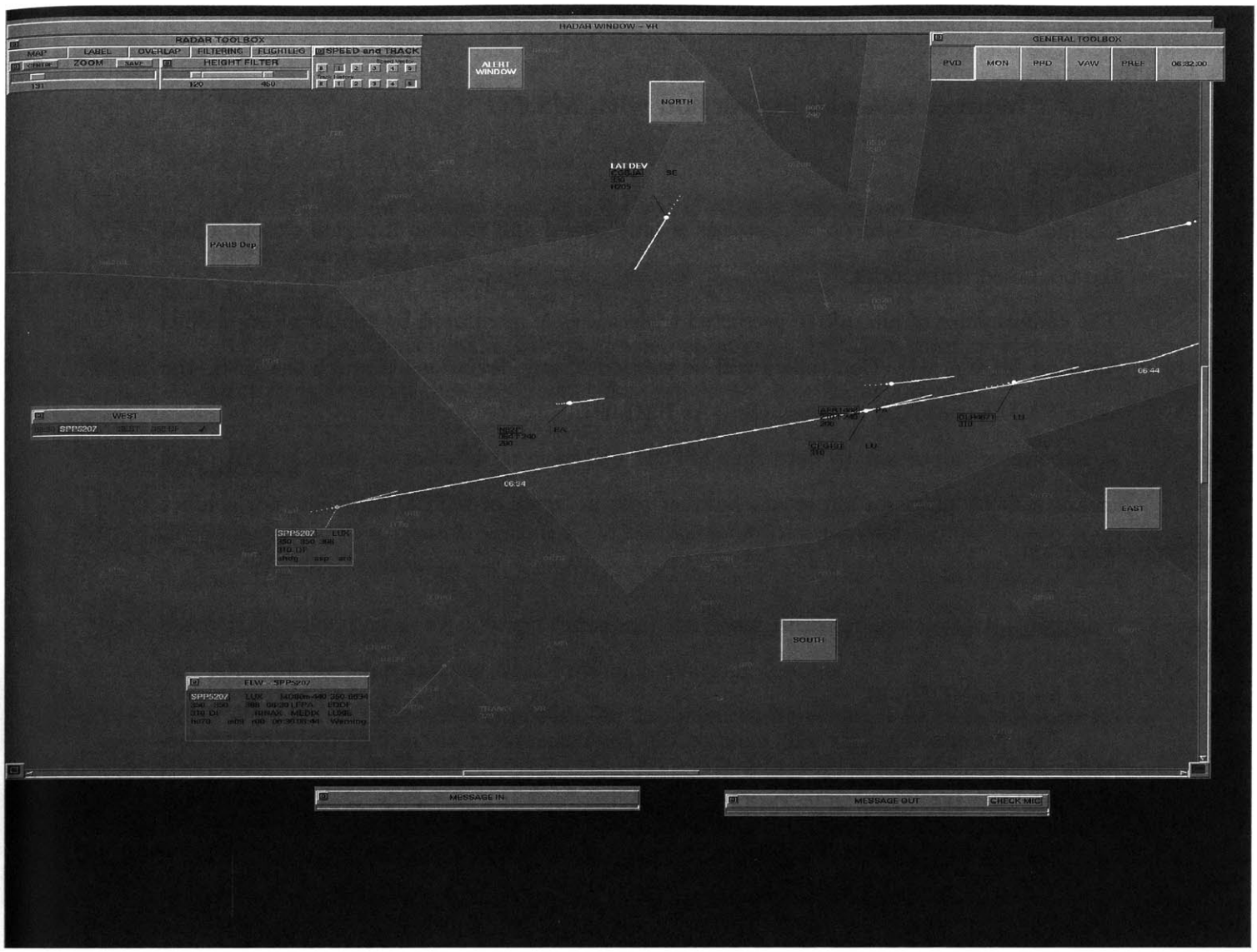

Figure 1-3: HMI picture

Constraints

- Env-Cstr-HMI-01

  The HMI (part related to MTCD) must not display MTCD conflict data in case the conflict has been detected by both MTCD and Safety nets.

  Rationale:

  Conflict alert with a typical horizon of 0..2 minutes will be covered by both the Safety Nets (Env-As-SafNet-01) and MTCD. In order not to confuse the controllers with different displays (displays for MTCD and for the Safety Nets data) for the same conflict, only the Safety Nets conflict data shall be displayed by the HMI.

1.6.3 Indirect role and interaction with MTCD

MONA

Conflict detection based on trajectories is only useful and reliable if flights closely follow the predicted trajectories.

The conformance of aircraft to predicted trajectories is monitored by the Monitoring Aids function MONA [11]. Controllers will be warned of any deviations through the HMI (the same HMI that MTCD is using).(Env-As-HMI-09).

It is however important to note that MTCD will have no interaction with MONA. This means MONA plays an important indirect role in terms of MTCD reliability for conflict detection.

Assumptions

- Env-As-MONA-01

  The Monitoring Aids will monitor the conformance of aircraft to predicted trajectories.

- Env-As-MONA-02

  MONA will either inform the controller or trigger a recalculation of the system trajectory (by FDPS) in case of a deviation.

- Env-As-MONA-03

  This function will allow some deviation from predicted trajectories before issuing a warning or invoking system trajectory re-calculation.

- Env-As-MONA-04

  Any system trajectory re-calculation caused by a MONA warning will be invisible to MTCD.

- Env-As-MONA-05

  MTCD uncertainty areas and MONA deviation thresholds, at which MONA will invoke a system trajectory re-calculation, will be compatible and tuned to each other.

Constraints

- Env-Cstr-MONA-01

  MTCD must not perform conflict detection if MONA detects aircraft deviations from the system trajectories.

  Rationale:

  If MTCD performs conflict detection when an aircraft deviation from its system trajectory is detected then the conflict data would not be reliable anymore.

Safety Nets

Conflict alert with a typical horizon of 0..2 minutes will be covered by the Safety Nets function [12].
Safety Nets is composed of 3 different functions: the Short Term conflict alert, the Minimum Safe Altitude Warning and the Area Proximity Warning function.
Safety Nets and MTCD will both cover the 0..2 minutes time frame. MTCD will have no knowledge of the existence of a Safety Nets alert and will provide independent conflict data to HMI.

Assumptions

- Env-As-SafNet-01

  The Safety Nets will provide the tactical and the planning controllers with alerts for aicraft conflicts, minimum safe altitude and minimum area proximity warnings with a typical horizon of 0..2 minutes.

Constraints

- Env-Cstr-SafNet-01

  MTCD must not interfere with the Safety Nets function.

- Env-Cstr-SafNet-02

  MTCD must have no knowledge of the existence of a Safety Nets alert tool and will provide conflict data to the HMI.

Sequencing managers

AMAN/DMAN is an Arrival/Departure sequencing manager that will help the controller to sequence aircraft for arrival/departure to/from an airport [13]. AMAN is, with the HMI, the only function identified to be later provided with conflict data.
The sequencing managers are still under development, thus they do not impose any assumptions or constraints on our system.

Assumptions

- Env-As-SMAN-01

  AMAN will be provided the same, or a subset of, the MTCD conflict data provided to the HMI.

Future functions: resolution and what-if probing

At the moment the HMI and AMAN are the only functions identified to be provided with conflict data.
In the future, other functions (e.g. conflict resolution advisory, what-if-probe) might be identified that will use MTCD conflict data.
Thus, MTCD is foreseen to be used for purposes other than detecting conflicts for system trajectories (e.g., support what-if probing, conflict resolution). This is the reason why we mention earlier tentative trajectories. A tentative trajectory is a trajectory that the controller is assessing as a potential conflict resolution. They are linked to the potential future MTCD functionality: "what-if" probing.

Assumptions

- Env-As-Other-01

  The MTCD conflict data required by future other functions (e.g. conflict resolution advisory, what-if probe) will be the same as, or a subset of, the data provided to HMI and AMAN.

1.6.4 Environment–>Preliminary Task and Hazard Analyzes

Now that we know the MTCD functional goals and the environment in which it is supposed to be built, we will perform a Preliminary Task Analysis and a Preliminary Hazard Analysis.

The Preliminary Hazard Analysis (PHA) makes sure we identify the hazards early enough in the process to be able to affect design decisions.

The Preliminary Task Analysis (PTA) makes sure that we adopt a human-centered approach when designing our system.

Based on these two preliminary analyses, we will generate the MTCD requirements and design constraints.

1.7 Preliminary Hazard analysis

A PHA is used in the early life cycle stages to identify critical system functions and broad system hazards. It has to be started early so that the information can be used in tradeoff studies and selection among design alternatives.
The PHA is divided into three steps: definition of the hazard list, fault tree analysis, and links with safety-related requirements and constraints.

To perform our PHA, we use the environment assumptions and the assumptions we made for our Preliminary Task Analysis. The process has to be iterative with the PHA being updated as more information about the design is obtained and as changes are made.

The PHA and the PTA are conducted in parallel in an iterative process. The PTA task allocations described in the next section guides us building our fault tree.

The PHA identifies new hazards induced by the introduction of new automation and requests new principles from the PTA.

The PTA provides new task allocation principles used in the PHA to check for new types of hazards induced by the introduction of the automation (MTCD).

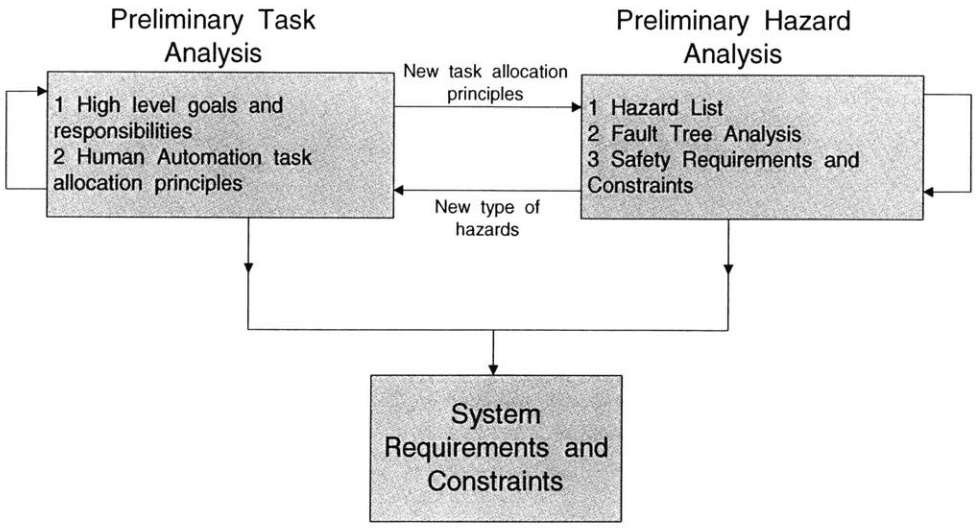

Figure 1-4: PHA/PTA iterative and parallel process

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. Here they would be related

to the loss of separation between aircraft.

Causes which can be divided into systematic, contributing and direct factors lead to 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.

We adopt the following terminology defined by Leveson in [22]:

- An Accident is an undesired and unplanned (but not necessarily unexpected) event that results in (at least) a specified level of loss.
- An Incident is an event that involves no loss (or only minor loss) but with the potential for loss under different circumstances.
- A Hazard is a state or set of conditions that, together with other conditions in the environment, will lead to an accident (loss event).
- A Failure is non-performance or inability of system or component to perform its intended function for a specified time under specified environmental conditions.

1.7.1 Hazard list

Leveson has identified the following generic hazards of the ATC activity [1]

1. A pair of controlled aircraft violates minimum separation standards.
2. A controlled airborne aircraft enters restricted airspace without authorization.
3. A controlled airborne aircraft gets too close to a fixed obstacle other than a safe point of touchdown on assigned runway.
4. A controlled airborne aircraft enters an unsafe atmospheric region.
5. A controlled aircraft and an intruder in controlled aircraft violate minimum separation standards.

According to the MTCD high-level functional goals, only the three first hazards need to be analyzed. We analyze hazard 1 for our demonstration.

[1] We only give the ATC generic hazards relevant and related to the use of MTCD. Leveson identified a total of 10 ATC generic hazards [28].

1.7.2 Fault tree analysis

Fault tree analysis is perhaps the most widely used system hazard analysis technique. We build a fault tree to identify the events that can contribute to hazard 1. The PTA is conducted in parallel and provides principles to guide the intuitive building of the fault tree. Related system requirements and constraints are identified for each leaf node in the fault tree. Thus PTA and PHA identify in parallel the major system requirements.

We adopted the fault tree representation used in [22] instead of the standard graphical representation widely used in the industry.because it increases readability and understandability. Each event is divided in contributing events or factors linked with -AND- and -OR- gates.

The OR gates are represented by vertical dashed lines and AND gates by vertical plain lines. Links are given to the system requirements.

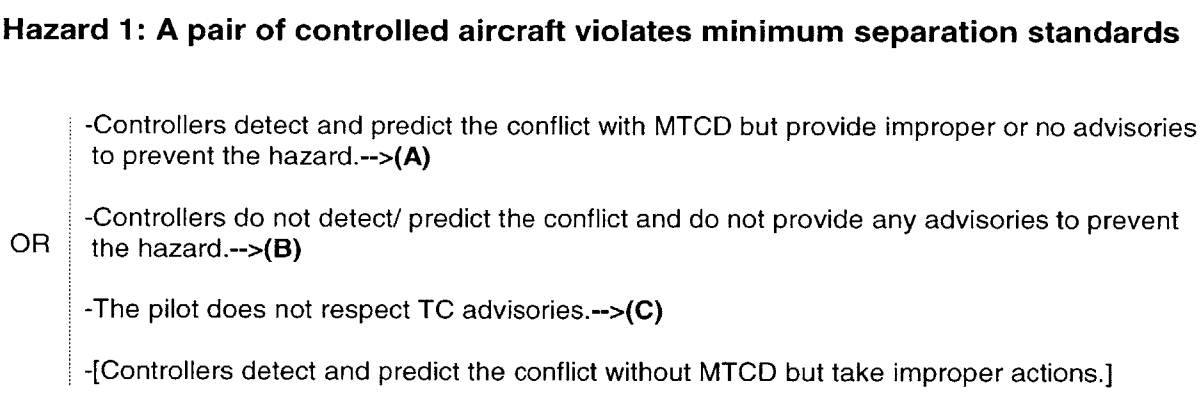

Figure 1-5: Fault tree - Hazard 1

-**(A)** Controllers detect and predict the conflict with MTCD but provide improper or no advisories to prevent the hazard.

- PC allocates the conflict to the TC who provides improper advisories to prevent the hazard.
  - -The TC provides advisories too late to the aircraft.
    - -PC allocates the conflict too late to the TC. (Op-04)
    - -TC is provided too late with conflict data.
      - -MTCD calculation takes too long. (1.27)
        - -The FDPS or EDPS provide MTCD too late with data. (Env-As-EDPS-03)(Env-As-FDPS-04)
      - -The HMI display of conflict takes too long.
    - -TC is busy dealing with another conflict.
      - -[Traffic is congested in the area of operation.]
      - -TC workload is too high and PC workload too low.(1.54)(Tr-04)
  - -The TC provides erroneous advisories to the aircraft.
    - -TC understands MTCD conflict data but takes improper actions.
      - -TC has not been trained sufficiently to use MTCD. (Tr-02)
      - -TC does not trust MTCD conflict data.
        - -TC often deals with false alarms. (1.52)(Op-05)
          - -Separation criteria provided by EDPS to MTCD is set too high. (Env-As-HMI-10)(Env-As-EDPS-01.2)(Op-05)
          - -The sector includes runways with close parallel approach procedures. (1.17)

OR

          - -The HMI conflict severity classification parameters are set too low. (1.HMI.08, 09)(Supv-04)
        - -MTCD displays too often non-critical conflicts to controllers' expectations. (1.HMI.07)
        - -TC is a senior controller and is not familiar with digital displays. (Tr-01)
      - -[TC does not respect the procedures to resolve the conflict.]
    - -TC does not understand MTCD conflict data displayed on the HMI.
      - -The MTCD HMI displays are confusing or misleading. (1.HMI.01)
      - -TC has not been trained sufficiently to use the MTCD HMI displays.(Tr-02)
      - -The Safety Nets provide TC with contradictory conflict data. (Op-07)
    - -The HMI displays provides incorrect data to the TC
      - -MTCD provides the HMI with inaccurate conflict data.(1.33)(1.34)(1.35)

-PC allocates the conflict to the TC who provides no advisories to prevent the hazard.

- -TC does not know he is in charge of that conflict. (Tr-05)
  - -One aircraft involved in the conflict is under adjacent sector's TC supervision and the other aircraft is under current sector's TC supervision. (1.07)
- -<TC is busy dealing with another conflict.>
- -<The Safety Nets and MTCD provide contradictory conflict data through the HMI and TC try to understand why.>

-PC does not allocate the conflict to the TC who thus provides no advisories. (Op-04)

-PC allocates the conflict too late to the TC. (Op-04)

-**(B)** Controllers do not detect/ predict the conflict and do not provide any advisories to prevent the hazard.

AND
- -Controllers do not detect and predict the conflict without MTCD, i.e. only based on the traffic and flight data displayed on their screen. (Tr-03)
- -Controllers do not detect and predict the conflict with MTCD.
  - -**(B1)**TC does not detect the conflict on the MTCD HMI
    - -TC does not pay attention to the MTCD conflict data displayed on the HMI
      - -<TC is busy dealing with another conflict.>
      - -[TC loses concentration.]
      - -<The HMI conflict severity classification parameters are set too low.>
      - -The TC is distracted by other displays on the HMI. (1.Cstr-HMI.04)

      OR
    - -The HMI does not display the conflict to the TC.
      - -MTCD does not send any conflict and does not send any conflict data to the HMI.
        - -MTCD calculation takes too long because of an overload of aircraft in the sector. (1.36)
        - -MTCD separation criteria are set too low. (Env-As-HMI-10)(Env-As-EDPS-01.2)
        - -<MTCD conflict calculation is inaccurate.>
        - -[MTCD conflict calculation is based on inaccurate aircraft trajectories.]

        AND
      - -The aircraft conflict has been unintentionally excluded from the HMI by the TC. (1.HMI.06)(1.09)
      - -[The HMI screen is blank or frozen]
  - -PC did not previously detect the conflict on the MTCD HMI before sector entry and did not allocate it to the TC
    - -c.f. (B1)

-**(C)** The pilot does not respect TC advisories.
- -The pilot follows ACAS advisories that are contradictory with TC advisories. (Op-09)
- -[The pilot does not understand properly TC advisories for any reason (communication problems...)]

1.8 Preliminary task analysis

This section has been written based on a master thesis by Daouk [8] She has defined what should be in a PTA and performed one for MTCD. For the PTA, we:

- Describe the Air traffic controller high-level goals,[2]
- Describe the current responsibilities the controller has to fulfill to achieve these high-level goals,
- Describe the human automation task allocation principles, and give the rationale for these principles,
- Trace these principles down to the MTCD system requirements.

1.8.1 ATCO high-level goals

The air traffic controllers high-level goals are to:

- **Op-G-01**
  Insure aircraft separation[3]
- Op-G-02
  Manage en-route flights, arrival, departure
- Op-G-03
  Manage pilot's requests (weather, conflicts, re-routing, etc...) and provide pilot with relevant information (trajectory, runway status).
- Op-G-04
  Manage adjacent controller's request(inter sector communication) and provide colleagues with relevant information(intra sector and hierarchical communication).
- Op-G-05
  Manage aircraft emergencies.
- Op-G-06
  Carry out hand-over from previous controller

[2]Terminology: to achieve a goal, to fulfill a responsibility, to perform a task.
[3]We use bold font for the controller's goals MTCD will support achieve

We can identify the MTCD functional goals that will help the controller in achieving this goal.
For instance: MTCD G.1 functional goal will help achieve controllers' Op-G-01 high-level goal. Even after the introduction of MTCD, these Operator high-level goals will remain unchanged. MTCD will only help achieve these goals.

1.8.2 Assumptions on the Tactical and planning controllers

In this section we describe the assumptions we use for the PTA and for the PHA.

- Op-As-01

  In the current Eurocontrol system, the controlling tasks are performed by two controllers, the planning (or strategical) controller and the executive (or tactical) controller.

- Op-As-02

  The high-level goals of the PC and the TC are similar and their areas of responsibilities are the same, but their areas of interests are different.

- Op-As-03

  The PC handles the pre-sector and sector entry, and works on the in-sector and sector exit areas if his/her workload allows it and/or if the TC asks for assistance.

- Op-As-04

  The TC handles the in-sector and is in direct communication with the aircraft.

- Op-As-05

  Each controller, PC and TC has an independent set of ATC tools (MTCD, MONA, Safety Nets and HMI...)

The main goal of this division of responsibility is a better management of the controllers' workload.

In the next section we identify the current operator's responsibilities to achieve the Op-G-01 high-level goal.

1.8.3 Planning and Tactical controller's responsibility list

In this section, we list the responsibilities the planning and the tactical controllers have to fulfill respectively in order to achieve their goals. We give an example with Op-G-01 and highlight (in bold font) the responsibilities that are related to the use and implementation of MTCD.

PC responsibility list to insure aircraft seperation (OP-G-01)

- PC-R-01 Detect traffic on displays.
- **PC-R-02 Predict potential conflict.**
- **PC-R-03 Assess the need for intervention.**
- PC-R-04 Allocate these problems to the TC.
- **PC-R-05 Assess the impact of the resolution on the overall traffic.**
- PC-R-06 Monitor TC's workload.
- PC-R-07 Take over TC's tasks if requested

TC responsibility list to insure aircraft seperation (OP-G-01)

- TC-R-01 Detect traffic on displays
- **TC-R-02 Predict potential conflict**
- TC-R-03.1 Receive conflicts identified and allocated by PC.
- TC-R-03.2 Receive conflicts allocated by adjacent sector's TC
- **TC-R-04 Assess the need for intervention** [4]
- TC-R-05 Ask for PC assistance, allocating him the conflict.
- TC-R-06 Resolve the conflict
- TC-R-07 Communicate advisories to the pilots

[4]This might be part of the future "What-If Probing" functionality of MTCD. What-if probing and conflict resolution might be developed later by Eurocontrol and are not taken into account in this study.

- TC-R-08 Issue advisory for flight level change.
- TC-R-09 Issue advisory for heading change.
- **TC-R-10 Assess the impact of the conflict resolution on the overall traffic**

These two lists give the responsibilities allocated to the Planning and to the Tactical controller to fulfill the high-level goal Op-G-01.

Of this list we identify the responsibilities that MTCD will help fulfill:

- For the Planning Controller: PC-R-02, PC-R-03, PC-R-05,
- For the Tactical Controller: TC-R-02, TC-R-04, TC-R-10.

We also identify the responsibilities that MTCD will not help fulfill:

- For the Planning Controller: PC-R-01, PC-R-04, PC-R-06, PC-R-07.
- For the Tactical Controller: TC-R-01, TC-R-03.1, TC-R-03.2, TC-R-05, TC-R-06, TC-R-07, TC-R-08, TC-R-09.

We will concentrate on the Planning controller's responsibilities that MTCD will help fulfill than on Tactical's controller's because MTCD is primarily a planning tool.

1.8.4 Controllers/Automation task allocation principles

In this section:

- We give a list of the task allocation principles
- We trace down these principles to the System requirements and constraints.

We use these principles in our Preliminary Hazard Analysis to guide the building of the fault trees.
These principles come from key human factors issues identified by Human Automation interaction studies.These principles also come from problems that we identified in the PHA building the fault tree for the system hazards.

Task allocation principles for MTCD to help "predict potential conflict"

To help the controller fulfill these responsibilities, we introduce automation (MTCD) in our ATC system.
The way automation is to be introduced has to respect the principles described below.

Principle 1: The high level goals of the PC and the TC should not change after the introduction of the MTCD automation. Only their responsibilities might be affected and changed by the introduction of an automated ATM system.

Thus controllers high-level goals remain unchanged: Op-G-01,-02,-03,-04,-05,-06.

Principle 2: MTCD should be able to detect all the conflicts normally assigned to the controller.
Rationale:
This PTA principle as well as the preceding PHA conduct both to the need for the same requirements. These conflicts usually are: aircraft conflict, nominal routes overlap, descent below lowest usable flight level, and special use airspace penetration.
Traceability: 1.01, 1.02, 1.03, 1.04.

Principle 3: The automated detection tool should have the ability to predict problems as far as possible into the future.

Rationale:

The automation shall compensate for the limit in human prediction, especially in the context of the foreseen tremendous increase of traffic.

Traceability: 1.01, 1.02, 1.03, 1.04, 1.05.

Principle 4: TC and PC's workload should be balanced with the introduction of MTCD and assessed to make sure it is neither too high nor too low.

Rationale:

This principle is being taken into consideration because one of the reasons 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.

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 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 boredom if they must also be used when the controller is lightly loaded, for then they reduce workload even further.

Traceability: 1.53, Tr-04.

Principle 5: The MTCD Human machine interface should not distract the controller with excessive displays and messages from his/her responsibility: to detect potential conflict.

Rationale:

If a conflict is detected by MTCD, the HMI shall avoid to distract the controller with displays that could distract the controller from his/her responsibility.

Another case is if the Safety Nets and MTCD detects a conflict in the short term (0-2minutes), then the controller might be distracted by the simultaneous MTCD and Safety Nets conflict data displayed on the HMI.

Traceability: 1.HMI.01, 1.Cstr-HMI.04, Op-07.

Principle 6 The parameters of the automation should be chosen by the human in such a way that the rate of false alarms vs. misses match the human's cognitive/workload capacity and his/her problem solving methods.

Rationale:

There is a need to give some freedom to the user (controller and MTCD supervisor) to configure MTCD behavior to minimize the number of misses and the number of false alarms. This principle is important because automation behavior as it is perceived must not affect controller's trust in automation and shall match controller's conflict detection expectations. This principle is linked to the MTCD perceived automation, scope, and configuration system requirements and to supervisor requirements..

Traceability: Supv-02, 1.29, 1.30, 1.31, 1.32, 1.48, 1.51.

Principle 7 The human should have final authority as far as the use of the prediction tool, the need for intervention and the criticality of the situation are concerned.

Rationale:

As far as MTCD is concerned, the automation does not control the aircraft in the airspace. MTCD is only a planning and information tool.

However the authority to set and maintain the MTCD configuration parameters shall be given to the human and more precisely to the MTCD supervisor only.

Furthermore, the human (MTCD supervisor) shall have the authority to set the HMI conflict severity classification parameters.

Traceability: Supv-02, 1.HMI.07, Supv-04.

Finally We could also identify the task allocation principles regarding the other controller's responsibilities that MTCD will help fulfill for goal Op-G-01.

These other responsibilities as listed in section 1.8.3: "to assess the need for intervention" and "to assess the impact of the foreseen resolution on the overall traffic" (PC-R-03, PC-R-05 and TC-R-4, TC-R-10). At this point, however, the MTCD "what-if" functionality that is supposed to help fulfill these controllers' responsibilities has not been implemented by Eurocontrol yet.

In the next section we give an overview of the MTCD system requirements that are based on the PHA and on this PTA.

1.9 Overview of the System Requirements and Constraints

We use the following classification:[5]

- MTCD,
  - General requirements and constraints,
  - Functional and performance requirements and constraints,
  - Interface requirements and constraints
    - Interactions with the FDPS,
    - Interactions with the EDPS,
    - Interactions with the HMI,
    - Interactions with the Recording function,
    - Indirect interactions with the other ATC functions.
  - Other requirements and constraints.
- The Human Machine Interface
  - General requirements and constraints,
  - Other requirements and constraints.
- Operator
  - Controller requirements and constraints,
  - MTCD Supervisor requirements and constraints,
  - Training requirements and constraints.

[5]Notations for the requirements:
MTCD: 1.**, HMI: 1.HMI.**, Controller & Supervisor: Op-**, Supv-**, Training: Tr-**

1.10 MTCD System requirements and constraints

1.10.1 Conflict types and prediction horizons

- 1.01

  MTCD shall detect within the area of operation, all aircraft conflicts in the aircraft conflict prediction horizon. The default values of this prediction horizon are 0-20 minutes.

  Rationale:

  We have been provided with no information to explain why this number has been chosen for the prediction horizon. This number is probably linked to the prediction horizon ability of the tactical controller(let's say 5-10 minutes in average) in the current ATC system (To be confirmed TBC). For traffic planning purposes, it might have been decided to augment this prediction to 20 minutes with the automation (TBC).

  Traceability: [2.2].

- 1.02

  MTCD shall detect all special use airspace penetrations within the area of operation in the special use airspace penetration prediction horizon. The default values of this prediction horizon are 0-60 minutes.

  Rationale:

  We have been provided with no information to explain why this number has been chosen for the prediction horizon. This number is probably linked to the prediction horizon ability of the controller(let's say 30 minutes in average) in the current ATC system (TBC). For planning purposes it might have been decided to augment this prediction horizon to 60 minutes with the automation (TBC).

  Traceability: [2.4].

- 1.03

  MTCD shall detect all descents below lowest usable flight level within the area of operation in the descent below lowest usable flight level prediction horizon. The default values are 0-60 minutes.

  Rationale:

  We have been provided with no information to explain why this number has been cho-

sen for the prediction horizon. It is probably linked to the prediction horizon ability of the controller (30 minutes in average TBC) in the current ATC system. For planning, it has been decided to augment this to 60 minutes with the automation (TBC).

Traceability: [2.5].

- 1.04

  MTCD shall detect all nominal routes overlaps within the area of operation in the nominal routes overlaps prediction horizon. The default values of this prediction horizon are 20-60 minutes.

  Rationale:

  We have been provided with no information to explain why this number has been chosen for the prediction horizon. This number is probably linked to the limited ability of the controller to predict nominal routes overlap accurately in the current ATC system (TBC). For planning purposes it might have been decided to provide the controller with an accurate nominal routes overlap detection ability. The default values has probably been chosen not to overlap with the aircraft conflict prediction horizon (0-20 minutes) (TBC).

  Traceability: [2.3].

- 1.05

  MTCD shall allow changes of the values of the prediction horizons during the configuration phasc.

  Rationale:

  MTCD might be provided to ATC services in different countries with different sectors (size...) and different prediction horizon for the planner position. It has thus been decided to provide the possibility of changing the prediction horizons. (1.49).

  Traceability: [2.18].

- 1.06

  It shall be possible to disable MTCD on-line by the MTCD supervisor. It shall be possible to do so without affecting the nominal use of the HMI or of the others ATC tools: the Monitoring Aids, the Sequencing Managers, the Safety Nets; or the functioning of the basic functions: the environment data processing, the flight data processing systems, the surveillance track data and recording data systems.

Rationale:

The final authority to disable MTCD is left to the MTCD supervisor. This implies the controller can not disable MTCD on-line without MTCD supervisor agreement. MTCD is not a stand-alone function, but is part of a larger ATC system that includes functions MTCD has to interact with, or depends upon. All functions in the system must cooperate and be tuned to each other to create this environment.Furthermore, to disable MTCD should not affect the proper functioning of the other functions not depending upon MTCD conflict data (i.e. MONA, the Safety Nets, the basic functions: EDPS, FDPS, and the Surveillance Track Data function).

1.10.2 Scope

- 1.07

  MTCD shall perform conflict detection for all eligible flights.

  Assumption:

  The eligible flights are all the controlled flights the Tactical controllers is or will be in charge and responsible for, at a certain time in the future.

  Rationale:

  This definition allows for example to avoid the situation with an aircraft conflict at the boundary of the area of operations with one aircraft under the supervision of the TC of the adjacent sector and the other aircraft under the supervision of the TC of the current sector.

  The definition of an eligible flight for conflict detection calculations is further detailed in our design in level 2.

  Traceability: [2.6].

- 1.08

  MTCD shall allow the automatic exclusion of flights for conflict calculation based on specific phases-of-flight, classes-of-flight, or specific airspaces. These types of exclusions shall be defined, set and maintained during the configuration phase by MTCD supervisor

  Assumption:

  This automatic exclusion of flight is set and maintained during MTCD configuration

phase.

The rationale of this requirement is given in our design in level 2.

Traceability: [2.6.5].

- 1.09

  MTCD shall allow the inclusion/exclusion of flights for conflict calculation by the controller at any time through the HMI.(1.HMI.02, 1.HMI.03)

  Rationale:

  The controller usually includes a flight at sector entry, excludes a flight at sector exit. It has been decided that the controller would have the final authority to include/exclude a flight of conflict calculation in any other cases not defined by Eurocontrol documents but that should be clearly specified (TBD).

  Traceability: [2.6.4], [2.6.6].

1.10.3 Calculation frequency

- 1.10

  Whenever a trajectory (new or re-calculated) of a flight is received, MTCD shall perform a complete conflict detection for this flight.

  Assumption:

  New or re-calculated trajectories are provided by the FDPS (Env-As-FDPS-04).

- 1.11

  Whenever a flight leaves the area of operation, MTCD shall end all existing conflicts for this flight.

  Assumption:

  The FDPS will provide MTCD with this information (i.e. when a flight leaves the area of operation) (Env-As-FDPS-01)

- 1.12

  Whenever a flight is excluded for MTCD calculations, MTCD shall end all existing conflicts for this flight.

  Assumption:

  The HMI will allow the controller to deselect a flight for MTCD calculations at any time (1.HMI.06).

- 1.13

  Whenever a flight is included or re-included by the controller for MTCD calculations, MTCD shall perform a complete conflict detection for this flight.

  Assumption:

  The HMI will allow the controller to include or re-include a flight for MTCD calculations at any time (1.HMI.06).

- 1.14

  Whenever environment data changes, MTCD shall perform a complete conflict detection calculation for the type of conflict related to the change. In particular we shalle have the following:

  - 1.14.1

    Whenever special use airspace data changes, MTCD shall perform special use airspace penetration calculations for all eligible flights.

  - 1.14.2

    Whenever lowest usable flight levels change, MTCD shall perform aircraft descent below lowest usable flight level calculations for all eligible flights.

  - 1.14.3

    Whenever separation criteria change, MTCD shall perform aircraft conflict calculations for all eligible flights.

  - 1.14.4

    Whenever separation criteria change, MTCD shall perform nominal routes overlap calculations for all eligible flights.

  Assumption:

  These changes of the environment data might be done on-line. MTCD will have no knowledge of the change until the input is sent from the EDPS. The development of the EDPS will cover how these environment data changes will be performed.(Env-As-EDPS-02)

  Rationale:

  MTCD will thus allow changes in separation criteria and might (To be demonstrated) contribute to an increase in airspace capacity.

- 1.15

  Whenever the navigational capabilities, or the phase-of-flight, or the current position of a flight changes, MTCD shall perform a complete conflict detection for this flight.

  Assumption:

  This data is updated at every surveillance data update, whose rate is (TBD), (Env-As-FDPS-01).

- 1.16

  MTCD shall perform a complete conflict detection for a flight when a pre-defined time (TBD) has passed since the last complete conflict detection for this flight has been performed.

1.10.4 Separation criteria

- 1.17

  MTCD shall allow the definition of different applicable separation criteria between flights:

  - depending on the airspaces they are within,
  - depending on the phase-of-flight,
  - depending on the class-of-flight of these flights,
  - depending on the geometry of the trajectories,
  - on parallel ATS routes.

  Rationale:

  Airspaces, for example with low radar coverage, require larger separation criteria between flights.

  According to the phases-of-flight, different separation criteria might be applied, for example for climbing aircraft, en-route flights...

  For the class-of-flight, an example is the situation where a small aircraft follows a heavy aircraft. In the current ATC system, controllers intuitively apply larger separation criteria because of the dangerous wake-vortex phenomenon.

  In the current ATC system, controllers for example intuitively apply larger separation criteria for aircraft with specific geometry of the trajectories. The well-known worst case scenario is a conflict between two aircraft heading in opposite directions.

  On parallel ATS routes the separation criteria are usually different whether aircraft follow one another or go in opposite directions.

  Assumption:(Env-As-EDPS-01)

  Traceability: [2.12.2]

- 1.18

  When no other separation criteria apply, MTCD shall use the default separation criteria applicable to the area of operation.

  Assumption:(Env-As-EDPS-01)

  Traceability: [2.12.1]

1.10.5 Uncertainty areas

- 1.19

  MTCD shall allow the definition of different applicable uncertainty areas for flights:

  - depending on the airspaces within they are,
  - depending on the phase-of-flight,
  - depending on the navigational capabilities,

  Assumption:

  Uncertainty areas, airspaces provided by the EDPS (Env-As-EDPS-01),

  Phase-of-flight, navigational capabilities provided by the FDPS (Env-As-FDPS-01).

  Traceability: [2.10.2]

- 1.20

  MTCD shall allow the use the default uncertainty areas applicable to the area of operation when no other uncertainty areas apply.

  Assumption:(Env-As-EDPS-01)

  Traceability: [2.10.1]

1.10.6 Lowest usable flight levels

- 1.21

  MTCD shall allow the definition of different applicable lowest usable flight levels depending on the airspace.

  Rationale:

  The airspaces can be the area of operation with the default lowest usable flight level, or specific airspaces with lowest usable flight levels separately defined.

  The lowest usable flight level are obviously different according to the topography of terrain or according to specific airspaces.

  Assumption: (Env-As-EDPS-01)

1.10.7 Special use airspaces

- 1.22

  MTCD shall allow the definition of permanent and of temporary special use airspaces. Conflict detection calculations shall take place normally for temporary special use airspace, i.e. as for permanent special use airspaces.

  Assumption: (Env-As-EDPS-01)

1.10.8 Tentative trajectories

- 1.23

  As well as system trajectories, MTCD shall be able to process tentative trajectories.

  Definition:

  A system trajectory is the trajectory provided by the FDPS to MTCD and that MTCD uses for conflict detection calculation.

  A tentative trajectory is a trajectory for which the controller assesses the impact of a conflict resolution using MTCD conflict detection ability before actually contacting the pilot to address advisories. This is called the "what-if probing" functionality of MTCD. This is not implemented yet, however MTCD must be able to process tentative trajectories for this future improvement.

  Assumption: (Env-As-FDPS-01)

- 1.24

  MTCD shall exclude conflict detection between the system trajectory and a tentative trajectory of the same flight, and between two tentative trajectories of the same flight.

- 1.25

  MTCD shall update conflicts in which a tentative trajectory is involved.

  Rationale:

  When the controller is assessing a tentative trajectory, the system trajectory, that might later be replaced by the tentative trajectory, will not be taken into account during this assessment process and the associated conflicts will be updated.

- 1.26

  MTCD shall perform a conflict detection calculation and send conflict data within 500 milliseconds (TBC) following an update in the FDPS or EDPS data or following a controller input i.e. inclusion, exclusion, or re-inclusion of a flight or a tentative trajectory. This delay of 500 milliseconds shall be shorter than a system predefined response time value (TBD)

  Definition:

  Response time: Between the controller input and the display of a conflict (in case of a conflict), the HMI will process the controller input, the FDPS will calculate and distribute the new system trajectory, the EDPS will provide the applicable parameters (e.g. uncertainty, separation, airspaces parameters), MTCD will perform conflict detection calculation and send the data to the HMI and the HMI will display the new conflict to the controller.

  This conflict detection loop is to be achieved within a system predefined response time value (TBD).

  Thus MTCD can achieve its system functional goal: G.4.

  Rationale:

  This requirement gives an upper limit for conflict calculation duration when the controller includes, excludes, or reincludes a flight or inputs a tentative trajectory into the system.

  The delay for conflict detection calculation shall obviously be shorter than the global predefined response time of the entire loop.

- 1.27

  If for any reasons, any of the subsystems (HMI, EDPS, FDPS) provides data to MTCD too late, such that the system predefined response time can not be met, MTCD shall however be able to support a worst-case conflict detection calculation duration up to 800 milliseconds. If it takes more than 800 milliseconds, MTCD shall send a failure message to the HMI to state the obsolescence of the conflict data.

- 1.28

  MTCD shall warn the controller and the MTCD supervisor of total losses, partial losses, and corruptions in the provision of conflict data.

Assumption:

The controller and the MTCD supervisor will be warned through the HMI. The losses of data might for example occur when either the EDPS or the FDPS data become obsolete.

1.10.9 Performance requirements

Conflict detection categorization

- 1.29

  MTCD shall detect at least 95 percent of all category 1 conflict.

  Definition:

  By definition, for a category 1, a notification is necessary if without controller interaction, the possible conflict situation will almost certainly evolve into a real conflict.

  Rationale:

  This requirement is important and shall be written because it comes from an important human/automation interaction principle : automation behavior as it is perceived must not affect controller's trust in automation and shall match controller's conflict detection expectations.

  However this definition, which comes from the official Eurocontrol requirements document for MTCD [10], is obviously too vague: a "necessary" notification, will "almost certainly" evolve. There is a need to specify under which circumstances a conflict situation will almost certainly evolve into a real conflict.

  Our rationale should also explain why the number 95 has been decided for the percentage of conflicts to be detected.

  Similar comments are true for the three following requirements.

- 1.30

  MTCD shall detect at least 80 percent of all category 2 conflict.

  Definition:

  By definition, for a category 2 a notification is desirable if without controller interaction, the possible conflict situation will likely evolve into a real conflict and the conflict is likely to be severe.

- 1.31

  MTCD shall give conflict notifications for category 3 conflicts in not more than 10 percent of all conflict notifications.

  Definition:

  By definition, for a category 3 a notification is unwanted if a possible conflict situaion is not likely to evolve into a real conflict, or the conflict will not be severe, orthe controller may be distracted by the notification.

- 1.32

  MTCD must not give any notifications for category 4 conflict.

  Definition:

  By definition, for a category 4 a notification is void, if errors elsewhere in the ATM system produced a phantom conflict that does not exist in reality.

Calculation Accuracy

- 1.33

  In determining times that are part of conflict data, MTCD shall achieve a calculation accuracy of better than +/- 1 sec.

- 1.34

  In determining positions that are part of conflict data, MTCD shall achieve a calculation accuracy of better than +/- 0.1NM.

- 1.35

  In determining altitudes that are part of conflict data, MTCD shall achieve a calculation accuracy of better than +/- 50ft.

Rationale:

The calculation accuracy marks the maximum inaccuracy that MTCD will make in its calculations. Inaccuracies will result from estimations made in the algorithms, and from computer rounding. The calculation accuracy does not include inaccuracies in the input. The rationale for these required numbers: 1sec, 0.1NM, 50ft is not documented by Eurocontrol and shall be specified.

Capacity

- 1.36

  MTCD shall be able to simultaneously process a number of trajectories that is equivalent to at least 150 percent of the maximum traffic load (TBD).

  Rationale:

  Since the developers need a number to define the maximum traffic load, the maximum traffic load for all sectors where MTCD will be used, shall be defined (TBD).

  The capacity needed for a proper functioning of MTCD concerns the number of trajectories that can be served at any one time, while maintaining an acceptable calculation speed and response time.

1.10.10 Interaction MTCD/ FDPS

FDPS–>MTCD

The interaction is defined in our environment assumptions on the Flight Data Processing and Distribution System (FDPS) in section 1.6.2.

MTCD requirements based on the interaction with the FDPS

- 1.37

  MTCD shall make use of the trajectory, aircraft, and flight data provided by the FDPS to perform conflict detection calculations. In particular MTCD shall make use of the current position of a flight to perform conflict detection calculationsm (Env-As-FDPS-01).

MTCD–>FDPS

- 1.Cstr.1

  MTCD must not send any data to the FDPS.

  Rationale:

  Since MTCD is not a stand-alone function using the FDPS, but part of a large ATM architecture, it shall use the FDPS passively, i.e. without affecting the FDPS behavior.

1.10.11 Interaction MTCD/ EDPS

EDPS–>MTCD

The interaction is defined in our environment assumptions on the Environment Data Processing and Distribution System (EDPS) in section 1.6.2.

MTCD requirements based on the interaction with the EDPS

- 1.38

  MTCD shall make use of the environment data provided by the EDPS to perform conflict detection calculations: airspaces, separation criteria, uncertainty areas, lowest usable flight levels, parallel ATS routes.

- 1.39

  MTCD shall use any changes in the environment data at the first conflict detection re-calculation after the data is received.

MTCD–>EDPS

- 1.Cstr.2

  MTCD must not send any data to the EDPS

  Rationale:

  Since MTCD is not a stand-alone function using the EDPS, but part of a large ATM architecture, it shall use the EDPS passively, i.e. without affecting the EDPS behavior.

1.10.12 Interaction MTCD/ Recording function

Recording function–>MTCD

The interaction is defined in our environment assumptions on the Recording data function in section 1.6.2.

MTCD requirements based on the interaction with the Recording function

- 1.Cstr.3

  MTCD functional behavior must not be affected by the Recording function.

MTCD–>Recording function

- 1.40

  MTCD shall send recording data identified by management during MTCD operations and use by controllers to the Recording function.

- 1.41

  The data to be recorded shall be later identified and MTCD shall at least consist of all MTCD outgoing data sent to the Recording function:

  – conflict data,

  – acknowledgments of flight inclusions,

– acknowldegments of flight exclusions.

1.10.13 Interactions HMI/MTCD

MTCD requirements based on the interaction with the HMI

- 1.42

  MTCD shall be started/ stopped by the HMI.

- 1.43

  MTCD shall allow the exclusion, inclusion, re-inclusion of a flight from the HMI.

  Rationale:

  The interface from HMI to MTCD shall be limited to exclusions, inclusions and re-inclusions of individual flights decided by the controller.

  Individual flights to be excluded, included or re-included from conflict detection calculations shall be passed by the HMI to MTCD

- 1.44

  MTCD shall be stopped and switched to a configuration mode for the configuration phase through the HMI by the MTCD supervisor. The configuration parameters shall always be received off-line by MTCD through the HMI.

  Traceability: [2.18.2]

MTCD->HMI

- 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.

  Assumption:

  As described in the requirement (1.26), conflict detection calculation shall be performed nominally within 500 milliseconds (TBC) (or in a worst-case scenario (1.27): within 800 milliseconds) following an update in the FDPS or in the EDPS data provided to MTCD or following a controller input through the HMI.

- 1.46

  MTCD shall send acknowledgements to the HMI for controller's flight exclusion, inclusion and re-inclusion actions.

- 1.47

  On controller's request through the HMI, MTCD shall provide the HMI with feedback on the current configuration parameters values. The configuration parameters are listed in (1.49).

1.10.14 Configuration

- 1.48

  MTCD shall be configurable by the MTCD supervisor during a configuration phase. Configuration parameters shall be set and maintained during this phase.

  Assumption:

  With the following HMI requirement: (1.Cstr-HMI.02).

- 1.49

  MTCD shall allow off-line settings and changes of the following configuration parameters during the configuration phase by the MTCD supervisor:

  - prediction horizons,
  - phases-of-flight to be excluded,
  - classes-of-flight to be excluded,
  - airspaces to be excluded,
  - maximum time allowed between detection and first infringement,
  - maximum time between two complete conflict detections of the same flight,

  Rationale:

  These parameters need to be configurable according to the differences between airspaces where MTCD is to be set up.

  Controllers are also given the opportunity to choose when to be alerted of any conflict. They can ask to be alerted sometimes later than when MTCD first detected the conflict but not later that the maximum time allowed between detection and first

infringement. For example MTCD could detect a conflict between two aircraft in 15 minutes (the time of first infringement), 15 minutes being within MTCD's prediction horizon of 0-20 minutes. If the maximum time allowed between detection and first infringement is 10 minutes, then the HMI will display this conflict only in 5 minutes. In congested areas, this can avoid the situation where the controllers get confused by too many conflicts.

The maximum time between two complete conflict detections of the same flight is to be set according to the different surveillance update rates of the different ATC centers where MTCD is to be set up.

See also level 2, [2.6.5].

- 1.50

  MTCD shall be configurable for sector specific procedures.

  Rationale:

  For example: MTCD shall be configurable for parallel runways and parallel approach paths violating nominal longitudinal separation standards.

1.10.15 Trust, perceived automation

- 1.51

  MTCD conflict detection ability shall be assessed during real-time simulations with controllers to make sure the conflicts detected by MTCD meet controller's conflict detection expectations and do not affect controller's trust in automation.

  Rationale:

  These simulations to assess MTCD behavior, i.e. the automation as it is perceived by the controllers, shall help reduce the rate of false alarms and misses for conflict detection.

  The way these simulations should be conducted is not covered in this study.

1.10.16 Indirect interaction with the other functions

- 1.52

  MTCD shall later provide AMAN with conflict data and shall be able with minimal changes to provide conflict data to other functions than HMI and AMAN.

Rationale:

At that time, we know that AMAN will be provided by conflict data and we also know that future conflict resolutions tool might also be provided with conflict data. Even if this requirement is not very explicit, it highlights the important fact that these future functions must not affect the way conflict detection calculations have been foreseen in MTCD.

MTCD will be based on the assumption that the data required by the future other functions will be the same as, or a subset from, the data provided to HMI and AMAN.

1.10.17 Workload

- 1.53

  Before deployment, MTCD shall prove in real-time simulations that it actually decreases tactical controller workload, and allocates planning controller more workload than before.

  Rationale:

  These simulations to assess MTCD impact on controller's workload shall verify and validate the balance of workload between the TC and the PC. <G.7>

  The way these simulations should be conducted is not covered in this study.

1.11 HMI REQUIREMENTS

We remind that in our system definition, we study the Human Machine Interface related to MTCD conflict detection. The baseline of the HMI, upon which our study is based, comes from Eurocontrol and has been developed to support all the controllers' tools (i.e. not only MTCD): MONA, AMAN, DMAN, the Safety Nets.
According to our system definition, boundaries and interfaces (section 1.6.1), we study the HMI viewed as a blackbox (i.e., the flows of information through the HMI, but not the physical design) and the interaction between MTCD and the HMI.

1.11.1 General

- 1.HMI.01

  HMI shall handle all the interactions between the controllers and MTCD and between the MTCD supervisor and MTCD.

  HMI shall cover all requirements with respect to conflict display and shall be responsible for distribution of conflicts to controllers.

  Rationale:

  The controllers use MTCD during nominal air traffic control operations. The MTCD supervisor uses MTCD during MTCD configuration phase.

  The display of conflict detection data shall be achieved in a clear and unambiguous way, with no excessive optional information...i.e. according to Human Factor principles not further analyzed in this study.

- 1.HMI.02

  The HMI shall allow MTCD supervisor to start/stop MTCD.

- 1.Cstr-HMI.01

  The HMI must not allow the controller to start/stop MTCD.

  Rationale:

  The authority to start/stop MTCD is given to the MTCD supervisor only.

- 1.HMI.03

  The HMI shall allow the MTCD supervisor to set, change the MTCD configuration parameters during the MTCD configuration phase.

- 1.Cstr-HMI.02

  The HMI must not allow the controller to set, or change the MTCD configuration parameters during nominal operation.

  Rationale:

  The authority to set, and change the MTCD configuration parameters is given to the MTCD supervisor only.

- 1.HMI.04

  The HMI shall provide the controller and the MTCD supervisor with proper feedback about the values of the configuration parameters.

- 1.HMI.05

  The HMI shall provide appropriate feedback on the state of the automation: either operational, configuration or off state.

- 1.HMI.06

  The HMI shall allow the controller to exclude, include, re-include flights from MTCD conflict detection calculations at any time.

- 1.Cstr-HMI.03

  Conflict display acknowledgement, and demand for extra information, must not be visible to MTCD and must not imply changes in MTCD operational behavior.

- 1.Cstr-HMI.04

  HMI must not allow optional displays when a conflict has been detected.

  Rationale:

  Optional displays are distracting and are undesirable when a conflict situation has been detected by MTCD.

1.11.2 Conflict severity classification

- 1.HMI.07

  The HMI shall support conflict severity classification.

  Rationale:

  The conflict severity classification helps improve controllers trust in automation and reduce the number of false alerts or the number of minor importance conflict displayed.

  In the design of the HMI (not covered in our methodology), it could be based on two main parameters: the time to conflict and the severity of the loss of separation between two aircraft.

  We could for example scale the severity of a conflict from 1 to 3 with a table scaling on one axis the time to conflict and on the other axis the distance between two aircraft (for aircraft conflict or nominal routes overlap) or between a trajectory and a special use airspace (for a special use airspace penetration) or a trajectory and a lowest usable flight level (for a descent below lowest usable flight level).

- 1.HMI.08

  The HMI shall allow changes in conflict severity classification parameters by the MTCD supervisor.

- 1.Cstr-HMI.05

  The HMI must not allow changes in conflict severity classification parameters by the controllers.

- 1.HMI.09

  The HMI shall provide the controller and the MTCD supervisor with proper feedback about the value of the conflict severity classification parameters.

1.11.3 Time

- 1.HMI.10

  The HMI shall process any controller input within a predefined time delay (TBD). This delay shall be shorter than a predefined response time value (TBD).
  Furthermore the HMI shall display the conflict data provided by MTCD within a predefined time delay (TBD). This delay shall also be shorter than a predefined system response time value.
  Rationale:
  Between the controller input and the display of a conflict (in case of a conflict), the HMI will process the controller input, the FDPS will calculate and distribute the new system trajectory, the EDPS will provide the applicable environment parameters (e.g. uncertainty, separation, airspaces parameters), MTCD will perform conflict detection calculation and send the data to the HMI and the HMI will display the new conflict to the controller.
  This conflict detection loop is to be achieved within a system predefined response time value (TBD).

  This requirement gives an upper limit for controller input processing delay and for conflict data display delay for the HMI.
  These delays for the HMI shall obviously be shorter than the global predefined response time of the entire conflict detection loop.

1.12 Operator REQUIREMENTS

This section contains assumptions, requirements and constraints involving air traffic controllers behavior. This information is used in the design of the HMI, the MTCD logic, controllers tasks and procedures, MTCD user manuals, and training plans and programs.

1.12.1 Controller requirements

- Op-01

  PC shall use MTCD during normal operations to plan traffic based on the MTCD conflict data.

- Op-02

  Controllers shall include an aircraft at the entry of the sector and exclude an aircraft at the exit of the sector for MTCD calculations.

- Op-03

  Controllers shall re-include an aircraft unintentionally deleted and shall exclude an aircraft unintentionally included for MTCD calculations.

- Op-04

  PC shall notify TC early enough of the existence and persistence of a conflict so that TC can resolve the conflict.

  Rationale:

  The PC shall notify the TC as soon as the conflict is detected by MTCD so that proper actions is taken to prevent the hazard. If the TC is too busy, the PC must not distract the TC with notifications made too early.

- Op-05

  Controllers shall ignore MTCD conflict data if incorrect or inconvenient behavior (e.g. high rate of false alarms) is observed.

  Rationale:

  The MTCD supervisor will check the MTCD configuration parameters and the environment data provided by the EDPS to MTCD (separation criteria).

- Op-06

  The controller shall address conflicts detected by MTCD in a criticality-based order.

  Rationale:

  The controller must not address conflicts detected by MTCD only on a time-based order.

- Op-07

  Controller shall use the Safety Nets conflict data rather than MTCD in case of contradictory data is displayed for a conflict detected in the 0-2minutes prediction horizon. Controllers shall use MTCD as a planning tool.

  Rationale:

  Controllers will not use MTCD as a conflict alert tool. They must not rely on the automation to detect and predict conflicts. They should be able to keep a critical view on the conflict data displayed on the HMI. Thus, controllers shall be trained to

- Op-08

  The controller must not use MTCD conflict data if MONA detects aircraft deviations from the system trajectories.

  Rationale:

  If MONA detects a deviation, MTCD conflict data is based on incorrect system trajectories and is not reliable anymore.

- Op-09

  Using MTCD, controllers must not interfere with on-board conflict detection and resolution instruments.

  Rationale:

  We have no information how on-board conflict detection and resolution tools work (e.g. ACAS). We can thus imagine a case where ground (MTCD) and on-board (ACAS) conflict detection tools provide contradictory information to the pilot and to the controllers. In that case, as specified by the ICAO procedures (TBC), pilot has to ignore ground advisories.

1.12.2 MTCD supervisor requirements

- Supv-01

  The MTCD supervisor shall be responsible for starting and stopping MTCD.

- Supv-02

  The MTCD supervisor shall be responsible for setting and maintaining the MTCD configuration parameters during the configuration phase.

- Supv-03

  The MTCD supervisor shall stop MTCD before to configure or to change configuration parameters.

- Supv-04

  The MTCD supervisor shall be responsible for setting and maintaining the HMI conflict severity classification parameters.

- Supv-05

  The MTCD supervisor shall stop the HMI before to configure or to change the HMI conflict severity classification parameters.

- Supv-06

  The MTCD supervisor shall identify the data to be recorded and sent from MTCD to the recording function.

1.12.3 Training requirements

- Tr-01

  Controllers shall be trained to use the new procedures associated to the introduction of the EATCHIP Medium Term Conflict Detection Tool automated tools in a simulated environment.

  Rationale:

  This training shall allow a smooth transtion to the new digital and stripless environment and shall be done before controllers are trained on using MTCD.

- Tr-02

  Controllers shall be trained to use MTCD, and the MTCD HMI.

Rationale:

Training shall assess and improve TC and PC ability to read conflict data displayed on the HMI and to assess the need for intervention.

- Tr-03

  Training shall assess controller's ability to detect and predict conflicts without MTCD long after its introduction.

  Rationale:

  Because of natural reliance on automation, controllers shall be trained to keep their ability to detect and predict conflict without MTCD.

- Tr-04

  Training shall assess PC and TC's workload using MTCD.

  Rationale:

  This training shall verify that requirement 1.53 is fulfilled.

- Tr-05

  Training shall also assess TC awareness of responsibility for any conflict allocated to him by the PC or by the adjacent sector's TC.

  Rationale:

  Controllers shall be trained for the new digital inter-sector hand-over procedures.

1.13 MTCD system limitations

- L.1

  MTCD is not a conflict alert tool.

  Rationale:

  In case of a conflict identified by both the Safety Nets and the MTCD, MTCD conflict data shall be ignored.

  Traceability: Env-Cstr-SafNet-01, Op-07

- L.2

  MTCD provides support, the responsibility of separating aircraft remains, at all times with the controller

- L.3

  MTCD is not a conflict resolution tool.

  Rationale:

  As stated earlier, our study does not cover the future "What-if probing functionality of MTCD. It does not cover either the future resolution automated tools to be implemented based on MTCD conflict data.

Chapter 2

Level 2, System design principles

2.1 What is level 2 in our Intent Specifications?

The second level of the specification contains the basic system scientific and engineering design principles needed to achieve the behavior specified in the top level. It answers the question "why" for the design decisions and describes any basic principles or assumptions upon which the system design depends.
It also describes how the requirements above will be achieved and how the constraints will be enforced. Information at this level may be specified using English or other types of engineering and mathematical notations such as differential equations. The notations should however be familiar to and commonly used by engineers.
Finally, principles may reflect tradeoffs between higher-level goals and constraints.

In the first section of this chapter, we give the general definitions we use for the different types of conflicts to be detected by MTCD. These definitions are derived from Eurocontrol work on MTCD [33], [32]. For the purpose of our demonstration we focus in this section and in the following on the aircraft conflict and the nominal routes overlap conflict types.

In the second section, we describe in more details the system design principles. We link these principles with the system requirements described in the level 1 and show how they achieve these requirements.

In the third and last section, we highlight the tradeoffs and the rationale for the design decisions that have been made for the system and for the detection calculation algorithm [9]. The algorithm has been described in the technical document [17].

2.2 General definitions: types of conflict, eligible flights

MTCD detects four types of conflict: aircraft conflict, nominal routes overlap, special use airspace penetration, and descent below lowest usable flight level. The detection horizons for the different types of conflicts are typically 0..20 minutes for aircraft conflict, 20..60 minutes for nominal routes overlap, and 0..60 minutes for special use airspace penetration and descent below lowest usable flight level.

In this section we give the general definitions adopted by Eurocontrol in [32] and that we use for the design of MTCD.:

- Definition of a conflict (2.1),
- Definition of an aircraft conflict (2.2),
- Definition of a nominal routes overlap (2.3),
- Definition of a special use airspace penetration (2.4),
- Definition of a descent below lowest usable flight level (2.5),
- and Definition of an eligible flight (2.6).

These definitions will help understand the design principles we detail in section 2.3 to model the different types of conflict, identified in level 1, that MTCD has to detect.

2.2.1 Definition of a conflict

By definition:

- The probable positions of a flight at a certain moment in time are all positions within the uncertainty areas of the flight.
- The uncertainty area is an airspace buffer around a system trajectory to allow for deviations of the aircraft from the system trajectory.
- Separation is the situation in which two aircraft do not violate the separation criteria.

- The separation criteria is the required minimum vertical, lateral and longitudinal distances between two aircraft in order not to be involved in an aircraft conflict or a nominal routes overlap.

The definition we adopt for a conflict is:

[2.1]

A conflict is defined as a state in which the closest distance between the probable positions of an aircraft and a specific object is less than a minimum required legal separation plus a buffer.

Rationale:

Such a specific object can either be the probable positions of another aircraft, a special use airspace, or the area below the lowest usable flight level of an airspace.

[2.1.1]

Once a conflict is detected, the conflict data, as well as an associated severity indicator, is passed by MTCD to the HMI for display. We however are not interested in how HMI actually displays it. (This is not part of our system).

Traceability: 1.HMI.01, 1.HMI.07

The following sections give the definitions we adopt for the design of the four conflict types and the data associated with each type of conflict.

2.2.2 Aircraft conflict

[2.2]

Two aircraft are involved in an aircraft conflict if the distance between any position within the uncertainty area of one aircraft at a certain time, and any position within the uncertainty area of the other aircraft at at that same time, will be less than the required separation criteria.

Rationale:

The position of an aircraft in the future is not known for sure. We assume that by modeling the degree to which the trajectory path will be followed, in terms of uncertainty parameters, we can construct a reliable MTCD tool to predict aicraft conflict.

Traceability: 1.01

[2.2.1]
Once an aircraft conflict is detected, the aircraft conflict data is passed by MTCD to the HMI for display.
Traceability: 1.45

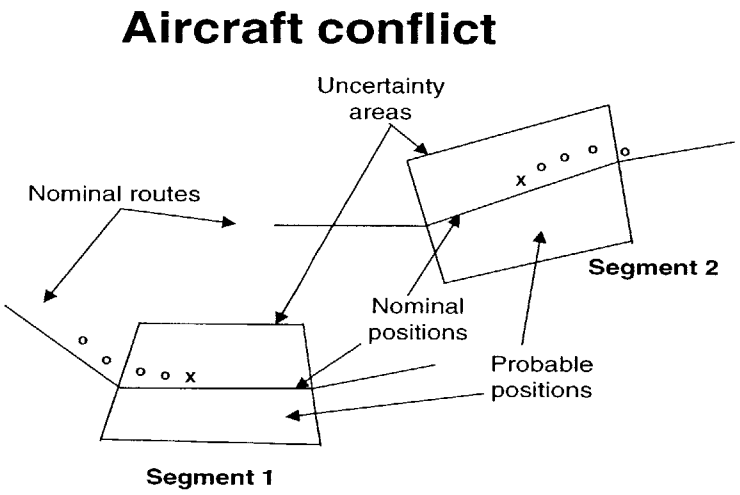

Figure 2-1: Aircraft conflict

[2.2.2]
The aircraft conflict data consists of:

- system trajectory identifications (2 aircraft),
- time of first loss of separation between aircraft probable positions of both aircraft at this time, plus worst probable positions of both aircraft, plus nominal positions of both aircraft at this time,
- time of minimum distance between aircraft probable positions, plus worst probable positions of both aircraft at this time, plus nominal positions of both aircraft at this time.
- time of last loss of separation between aircraft probable positions, plus worst probable positions of both aircraft, plus nominal positions of both aircraft at this time.

Rationale:
This the data the HMI uses to generate the displays of the conflict between two aircraft.

Using the following definition:
The worst probable position is the probable position of an aircraft at the time of a conflict for which the separation criteria are violated most.

2.2.3 Nominal routes overlap

[2.3]

Two aircraft are involved in a nominal routes overlap if the distance between the nominal position of one aircraft at a certain moment in time, and the nominal position of the other aircraft at the same moment, will be less than the required separation criteria.

Rationale:

There are no considerations of uncertainty of the positions of an aircraft for a nominal routes overlap because it is detected for traffic planning purpose rather than to keep immediate separation between aircraft. This is also the reason why the prediction horizon of nominal routes overlap is longer (20-60 minutes) than for an aircraft conflict.

Traceability: 1.04

Using the following definition:

A nominal route is the route of an aircraft as defined by its system trajectory.

[2.3.1]

Once a nominal routes overlap is detected, the nominal routes overlap data is passed by MTCD to the HMI for display (Int-HMI-05.2):

Traceability: 1.45

[2.3.2]

The nominal routes overlap data consists of:

- system trajectory identification,
- time of first infringement of separation criteria between aircraft nominal positions, plus nominal position of both aircraft at this time,
- time of minimum distance between aircraft nominal positions, plus nominal positions of both aircraft at this time.

Rationale:

This the data the HMI uses to generate the displays of the nominal routes overlap between two trajectories. We however are not interested in how HMI actually realizes it. (This is not part of our system).

2.2.4 Special use airspace penetration

[2.4]
An aircraft is involved in a special use airspace penetration if any probable position of the aircraft at a certain moment in time is within an airspace that is declared restricted at that same moment in time.
Assumption: The data concerning Special Use Airspaces is derived from EDPS.
Traceability: 1.02

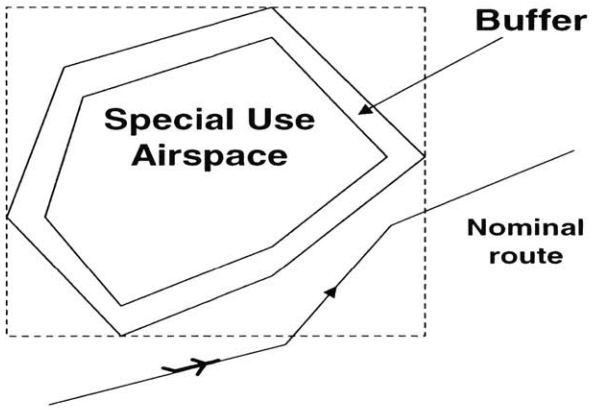

Figure 2-2: Special Use Airspace Penetration

[2.4.1]
Once a special use airspace penetration is detected, the special use airspace penetration data is passed by MTCD to the HMI for display.
Traceability: 1.45

[2.4.2]
Special use airspace penetration data consists of:

- system trajectory identification,
- special use airspace identification,
- time of first loss of separation between aircraft probable positions and special use airspace and worst probable position of aircraft at this time, plus nominal position of aircraft at this time,
- time of minimum distance between nominal route of aircraft and special use airspace, plus nominal position of aircraft at this time.

Rationale:
This the data the HMI uses to generate the displays of the special use airspace penetration between a flight and a special use airspace. We however are not interested in how HMI actually realizes it. (This is not part of our system).

2.2.5 Descent below lowest usable flight level

[2.5]
An aircraft is involved in a descent below lowest usable flight level if the minimum level at any probable position of the aircraft in an airspace is less than the lowest usable flight level defined for that airspace.
Traceability: 1.03
We use the following definitions:

- A flight level is a surface of constant atmospheric pressure which is related to a specific pressure datum, 1013.2 hectopascals (hPa), which is separated from other surfaces by specific pressure intervals.
- The lowest usable flight level is the minimum level aircraft are allowed to use within a certain airspace.
- The data concerning the lowest usable flight level of an airspace is made available by the EDPS.

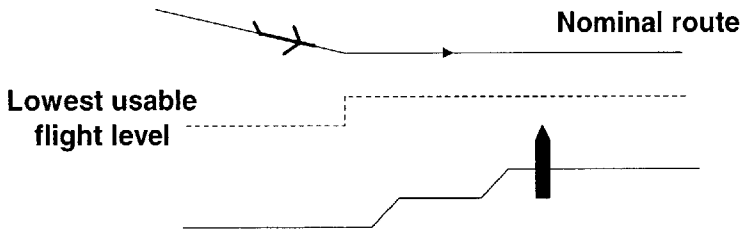

Figure 2-3: Descent Below Lowest Usable Flight Level

[2.5.1]
Once a descent below lowest usable flight level is detected, the descent below lowest usable flight level data is passed by MTCD to the HMI for display:
Traceability: 1.45

[2.5.2]
Descent below lowest usable flight level data consists of:

- trajectory identification,
- airspace identification,
- time of first descent below lowest usable flight level, plus worst probable position of aircraft at this time, plus nominal position of aircraft at this time.

Rationale:
This the data the HMI uses to generate the displays of the special use airspace penetration between a flight and a special use airspace. We however are not interested in how HMI actually realizes it. (This is not part of our system).

2.2.6 Eligible flights for conflict detection calculations

An important step in our design is to specify in our design what are the flights eligible for conflict detection calculations.

[2.6] By definition, a flight is eligible for MTCD calculations if all the following are true:

- [2.6.1]

  A system trajectory is available, and

- [2.6.2]

  The flight will be under control of one, or more, controllers of the area of operation at some moment in time,

- [2.6.3]

  The flight's time to entry to the AOO is less than a pre-defined configuration parameter.

Rationale:
The pre-defined configuration parameter is related to the size of the area of operation and to the prediction horizon of the four types of conflict. It should logically be 60 minutes that corresponds to the upper limit of MTCD conflict detection prediction horizon. (TBC)
Traceability: 1.07

- [2.6.4]

  A flight is no longer eligible for MTCD calculations if it leaves the area of operation, i.e. the airspace in which MTCD operates.

  Rationale:

  The controller is not responsible of a flight if it leaves the area of operation so there is no need to keep performing conflict detection calculation for it.

- [2.6.5]

  Specific flights can be automatically excluded from MTCD conflict detection to avoid conflict notifications and these exclusions can be based on:

  - [2.6.5.1]the phases of flight,
  - [2.6.5.2]the classes of flight,
  - [2.6.5.3]the airspaces.

Rationale:

This ability to automatically exclude specific flights is given to air traffic control centres with specific operational procedures. For example a controller in an airport sector might not be interested in conflict notifications for aircraft flying over an airport in cruise phase-of-flight. He/she thus would be given the opportunity to exclude any flight in cruise phase-of-flight. This automatic exclusion of flights from conflict detection is to be set and maintained by management through the HMI during configuration.(2.18)

Traceability: 1.08

- [2.6.6]

  Individual flights can be manually deselected (or reselected) by the ATC controller team using the HMI, so that the flights are excluded (or re-included) from MTCD conflict detection.

  Rationale:

  This design principle gives controllers the ability to deselect flights from conflict detection calculations in a less complicated manner than the contraining automatic exclusion of flight, set during the MTCD configuration phase.(2.18)

  After that a flight is manually deselected (or reselected) by a controller through the HMI, MTCD sends an acknowledgement message to the HMI to confirm the flight has been excluded (reincluded) from MTCD conflict calculations.

  Traceability: 1.09, 1.43, 1.HMI.06, 1.46.

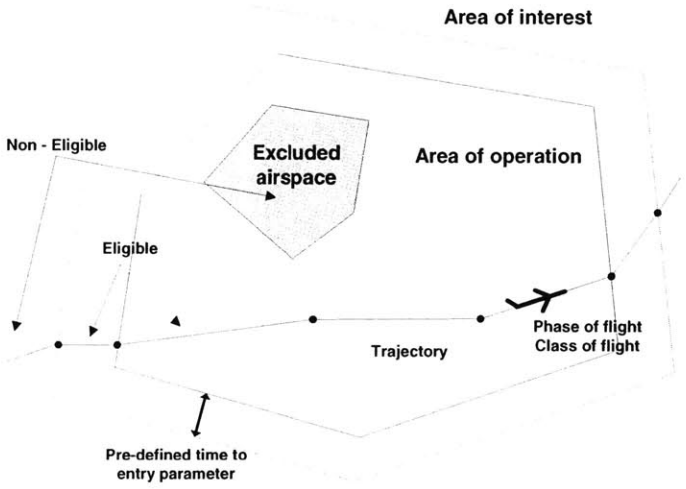

Figure 2-4: Eligible Flights, Area of Operation and Area of Interest

2.3 General system design and conflict detection principles

In this section, we describe the general conflict detection calculation principle. It is labeled as the design principle [2.7]

In the next section 2.4, we describe in more details the principles for each step of the model we used for conflict detection in more details.

These general system design principles, described by Eurocontrol in ??, allow us to start with some abstraction before to go into the details of the conflict detection calculations in the next section 2.4. This approach highlights the importance of the intellectual manageability of the information we bring to a first-time reader not familiar with MTCD. This is due to the need of having a readable and reviewable design with minimal training by people by a large variety of backgrounds and expertise.

[2.7.1]

For each conflict detection, MTCD divides trajectories in Trajectory Segments according to pre-defined Significant Points such that for each pair of segments of different flights, the applicable separation criteria can be determined unambiguously.

[2.7.2]

Furthermore, for each trajectory segment, one set of uncertainty parameters will be applicable. A trajectory segment plus uncertainty parameters determine a Trajectory Buffer.

[2.7.3]

During conflict detection calculations, segments and buffers of different flights are compared to each other and to special use airspace and to lowest usable flight levels within airspace.

[2.7.4]

Violations are stored in

- [2.7.4.1]

  Segment violations (between two trajectory segments),

- [2.7.4.2]
  Buffer violations (between two trajectory buffers),
- [2.7.4.3]
  Airspace violations (between a trajectory buffer and a special use airspace),
- [2.7.4.4]
  Level violations (between a trajectory buffer and an airspace with lowest usable flight level defined).

[2.7.5]
The detected violations must be grouped into sets of violations covering consecutive time frames, i.e, the end time of one violation coincides with the start time of another violation.
Rationale:
MTCD is supposed to detect conflict between trajectories, not violations between trajectory segments or buffers, this is why end/start times have to coincide.

In this way,

- [2.7.5.1]
  Consecutive Segment Violations are grouped into Nominal Routes Overlap between these two trajectories,
- [2.7.5.2]
  Consecutive Buffer Violations are grouped into Aircraft conflicts,
- [2.7.5.3]
  Consecutive Airspace Violations are grouped into Special Use Airspace Penetration
- [2.7.5.4]
  Consecutive Level Violations are grouped into Descent Below Levels.

Finally as stated in (2.1.2), this conflict data is then being sent to HMI for display.

These conflict detection calculations principles are shown in the following figure 2-5.

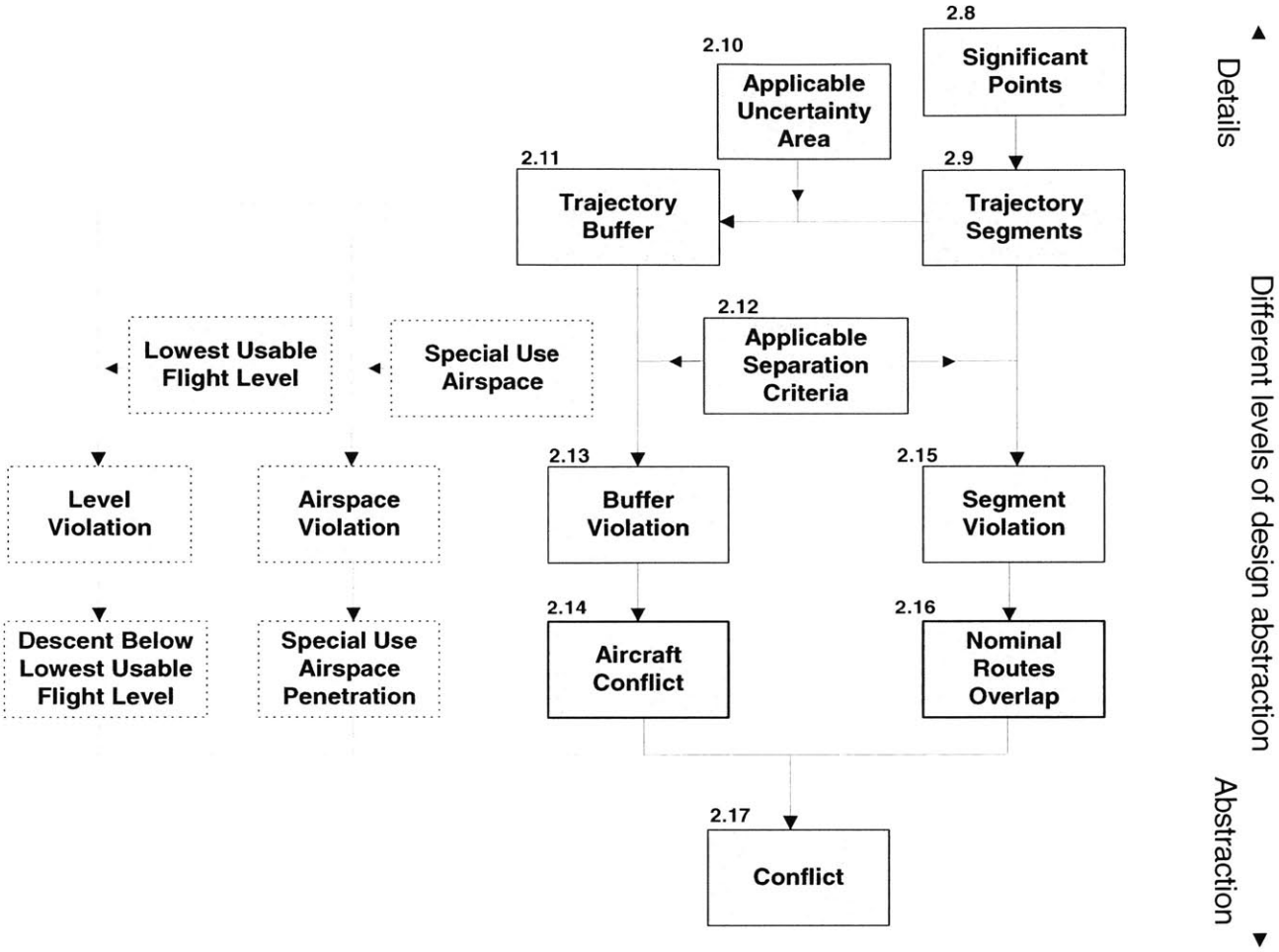

Figure 2-5: Conflict detection calculation principles

To introduce the internal system design and conflict detection , we decided to start from the details (significant points of a trajectory) or the lowest level of abstraction and to proceed to higher level of abstraction (conflict).

For the purpose of our demonstration, we focus on the types of conflict (the most complicated) involving two aircraft: nominal routes overlap and aircraft conflict.
Thus, we give the design principles of the following elements: significant point, trajectory segment, uncertainty area, trajectory buffer, separation criteria, buffer violation, segment violation, aircraft conflict, nominal routes overlap and conflict.

2.4 Detailed system design and calculation principles

2.4.1 Significant point

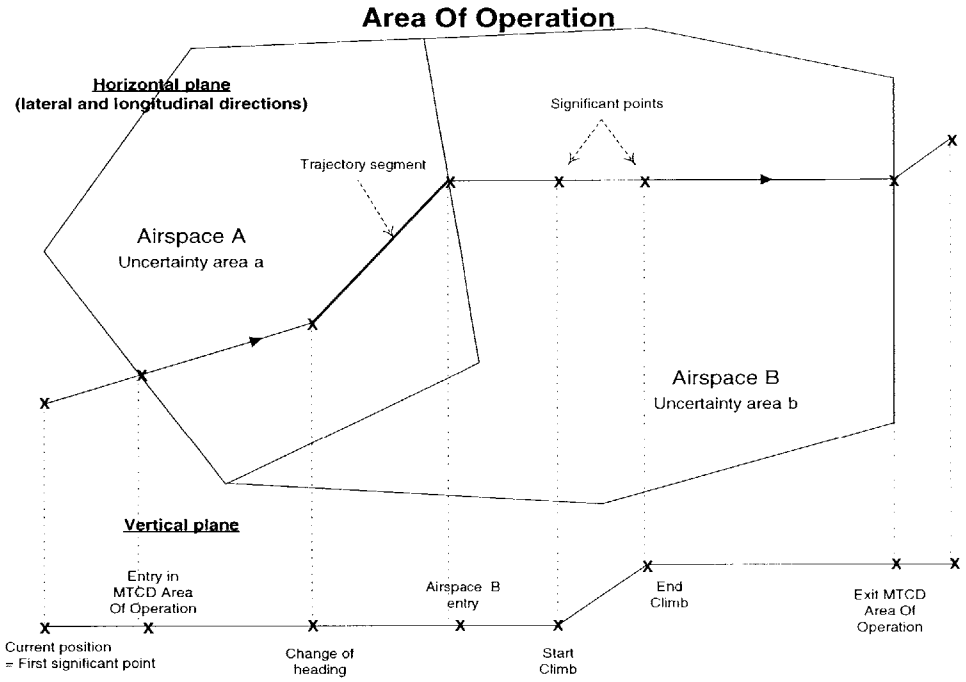

Figure 2-6: Construction of significant points

[2.8]

A significant point is a geographical location used in defining a predicted route of a flight.

- [2.8.1]

  The following points on a predicted trajectory shall be provided as significant points to MTCD:

  - change of heading,
  - start/end of climb/descent (change in flight-level),
  - change in phase of flight,
  - entry/exit of MTCD area of operation,
  - entry and exit of an user-defined-airspace,

  Rationale:

  These points have been chosen in such a way that MTCD will be able to determine

unambiguously the applicable uncertainty areas for a segment between two consecutive significant points (2.7.2).

These points have also been chosen in such a way that MTCD will be able to determine unambiguously the applicable separation criteria between any pair of segments (2.7.1).

- [2.8.2]

  This choice of significant points has the advantage that segments are straight lines. Therefore, at most one segment violation (2.16), and at most one buffer violation (2.14), can be possible between two segments.

- [2.8.3]

  The first significant point is the current position of a flight, possibly outside the area of operation. The next significant points are significant points of a trajectory within the area of operation for which estimated time is later than current time.

  Rationale:

  It is important to always take into account the current position of a flight, especially if it does not correspond to the data in the flight plan.

  Traceability: 1.37

2.4.2 Trajectory segments

[2.9]

A trajectory segment is a part of the nominal route of a flight between two consecutive significant points (2.8).

- [2.9.1]

  First MTCD determines if a segment lies within the MTCD area of operation. If not, a trajectory segment will not be created and conflict detection calculations will not be performed.

  Rationale:

  Conflict detection calculation is performed only within the area of operation.

  Traceability: 1.01

- [2.9.2]

  Next, MTCD determines whether a segment lies within an excluded airspace. If so, a trajectory segment will not be created and conflict detection calculations will not be performed. (2.6.5.3)

  Traceability: 1.07

- [2.9.3]

  Finallly, MTCD determines whether the segment lies in an airspace with different separation criteria, with different uncertainty, with different lowest usable flight level, or whether it lies on a leg of parallel ATS routes.

  Rationale:

  This information is later used to determine applicable separation criteria, uncertainty areas, and lowest usable flight level for the segment.

  Traceability: 1.17, 1.18, 1.19, 1.20

- [2.9.4]

  When a segment is updated, MTCD checks to see if the segment is still inside the prediction horizon. If the estimated time over the end point of the segment is less than the lower bounds of the prediction horizons away from the current time, the segment is deleted.

Rationale:

MTCD performs conflict detection calculations only within the prediction horizon.

Traceability: 1.01

- [2.9.5]

  Otherwise, when a segment is updated, existing segment violations are updated and nominal route overlap detection is triggered. If a trajectory buffer exists for this segment, the buffer will be updated too.

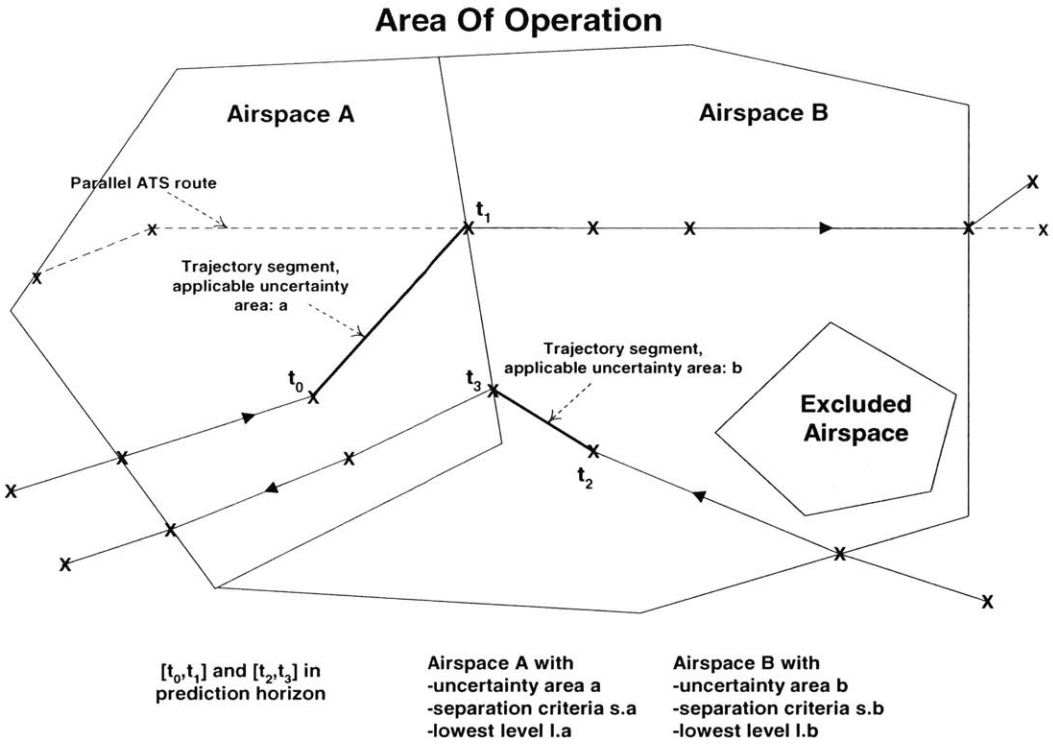

Figure 2-7: Construction of Trajectory Segments

With the trajectory segments, the detection of nominal route overlap and the creation of a trajectory buffer based on applicable uncertainty areas are triggered. To determine if there are any nominal routes overlap (2.2.3 we first need to determine if there are any segment violations (2.4.7). To determine if there are any aircraft conflict, we first build trajectory buffer (2.4.4), then determine if there are any buffer violations (2.4.5).

2.4.3 Uncertainty areas and Trajectory buffer

Uncertainty areas

The position of an aircraft in the future is not known for sure. We assume that by modeling the degree to which the trajectory path will be followed, in terms of uncertainty parameters, we can construct a reliable Medium Term Conflict Detection tool to predict aicraft conflict, special use airspace penetrations, nominal routes overlap and descent below lowest usable flight level.

[2.10]
To model the degree to which a trajectory path will be followed and to account for unpredictable aircraft maneuvers, MTCD constructs uncertainty areas around the trajectories that it receives from the FDPS. (Env-As-FDPS-01). The vertical and horizontal uncertainty areas around a trajectory define the probable positions of an aircraft at any given time.

- [2.10.1]
  MTCD uses default values for the uncertainty parameters for each AOO.
  Traceability: 1.20

- [2.10.2]
  MTCD uses other values than the default values for uncertainty areas when dealing with:

  - specific airspaces,
  - specific phases of flight,
  - specific navigational capabilities.

  Rationale:

  - Radar coverage depends on specific airspace (e.g oceanic sector). The uncertainty on the position of flight is larger for an airspace with low radar coverage than for an airspace with higher radar coverage.
  - For example, to model the climb/descent phase-of-flight uncertainty is less accurate than to model the cruise phase-of-flight uncertainty because of the uncertainty on the value of rate of climb/descent for each type of aircraft. The

rate-of-climb/-of descent is for example related to the local climatic conditions (pressure).

- Depending on navigational capabilities (different types TBD) the pilot can more or less accurately follow the trajectory described in his/her flight plan and provided by MTCD through the FDPS. (Env-As-FDPS-01).

These "specific" airspaces, phases-of-flight, navigational capabilities need to be specified (TBD). Each of these issues address technology limitations and might improve with the years.

Traceability: 1.19

- [2.10.3]

  The uncertainty parameters are resolved into lateral, longitudinal and vertical directions and are constructed independently for vertical and horizontal inacurracies.

- [2.10.4]

  MTCD uncertainty areas and the Monitoring Aids (MONA) deviations thresholds are tuned to each other. (Env-As-MONA-05)

  Rationale:

  If MONA lateral deviations thresholds were set to 1.5 nm and MTCD lateral uncertainty at 1nm, we could have an aircraft 1.3nm laterally off its path without any trajectory re-calculation being triggered. In this case, MTCD conflict data would not be reliable any more.(Env-Cstr-MONA-02)

  This is the reason why the basic lateral and longitudinal uncertainties are related to the MONA thresholds. .MONA would warn the controller and trigger a trajectory re-calculation whenever an aircraft deviates laterally and the aircraft would be brought back on track. (Env-As-MONA-02)

  The figure 2-8 and the following design principles highlight how these uncertainty areas will be built.

[2.10.5] Horizontal uncertainty

- [2.10.5.1]

  In the horizontal plane, a segment is defined as the probable positions of an aircraft in a given time interval. In MTCD these intervals are bounded by the earliest

time at which an aircraft is expected to be over one waypoint and the latest time at which it will be over the next waypoint.

- [2.10.5.2]
  Horizontal uncertainties are embodied in a buffer shape that is centered on the aircraft's nominal position.

- [2.10.5.3]
  The shape is a rectangle aligned to the aircraft's path with the corners rounded by a circle with diameter equal to the long axis of the rectangle.

- [2.10.5.4]
  MTCD uncertainty areas take into account a fixed value for lateral and vertical uncertainties.

- [2.10.5.5]
  MTCD uncertainty areas take into account ground speed unpredictability for longitudinal uncertainty.

[2.10.6] Vertical uncertainty

- [2.10.6.1]
  The vertical uncertainty area takes into account the inferred behavior from the trajectory.
  Rationale:
  The uncertainty is based on modeling the degree to which the trajectory path will be followed. It is for example important to determine at what level a climbing aircraft is going to level-off: an aircraft climbing at FL273 might seem to be in conflict with one level at FL290. However, if it is known that the climbing aircraft plans to level off at FL280 then no alert is required.

- [2.10.6.2]
  MTCD will not use the default vertical uncertainty values that we usually apply for a continuous flight level when an aicraft is at the end/start of climbing/descending flight phases.
  Rationale:
  A pilot might trigger a descent phase a little earlier than expected, thus our un-

certainty has to model this unpredictability. The shapes we adopt for end/start of descent/climb phases are given in the figure 2-8.

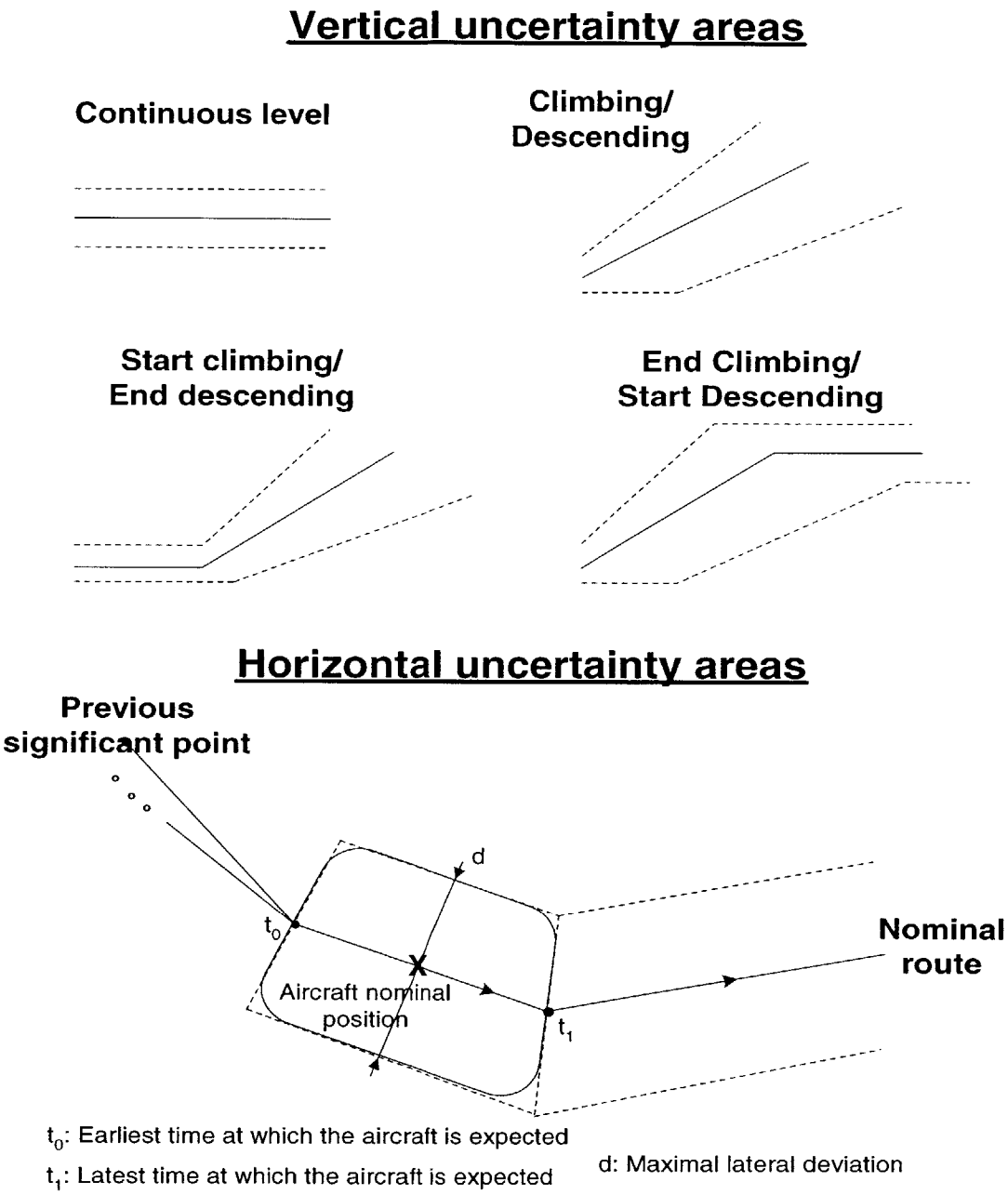

Figure 2-8: Vertical and Horizontal Uncertainty areas

2.4.4 Trajectory buffer

[2.11]

Using the trajectory segments and the uncertainty areas defined above we can build a trajectory buffer: A trajectory buffer is the uncertainty area around a trajectory segment to allow for deviations of any kind from the nominal route. For each trajectory segment within the prediction horizon of aircraft conflict, special use airspace penetration or descent below level, a trajectory buffer is created to account for deviations of a flight from its trajectory.

- [2.11.1]

  Exactly one uncertainty area will be applicable to each trajectory segment and thus to each trajectory buffer.

  Rationale: This refers to the design principle (2.10.1 and 2.10.2) and to the way we built our trajectory segment (2.7.2). This is summarized by the following principle:

- [2.11.2]

  The applicable uncertainty area will be determined for trajectory buffer following the criteria:

  - uncertainty area defined for airspace in which buffer lies, if any
  - phase of flight during the segment to which the buffer belongs,
  - navigational capabilities of the flight,
  - default uncertainty area defined for the area of operation.

  Uncertainty areas may be determined based on a prioritised list of these criteria.

- [2.11.3]

  A trajectory buffer will only be created when the time interval of the segment overlaps with the aircraft conflict, or special use airspace penetration or descent below lowest usable flight level prediction horizon.

- [2.11.4]

  When a buffer is updated, and the buffer is no longer inside any of the horizons, the buffer is deleted. Otherwise the applicable uncertainty areas are re-determined and confict detection calculations are either updated, or newly started, depending on

whether violations exist or not. When a trajectory buffer is deleted, all violations of this buffer are deleted as well.

Once the trajectory buffers have been built, three types of conflict detection calculations are triggered with buffer, level and airspace violations calculations.
Level and airspace violations are linked to the detection of descent below lowest usable flight level and of special use airspace penetration. Buffer violations are linked to the detection of aircraft conflict.
For our safety methodology demonstration, we now concentrate on buffer violation with aircraft conflict in sections 2.4.5 and 2.2.3 and on segment violations with nominal routes overlap in sections 2.4.7 and 2.2.3.

2.4.5 Separation criteria and Buffer violation

Separation criteria

[2.12]
The separation criteria is the minimum distance required between two flights in order not to be involved in an aircraft conflict and between nominal routes of two flights in order not to be involved in a nominal routes overlap.

The design principles for the definition of the separation criteria between flights are described below.

- [2.12.1]
  MTCD uses default values for the separation criteria per area of operation.
  Traceability: 1.18

- [2.12.2]
  MTCD uses different separation criteria for the following appropriate cases:

  - [2.12.2.1]
    Specific airspaces,
    Rationale:
    In the ECAC area, airspaces don't all have the same separation criteria. We can for example differentiate oceanic, continental airspaces...

  - [2.12.2.2]
    Specific phases of flight,
    Rationale:
    Separation criteria might be different for a flight in approach or cruise phase.

  - [2.12.2.3]
    Classes of flight,
    Rationale:
    Different types of aircraft with different size, velocity, rate of climb... require different separation criteria.

  - [2.12.2.4]
    The geometry of two trajectories,

Rationale:

To take into account worst case scenario (e.g. two aircraft in collision path, one in climb the other in descent phase).

- [2.12.2.5]

  Parallel ATS routes.

Traceability: 1.17

- [2.12.3]

MTCD will allow two ways of selecting the appropriate separation criteria for a pair of segment of different flights based on airspace, phases-of-flight, classes-of-flight, parallel ATS routes, and geometry of trajectories.

Rationale:

MTCD allows two ways of selecting the appropriate separation criteria to accomodate the different procedures used in the different countries and ATC services.

- [2.12.3.1]

  First, by means of a prioritised list of these parameters. MTCD will allow the definition of a list which will determine the order in which MTCD will search for applicable separation criteria.

  For instance if a list (parallel ATS routes, airspace, phase-of-flight, class-of-flight, geometry) has been defined, MTCD will first determine whether the segments are part of parallel ATS routes. If so, the separation criteria applicable to the parallel ATS route will be selected. If segments are not part of parallel ATS routes, MTCD will check whether separation criteria have been defined for the airspace in which the segments lie. If so, these separation criteria will be selected. If not, MTCD will check phases-of-flight, classes-of-flight, and geometry in a similar way. If applicable separation criteria have not been found, the default separation criteria of the MTCD area of operation will be applied.

- [2.12.3.2]

  Second, by means of a table covering all possible combinations of these parameters. If such a table has been defined, MTCD will determine the values of all parameters mentioned above and retrieve the applicable separation criteria from this table. If applicable separation criteria have not been found, the default

separation criteria of the MTCD area of operation will be applied.

- [2.12.4]

  The separation criteria values and means to select them are defined during MTCD and EDPS's configuration.

- [2.12.5]

  Whenever values are updated, all existing nominal routes overlaps and aircraft conflicts are deleted.

  Rationale:

  Otherwise the nominal routes overlap and aircraft conflict could be based on the old criteria.

Buffer violation

[2.13]

A buffer violation is the violation by two trajectory buffers of different flights of the separation criteria applicable to the pair of buffers.

- [2.13.1]

  It is created for a pair of trajectory buffers from trajectories of two different flights, only if the minimum distantce between any two possible positions within these buffers is less than the separation criteria applicable in this situation.

- [2.13.2]

  Trajectories have been split into buffers in such a way, that for each pair of buffer of trajectories of different flights, exactly one separation criteria will be applicable (2.7.1).

- [2.13.3]

  Before to perform conflict detection calculation, MTCD uses the uncertainty areas buffer shape, augments it with the applicable separation criteria in such a way that it includes an allowance for half the separation standard.

  Rationale:

  We just need to determine when buffers overlap which is easier than determining when buffers come within a certain distance (e.g. the separation standard) of each other.

- [2.13.4]

  From the algorithmic view point we consider the buffer shape as the intersection of a circle and a rectangle. When trying to determine overlaps between buffer shapes we first see if the circles overlap (circle test) and then check for rectangle overlap (box test).

  Rationale:

  The circle test is much quicker than the box test.

  In some ways an ellipse would have been more elegant as a buffer shape but satisfactory algorithms[1] that predict the times of overlaps of expanding ellipses have not been found.

[1]Section 2.5 describes the tradeoffs for the conflict detection calculations algorithm in more details

- [2.13.5]

  When a violation is detected, a check is performed to determine whether this violation should be part of an existing aicraft conflict.The violations of a conflict must be consecutive in time (consecutive means that the time of last violation of one violation is equal to the time of first violation of another violation of the same type).

  Rationale:

  MTCD is supposed to detect conflict between trajectories, not violations between trajectory segments or buffers, this is the violations have to be consecutive in time.

- [2.13.6]

  If an overlap in time exists with two conflicts, these two are joined. All violations of the second conflict are linked to the first one, and the second will be deleted. If the violation is not part of any existing aircraft conflict, a new conflict is created.

  When a violation is updated, the conflict detection calculations are performed again. Based on the result of these calculations, the buffer violation is either updated or deleted.

2.4.6 Aircraft conflict

The definition and design principle are described in the preceding section 2.2.2. We here give the calculation principle linked to buffer violation:

- [2.14]

  An aircraft conflict between two trajectories consists of a number of buffer violations of which the end/start times coincide.

  Rationale:

  MTCD is supposed to detect conflict between trajectories, not violations between trajectory segments or buffers, this is why end/start times have to coincide.

2.4.7 Segment violation

[2.15]

A segment violation is the violation by two trajectory segments of different flights of the separation criteria applicable to the pair of segments.

- [2.15.1]

  A segment violation is created for a pair of trajectory segments from trajectories of two different flights only if the minimimum distance between the segment is less than the separation criteria that are valid in this situation.

- [2.15.2]

  If an overlap in time exists with two conflicts, these two are joined. All violations of the second conflict are linked to the first one, and the second will be deleted.

- [2.15.3]

  When a violation is detected, a check is performed to determine whether this violation should be part of an existing nominal routes overlap. If the violation is not part of any existing nominal routes overlap, a new conflict is created. When a conflict is updated, the conflict detection calculations are performed again. Based on the result of these calculations, the segment violation is either updated or deleted.

Traceability: 1.10

2.4.8 Nominal routes overlap

The definition and design principle are described in the preceding section 2.2.3. We here give the calculation principle that is linked to segment violation:

A nominal routes overlap is the possible conflict between two aircraft based on their nominal routes.

- [2.16]

  A nominal routes overlap between two trajectories consists of a number of segment violations of which the end/start times coincide.The violations of a conflict must be consecutive in time (consecutive means that the time of last violation of one violation is equal to the time of first violation of another violation of the same type).

2.4.9 Conflict

A conflict is an abstract object introduced with [2.1]to represent either an aicraft conflict, or a special use airspace penetration, or a descent below lowest usable flight level, or a nominal routes overlap.

The dynamic behavior for all types of conflict are the same. Only difference is that a

different set of conflict data is sent to HMI for each type of conflict.
For each airspace violation, segment violation, level violation and buffer violation, an appropriate conflict object will be created or extended.

- [2.17.1]
  We have a conflict when a violation of any of these four types of violation does not have any overlap in time with any other violation of the same type between the same trajectories.

- [2.17.2]
  If the violation has an overlap in time with an existing conflict, then this conflict will be extended with the violation.

- [2.17.3]
  When detection is completed, all conflict data shall be gathered from the violations, the severity of the conflict shall be calculated and the conflict data shall be sent to HMI and other functions.

- [2.17.4]
  Conflicts shall be updated when one of the violations in the conflict is updated or deleted, or when a new violation is detected which overlaps in time with this conflict. When detection is completed, the conflict data will be sent to HMI and other functions.
  Conflicts will be deleted when all of the violations belonging to this conflict are removed.
  When violations belonging to a conflict are added, removed or updated, the first, last and worst violations must be recomputed.

- [2.17.5]
  Before sending out conflict data, the severity of the conflict has to be determined. This conflict severity categorization is defined, set, and maintainted by MTCD supervisor during the configuration phase.
  Rationale:
  The Eurocontrol only specifies that the conflict severity shall be calculated for MTCD but not how the severity shall be calculated. (TBD) The parameters on which the

severity shall be basedare left open for the States to implement to their own needs. This highlights the difficulties added for each Eurocontrol program that has to deal with a plethor of different nations and ATC services.

2.4.10 Configuration

Finally, describing the design principles, it is important to give the MTCD parameters that are to be set by the controllers or by management during the configuration phase. The preliminary hazard analysis, the preliminary task analysis in level 1 as well as our design in level 2 have identified important parameters that need to be configurable. We take them into account in our design:

[2.18]

The configuration parameters are MTCD parameters for tuning MTCD operational behavior, to be set and maintained by the MTCD supervisor, during the configuration phase, are:

- prediction horizons ,
- phases-of-flight to be excluded (2.6.5),
- classes-of-flight to be excluded (2.6.5),
- airspaces to be excluded (2.6.5),
- maximum time allowed between detection and first infringement,
- maximum time between two complete conflict detections of the same flight.

Traceability: 1.48, 1.49

[2.18.1]

MTCD is stopped and set in configuration mode in order to set/maintain the configuration parameters.

Rationale:

MTCD is to be configured after Startup. MTCD can not go operational if it has not been configured: in that case it is said to be failed.

Traceability: 1.49, Supv-05

[2.18.2]

Configuration parameters are set/maintained through the HMI.

Traceability: 1.HMI.03

[2.18.3]

MTCD provides the HMI with the current values of the configuration parameters.

Rationale:

At any time during operations, the MTCD supervisor and the PC shall be provided with feedback about the values of the configuration parameters.

Traceability: 1.47

2.5 MTCD algorithm principles, tradeoffs and design decisions

Introduction The level 1 provides the MTCD operational context and operational requirements and constraints. The preceding section gives the design and calculation principles of our system. No algorithmic treatment has been provided so far.

This section of level 2 provides a general algorithmic overview of MTCD for aircraft/aircraft conflict and highlights the associated tradeoff decisions. It has been based on work achieved by Eurocontrol in [17] and completed to highlight the scientific principles and the rationale for the design decisions.

Algorithmic overview In principle MTCD is really quite simple. The traffic and its evolution is specified by a set of trajectories, so all we need to do is examine these trajectories in pairs and report whenever such trajectories come too close. Complications occur because we need to:

- Specify how we model the aircraft trajectories: work on a ellipsoid or projection plane,
- Choose between an analytical or evolutive approach,
- Model uncertainties in aircraft behaviour,
- Introduce filtering mechanisms so that high traffic situations can be handled,
- Address controller concerns over specific conflict geometry, avoiding multiple and undesirable false alerts in approach situation.

The following sections discuss how these concerns are addressed in the algorithm and also consider how a number of fundamental design choices are made.

2.5.1 Ellipsoid or projection plane

We can calculate either in latitude/longitude or we can calculate in a flat projection which is simpler and faster but can introduce errors.

[2.19]

We choose to calculate in a flat projection first ensuring that the errors introduced are insignificant.

Rationale: Errors arise due to interpolation inaccuracies when trying to predict the aircraft position at times in between the specified trajectory points. These errors depend on segment length, segment orientation and distance from projection centre.

We have the following assumptions:

- 2.19-As.1

  In the projected plane the aircraft flies along a straight line (whereas in reality it flies along the image curve of a great circle geodesic),

- 2.19-As.2

  The segment lengths are of the order of 50NM(in fact this error increases as the square of the segment length).

- 2.19-As.3

  In the case of MTCD, we use a Mid-European projection centre and the worst case error for a relative distance measurement of 5NM is 1%,

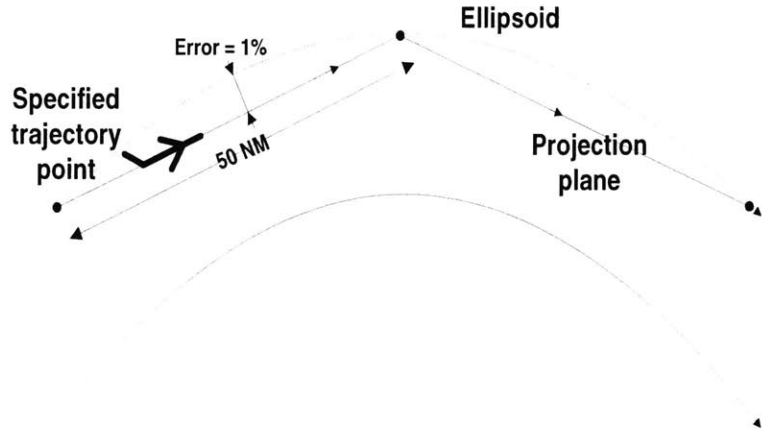

Figure 2-9: Ellipsoid or Projection plane

It is worth mentioning that long trajectory segments are not really a problem. But it

would be necessary to partition them along the great circle (if that is how aircraft navigates) prior to projection.

[2.19.1]

The actual projection system used transfer latitude/longitude on the WGS-84 (Earth Gravity Model) Ellipsoid to the projection plane in a straight forward manner.

[2.19.2]

The choice of working in the projection plane is to some extent linked to the adoption of the analytical approach (see section 2.5.2). If an evolutive approach had been taken, then working on the WSG-84 ellipsoid would have been viable.

2.5.2 Evolutive or analytical approach

We could either detect conflicts through an analytical or evolutive means.

In the evolutive approach, we step through the traffic development at regular intervals say (every 10secs) and at every snapshot we look for aircraft that are too close.

[2.20]

In the analytical approach, that we choose for our design, we figure out the relative dynamics of trajectories and determine the precise start and end time of conflicts.

Rationale:

The analytical approach is more accurate and in general faster than the evolutive approach. The evolutive approach is simpler and more extensible than the analytical approach.

We haven't found any more details about the rationale of this choice, this shall be specified (TBD).

However this choice seems similar (TBC) to the two major approaches of fluid dynamics: the Euler vs. the Reynolds approach. We either look at one fluid particle trajectory or we take a photograph of the full flow at every time steps t, t+dt...

2.5.3 Uncertainty modeling, probabilistic or geometric approach

At a specified future time the position of an aircraft is not known for sure, however its progress is likely to follow its planned profile. Uncertainty modeling is discussed in the following section with the design of the uncertainty parameters and the choice between a probabilistic and a geometric approach for MTCD.

Uncertainty parameters

The uncertainty parameters are resolved into lateral, longitudinal and vertical directions according to (2.10.3).

We give a numeric example to better illustrate the uncertainty issue of an aicraft's position.

- Lateral uncertainty= 1nm,
- Longitudinal uncertainty = 1nm + (look ahead time in seconds) * 0.002nm/s,
- Vertical uncertainty= 300ft.

[2.21.1]

The vertical uncertainty value of 300ft (TBC) applies to an aircraft flight's level uncertainty and is tuned to MONA thresholds (2.3.4).

Rationale:

The value of the vertical uncertainty parameter we use for MTCD is directly linked to the performance and accuracy of the altimeters currently used on-board aircraft (TBC). We also assume that this altitude, measured by these altimeters, is sent to the ground using transponders.

[2.21.2]

The longitudinal uncertainty parameter is modeled with a simple linear function:
a+b*(look ahead time in seconds).

Rationale: The longitudinal uncertainty grows with time to allow for ground speed unpredictability.

Ground speed unpredictability might be due to cross-, tail-winds (TBC).

We currently have no information about the value (0.002) given to "b" in the example

(TBD).

This longitudinal uncertainty might be reduced in the future if we improve our way to model ground speed unpredictability.

[2.21.3]

The values of the constants we use for the vertical uncertainty parameters (1nm for longitudinal and 1nm for lateral) are tuned to the MONA thresholds (2.3.4) and are linked to the current radar resolution.

Rationale:

Even if we don't really have the information, we can relate these numbers to assumptions linked to the actual radar resolution.

For example if we make the following assumptions to determine the azimuthal resolution:

- [2.21-As.1]

  The most common surveillance radar actually used has a 3dB beamwidth of 1.4 degrees,

- [2.21-As.2]

  And for an aircraft at 40 nautical miles from the location of the radar:

- [2.21-As.3]

  The azimuthal resolution is θ (3dB) * 40 NM = 1 nm

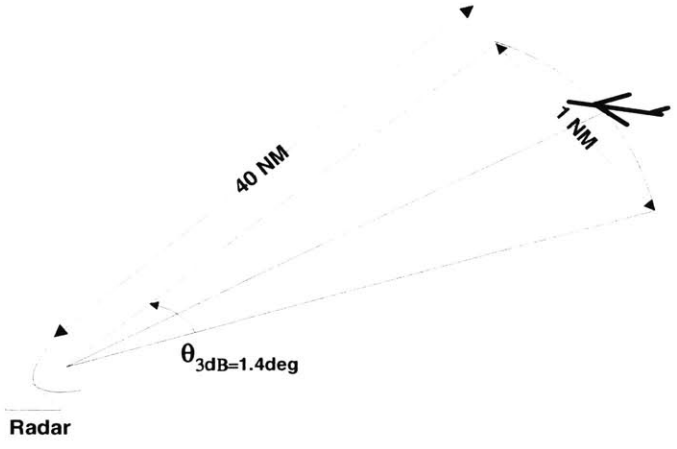

Figure 2-10: Radar azimuthal resolution

Rationale (cont'd):

For example, if we make the following assumptions to determine the range resolution:

- [2.21-As.4]

  The radar has a 10cm wavelength and a pulse width of 1 microsecond

- [2.21-As.5]

  The range resolution is given by Δ R = C * τ /2 = 450 ft

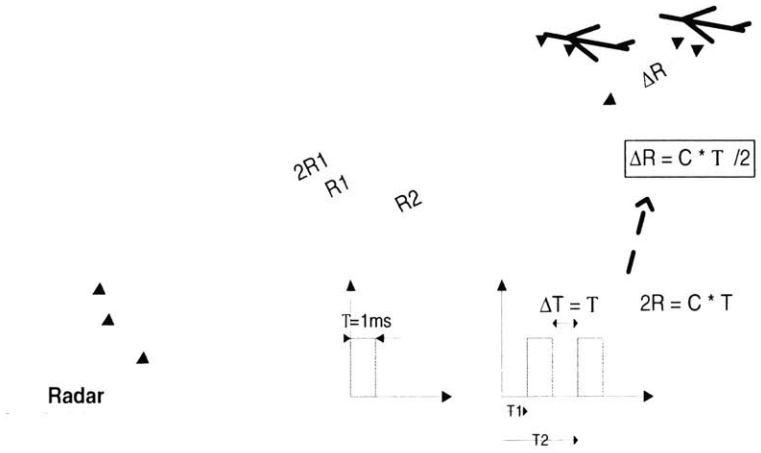

Figure 2-11: Radar range resolution

These radar range and azimuthal resolutions will give us the lateral and longitudinal uncertainties after a simple projection taking into account aircraft heading.

In the future, radar resolution may improve and allow smaller lateral and longitudinal uncertainty parameters.

This example we give to calculate the radar resolution is very simple. We don't claim it corresponds to the actual systems. We however use it to show what is the rationale hidden behind the MTCD uncertainty modeling design decisions.

Probabilistic vs. geometric MTCD

Probabilistic approach

In this paragraph, we define what is the probabilistic approach for the explicit calculation of conflict.

From uncertainty parameters we could derive some kind of shape that would contain for example 90% of an aircraft's possible positions at some future time. There could be several candidate shapes e.g. cuboid, ellipsoid or elliptical based cylinder. It might then be possible to, in effect, integrate projections through time of such shapes to determine the absolute probability of a conflict between aircraft.

This probabilistic treatment is attractive - for example if we decided that at most 10% false alerts are acceptable then we would give alerts whenever the conflict expectation reached 90%. However, the adoption of such an approach represents an HMI challenge in which controllers will need to be provided with views that indicate the probabilistic nature of conflicts.

Geometric approach

[2.22]
Even if it might be primarily thought that the objective of uncertainty modelling was to give some kind of probabilistic assessment of conflict likelihood, we rather choose a MTCD that can be, through tuning, aligned to the controllers conflict detection expectations.
Traceability: 1.51, 1.29, 1.30, 1.31, 1.32

- [2.22.1]
  For our approach, we construct a buffer shape that notionally surrounds an aircraft's future position.
  Rationale:
  We term this approach geometric because although the buffer sizes may embody uncertainties their ultimate values are determined through an optimization process that involves controller assessment of the MTCD tool.
  Traceability: 1.51

2.5.4 Filters

One filter

[2.23]

We decide to use a filtering process to apply a quick test that can easily eliminate most of the non-conflicts situations and that can reduce conflict calcuations computation duration with calculations not too long.

Rationale: In terms of computer processing, conflict detection is potentially the most demanding of automated ATC functions as it involves comparison between pairs of trajectories and this effort will increase as the square of the number of flights. Thus for 200 flights we have 19900 comparisons. Determining the exact details of a conflict depends on uncertainty models which in turn depend on segment containment in airspace volumes.Now in our 200 flight scenario we may only have 100 contemporary conflicts so 19800 checks would lead to a "no conflict" assessment.

One of the problem with these computations is that they might require more time than the required maximum response time: 500 milliseconds.

Traceability: 1.26

- [2.23.1]

  The conflict detection period is partitioned into a set of conflict timeslices.

- [2.23.2]

  For each trajectory the maximum and minimum extents (in both x and y coordinates of the projection) are determined for each timeslice. These extents define a set of airspace occupancy volumes (altitude is unrestricted).

- [2.23.3]

  Within a timeslice, trajectories cannot be in conflict if their corresponding airspace occupancy volumes are no closer than an amount specified by what is called "the airspace occupancy volume separation distance". It is determined by the largest horizontal separation standards and buffer uncertainties that apply for the timeslice airspace occupancy volumes.

- [2.23.4]

  The filter is used as a procedure that compares two trajectories for a given check

time interval and indicates whether or not a conflict is possible. If a conflict is possible, the procedure indicates a filtered time interval in which any conflict must occur. The filtered time interval is determined by the earliest and latest timeslices in which conflicts may occur (clipped if necessary by the check time interval).

- [2.23.5]
  For MTCD we need to perform conflict checks for up to 60 minutes in advance of the current time (nominal routes overlap).

Two-filter approach

[2.24]
The optimum size for timeslice will need to be tuned, however, more scope for achieving a good filtering performance may be achieved by cascading two such filters.
Rationale:
For example a primary filter can work with a timeslice of 16 minutes and a secondary filter would use a 2 minute timeslice. This two filter approach was used in the Eurocontrol MTCD simulation platform (Escape). It has been proved that the use of a third filter level offered no significant improvement when tried with 200 trajectory scenarios.

2.5.5 Conflict geometry: approach situations

[2.25]

To adjust the alerting of conflicts in approach situations, the horizontal conflict separation threshold is increased.

Rationale: In approach situations the geometry of the trajectories might lead to multiple, undesirable false alarms. This bothers controllers and therefore it is found necessary to introduce an adjustment for the detection of conflict in approach situations.

- [2.25.1]

  The following rule is being used:

  If and only if aircraft are approaching, then the conflict separation threshold is multiplied by the term $\sqrt{1 + cos(\theta)}$ where θ is the positive difference between headings. Since aircraft are approaching, θ is greater than 90 degrees and the multiplier ranges from 1 to $\sqrt{2}$ (at head on).

Rationale: According to Eurocontrol, the function used here is somewhat arbitraty but has however the right feel...

The rationale for this design decision should be clearly specified.(TBD)

If not possible, however, it at least clearly shows the limitation of our system on this point, and gives opportunities for improvements for future ATM tools.

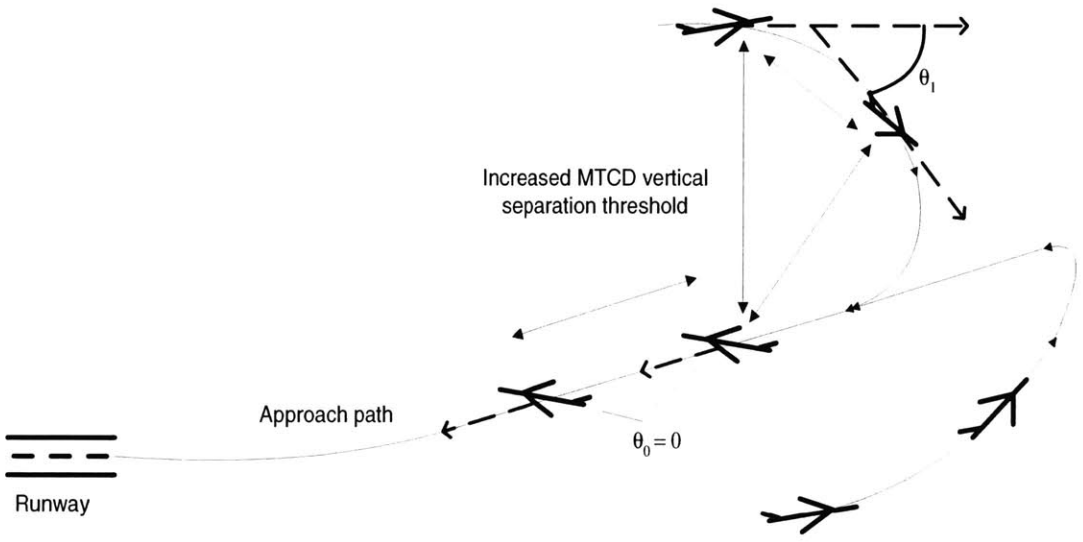

Figure 2-12: Increased conflict separation threshold for approach situations

Chapter 3

Level 3, System blackbox behavior

3.1 Introduction

Level 3 and SpecTRM-RL

As described by Leveson :

> The level 3 of the Intent Specifications describes the blackbox behavior of the system components, including humans, the logical aspects of the interfaces between the components and any assumed relevant environment behavior. The description includes no internal component design information and behavior is described only in terms of externally visible variables, objects and mathematical functions [23].

The Level 3 includes the assumed behavior of the external components: FDPS, EDPS, the HMI, (described in section 1.6.2) upon which the correctness of the system design is based and also contains a description of the interfaces and communication between the system and its environment.
The interactions with the Recording function and with the Arrival Manager are not detailed as no information is available at this time.

We use a specification language called SpecTRM-RL (Specification Tools and Requirements Methodology-Requirements Language) for modeling the blackbox software behavior using our system requirements (section 1.10).

> SpecTRM-RL was designed to satisfy both objectives: to be easily readable to serve as the official system specification of the behavioral requirements and to have an underlying formal model that can be executed and subjected to mathematical analysis [23].

Various types of analysis tools for SpecTRM-RL specifications are under development in the Software Engineering Research Laboratory at M.I.T. under the supervision of Pr Leveson.

For example, Completeness, Consistency, Deviation and State Machine Hazard Analyses (as defined in [28]) will later be performed as parts of a System and Subsystem Hazard Analysis.

For the purpose of this thesis, we describe in the following sections the blackbox model we built for MTCD.
This section has been done thanks to the contribution of Jayakanth Srinivasan.

3.2 Overview of the SpecTRM-RL model for MTCD

The SpecTRM-RL specification is composed of four main parts: a specification of the supervisory modes, a specification of the operating modes, a model of the controlled process that includes the inferred system states, and a specification of the inputs and outputs. These four main parts of our model are shown in figure 3-1.

> **A specification of the supervisory modes** of the controller being modeled: The supervisory modes determine who or what is controlling the component at any time. The control loops are organized hierarchically, with multiple controllers or components, each being controlled by the layer above and controlling the layer below. Each component may have multiple controllers (or supervisors). Required behavior may be different depending of what supervisory mode is currently in effect [23].

In our study, the controller being modeled is MTCD, and the supervisory modes identified for MTCD are:

- The PC,
- The MTCD supervisor,
- And the future AMAN.

(Section 3.3).

> **A specification of the operating modes for the controller**, The component operating modes control the behavior of the control component itself. They are used to control the interpretation of the component's interfaces. These are not internal states of our system but simply represent externally visible behavior about the controller's modes of operation [23].

In our study, MTCD is the controller of the airspace and the operating modes for the controller are:

- Startup,
- Configured,

- Operational,
- Failed,
- And Stopped.

(Section 3.4 and bottom left quadrant of figure 3-1).

> **A model of the controlled process** that includes the inferred operating modes and system states.[...]
> With the SpecTRM-RL language, we do not describe the actual transitions between the states into tables. We instead use the tables simply to represent one predicate logic statement about the conditions on one transition arrow between states. The tables are therefore of much more limited size and their use scales up ot large and complex system specifications while still remaining relatively small [23].

In our model the controlled process is the set of flights within the controlled airspace. The MTCD system states are:

- Flight [1...Number of flights],
  - Flight-Data,
  - Position,
  - Status,
  - Detect,
- Trajectory [1...Number of trajectories],
  - Data,
  - Type,
  - Status,
- Airspace [1...Number of airspaces],
  - Special-Use-Airspace,
  - Uncertainty-Airspace,

  - Separation-Criteria-Airspace,
  - Lowest-Flight-Level,
  - Airspace,
- Phase-of-Flight,
- Class-of-Flight,
- Display-Configuration,
- AOO,
- Navigational-Capabilities,
- Parallel-ATS-Routes.

(Section 3.10).

> **A specification of the inputs and outputs** to our system. The intended use of the SpecTRM-RL language is to define a blackbox function from outputs to inputs.

The next paragraph describes what are the outputs/inputs according to the SpecTRM-RL model as described by Leveson:

> By starting from the output specification, the specification reader can determine what inputs trigger that output and the relationship between the inputs and outputs. [...]
>
> The conditions under which an output is triggered or sent is simply a predicate logic statement over the various states, variables, and modes of the specification. [...]
>
> To find the triggering conditions required to accurately capture the requirements is complex. To overcome this problem, we use a tabular representation of disjunctive normal form that we call AND/OR tables [23].

In our figure 3-1, the outputs are denoted by outward pointing arrows and the inputs with arrows pointing to the blackbox.

In our model, the **Outputs** are sent from MTCD to the HMI. They are:

- Aircraft-Conflict,
- Nominal-Routes-Overlap,
- Special-Use-Airspace,
- Descent-Below-Lowest-Level,
- Configuration,
- Flight-Reincluded,
- Flight-Excluded

There will later be outputs sent from MTCD to the Recording function and to AMAN. (TBD).

In our model, the **Inputs** are received from the FDPS, EDPS, the HMI to MTCD:

- From the FDPS (section 3.7):
  - New-Flight,
  - Left-AOO,
  - Changed-Flight-Data,
  - New-Trajectory,
  - Recalculated-Trajectory,
  - Delete-Trajectory,
  - Current-Position.
- From the EDPS (section 3.8):
  - AOO,
  - Exclude-Airspace,
  - Special-Use-Airspace,
  - Separation-Criteria-Airspace,
  - Uncertainty-Area-Airspace,
  - Lowest-Flight-Level,
  - Phase-of-Flight,
  - Class-of-Flight,
  - Parallel-ATS-Routes,
- From the HMI (section 3.9):
  - MTCD-Start
  - MTCD-Stop
  - Exclude-Flight,
  - Reinclude-Flight.

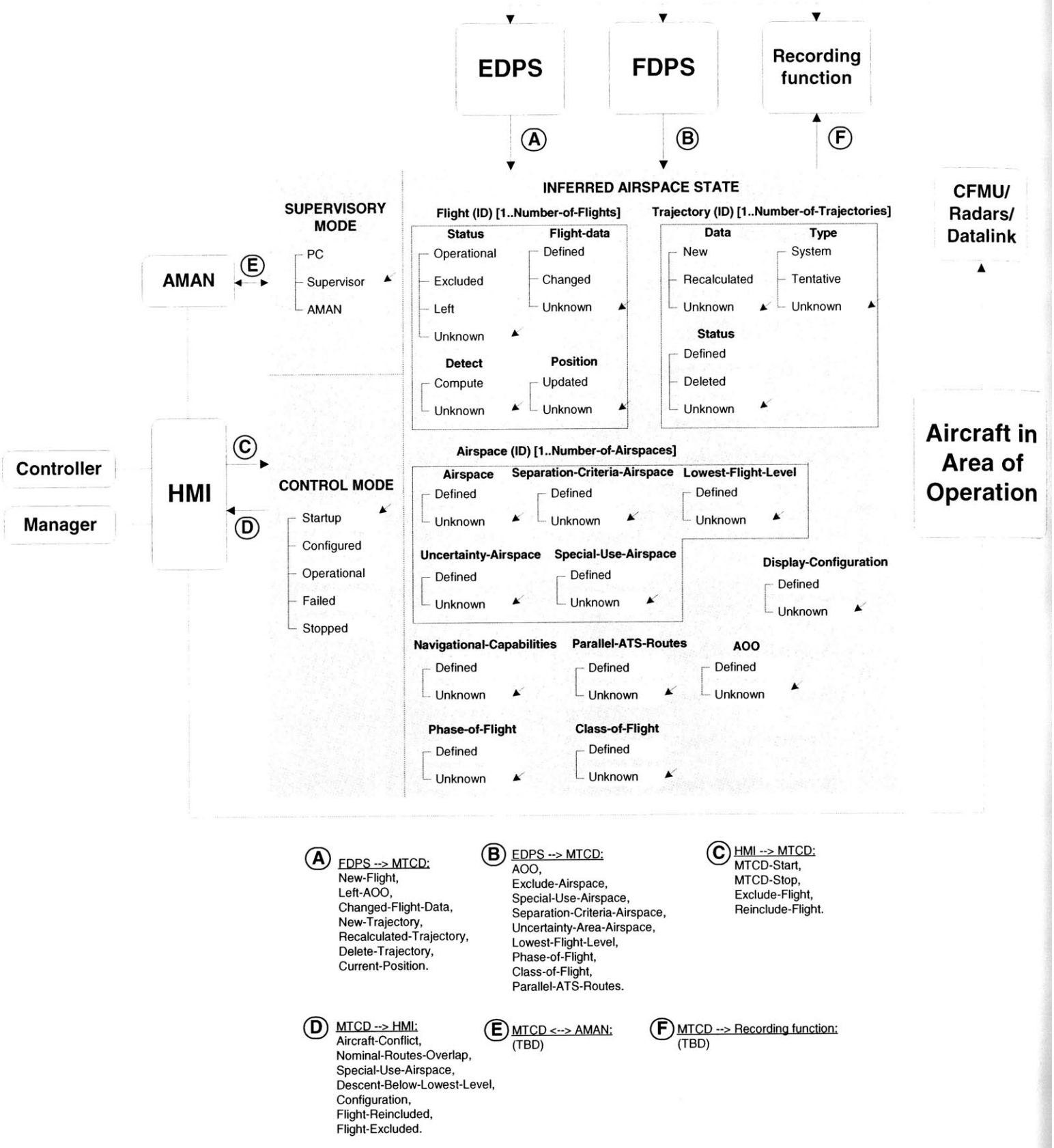

Figure 3-1: MTCD SpecTRM-RL Model

3.3 Supervisory Modes

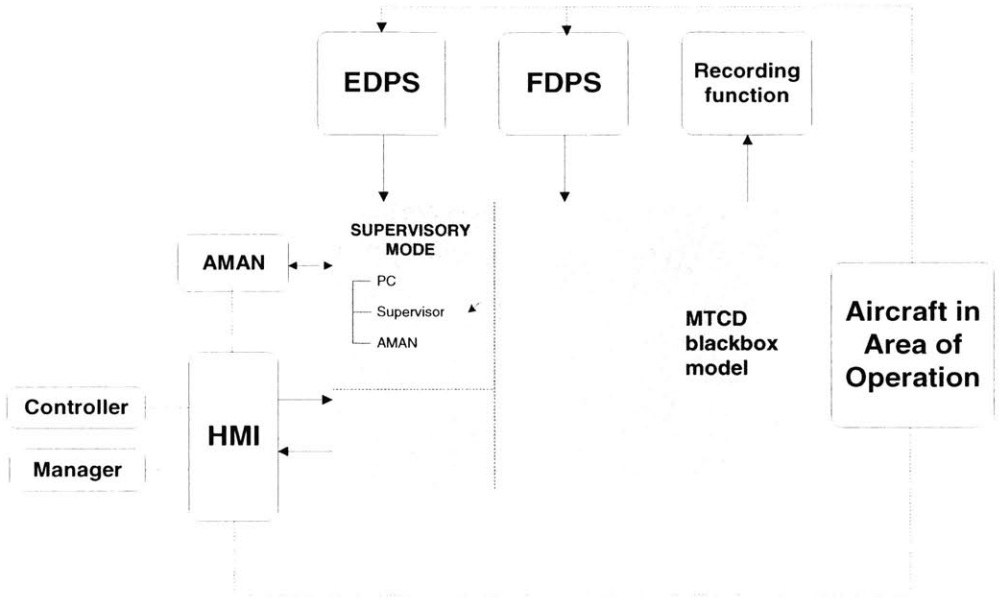

Figure 3-2: MTCD Supervisory Modes

- **Description:** The supervisory modes specify the MTCD supervisor at any time.
- **References:** Op-01, Supv-01, Supv-02, 1.52.[1]

Definition:

=AMAN

Control-Mode = Operational	T
Received User-Aman	T

=MTCD supervisor

Received MTCD-START	T
Control-Mode = Operational	F

= PC

Control-Mode = Operational	T
Received User-Aman	F

[1]Here the link should indeed be made with the Task Analysis of level 2 by Daouk [8].

3.4 Control Modes

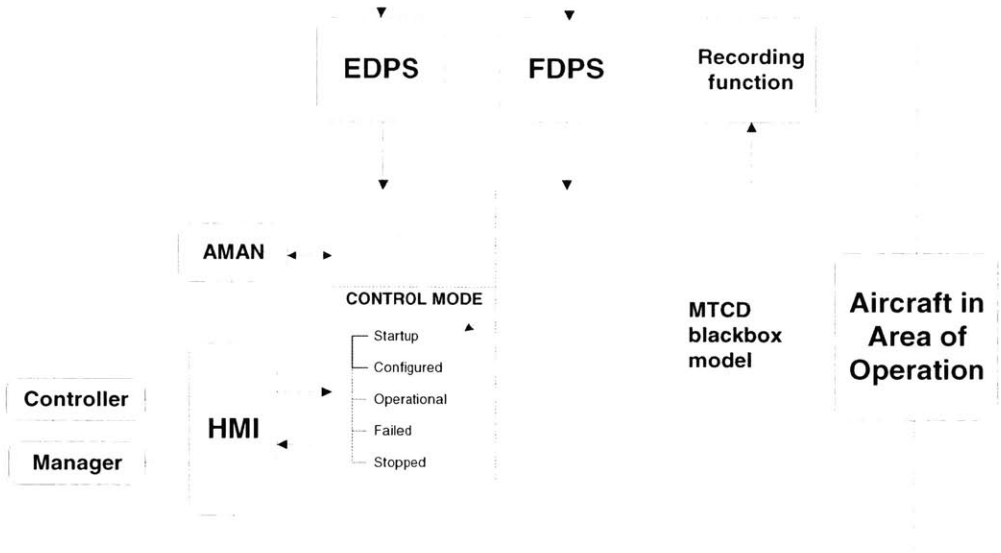

Figure 3-3: MTCD Control Modes

- **Description:** The control modes specify the modes of operation of MTCD.
- **References:** [2.18.1]

Definition:

=Startup

Received MTCD-START	T
MTCD-ConfiguredM	F

= Configured

Previous (Control-Mode) = Startup	T
Received MTCD-STOP	F
MTCD-ConfiguredM	T
Trajectory-DefinedM	F

= Operational

Previous (Control-Mode) = Configured	T
Received MTCD-STOP	F
MTCD-ConfiguredM	T
Trajectory-DefinedM	T

= Failed

Previous (Control-Mode) = Configured	T	-
Previous (Control-Mode) = Operational	-	T
Received MTCD-STOP	F	F
MTCD-ConfiguredM	F	F
Trajectory-DefinedM	F	F

= Stopped

Received MTCD-STOP	T

3.5 Output messages

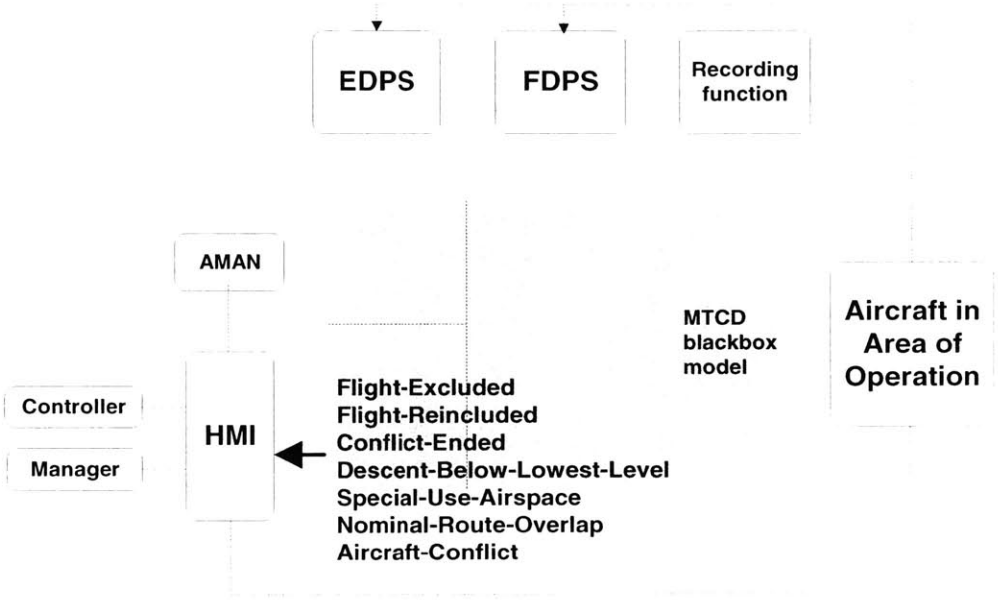

Figure 3-4: MTCD Outputs to HMI

Flight-Excluded

OUTPUT MESSAGE

- **Destination:** HMI
- **Description:** The message is sent to the HMI to acknowledge an Exclude-Flight message
- **References:** [2.6.6]
- **Contents:**

Field	Value
Flight-Id	Flight[J].Flight-Id

- **Trigger**

There exist I, J such that:

Control Mode	Operational	T
Inputs	Received Exclude-Flight Message	T
	Exclude-Flight[I].Flight-Id = Obsolete	F
	Flight[J].Flight-Id = Exclude-Flight[I].Flight-Id	T
State Variables	Flight[J].Status = Excluded	T

Flight-Reincluded

OUTPUT MESSAGE

- **Destination:** HMI
- **Description:** The message is sent to the HMI to acknowledge a Reinclude-Flight message
- **References:** [2.6.6]
- **Contents:**

Field	Value
Flight-Id	Flight[J].Flight-Id

- **Trigger**

There exist I, J such that:

Control Mode	Operational	T
Inputs	Received Reinclude-Flight Message	T
	Reinclude-Flight[I].Flight-Id = Obsolete	F
	Flight[J].Flight-Id = Reinclude-Flight[I].Flight-Id	T
State Variables	Flight[J].Status = Included	T

Configuration

OUTPUT MESSAGE

- **Destination:** HMI
- **Description:** The message is sent to the HMI to inform the controller about the current configuration
- **References:** [2.18.3]
- **Contents:**

Field	Value
Conflict-Id	MTCD-Configuration.Configuration-Id
Prediction-Horizon	MTCD-Configuration.Prediction-Horizon
Time-between-Detections	MTCD-Configuration.Time-between-Detections

- **Trigger**

Control Mode	Operational	T
Inputs	Received.DISPLAY-CONFIGURATION	T

Descent-Below-Lowest-Level

OUTPUT MESSAGE

- **Destination:** HMI
- **Description:** The message is sent to the HMI to inform it of the existence of a descent below lowest level violation.
- **References:** [2.1], [2.1.1]
- **Contents:**

Field	**Value**
Conflict-Id	Determine-Conflict-Id(Flight[I].Flight-Id, Trajectory[J].Trajectory-Id, 0, 0)
Conflict-Type	Descent-Below-Lowest-Level
Severity	TBD
Trajectory-Id	Trajectory[J].Trajectory-Id
Airspace-Id	Airspace-Violated(J, Descent-Below-Lowest-Level)
Time-First-Descent	First-Infringement(J, Airspace-Violated(J), Descent-Below-Lowest-Level)
Possible-Position	Possible-Position(J, Time-First-Descent)
Nominal-Position	Nominal-Position(J, Time-First-Descent)

- **Trigger**

There exist I such that for each J

Control Mode	Operational	T
State Variables	Flight[I].Detect = Compute	T
	Trajectory[J].Flight-Id = Flight[I].Flight-Id	T
	LFLV(J)	T

Special-Use-Airspace

OUTPUT MESSAGE

- **Destination:** HMI
- **Description:** The message is sent to the HMI to inform it of the existence of a special use airspace violation.
- **References:** [2.1], [2.1.1]
- **Contents:**

Field	**Value**
Conflict-Id	Determine-Conflict-Id(Flight[I].Flight-Id, Trajectory[J].Trajectory-Id, 0, 0)
Conflict-Type	Special Use Airspace
Severity	TBD
Trajectory-Id	Trajectory[J].Trajectory-Id
Airspace-Id	Airspace[Airspace-Violated(J)].Airspace-Id
Time-First-Loss	First-Infringement(J, Airspace-Violated(J), Special-Use-Airspace)
Possible-Position	Possible-Position(J, Time-First-Loss)
Nominal-Position	Nominal-Position(J, Time-First-Loss)
Time-Nominal-Min	Min-Nominal-Distance(J)
Min-Nominal-Distance	Nominal-Position(J, Time-Nominal-Distance)

- **Trigger**

There exist I, such that for each J

Control Mode	Operational	T
State Variables	Flight[I].Detect = Compute	T
	Trajectory[J].Flight-Id = Flight[I].Flight-Id	T
	Special-Use-Airspace(J)	T

Nominal-Route-Overlap

OUTPUT MESSAGE

- **Destination:** HMI
- **Description:** The message is sent to the HMI to inform it of the existence of a nominal routes overlap
- **References:** [2.1], [2.1.1]
- **Contents:**

Field	**Value**
Conflict-Id	Determine-Conflict-Id(Flight[I].Flight-Id, Trajectory[J].Trajectory-Id,
	Flight[K].Flight-Id, Trajectory[L].Trajectory-Id)
Conflict-Type	Nominal Route Overlap
Severity	TBD
Trajectory-Id-1	Trajectory[J].Trajectory-Id
Trajectory-Id-2	Trajectory[L].Trajectory-Id
Time-First-Loss	First-Infringement(J, L, Nominal-Routes-Overlap)
Nominal-Position-1	Nominal-Position(J, Time-First-Loss)
Nominal-Position-2	Nominal-Position(L, Time-First-Loss)
Time-Min-Nominal	Min-Nominal-Distance(J,L)
Min-Nominal-Position-1	Nominal-Position(J, Time-Min-Nominal)
Min-Nominal-Position-2	Nominal-Position(L, Time-Min-Nominal)

- **Trigger**

There exist I such that for all J, K, L:

Control Mode	Operational	T
State Variables	Flight[I].Detect = Compute	T
	Flight[I].Status = Operational	T
	Trajectory[J].Flight-Id = Flight[I].Flight-Id	T
	Trajectory[J].Status = Operational	T
	Flight[K].Flight-Id = Flight[I].Flight-Id	F
	Flight[K].Status = Operational	T
	Trajectory[L].Flight-Id = Flight[K].Flight-Id	T
	Trajectory[L].Status = Operational	T
	Nominal-Route-Overlap(J,L)	T
	Not-Detected-Earlier(J,L)	T

Aircraft-Conflict

OUTPUT MESSAGE

- **Destination:** HMI
- **Description:** The message is sent to the HMI to inform it of the existence of an aircraft conflict
- **References:** [2.1], [2.1.1]
- **Contents:**

Field	**Value**
Conflict-Id	Determine-Conflict-Id(Flight[I].Flight-Id, Trajectory[J].Trajectory-Id,
	Flight[K].Flight-Id, Trajectory[L].Trajectory-Id)
Conflict-Type	Aircraft-Conflict
Severity	TBD
Trajectory-Id -1	Trajectory[J].Trajectory-Id
Trajectory-Id -2	Trajectory[L].Trajectory-Id
Time-First-Loss	First-Infringement(J, L, Aircraf-Conflict)
First-Loss-Position-1	Possible-Position(J, Time-First-Loss)
First-Loss-Position-2	Possible-Position(L, Time-First-Loss)
First-Nominal-Position-1	Nominal-Position(J, Time-First-Loss)
First-Nominal-Position-2	Nominal-Position(L, Time-First-Loss)
Time-Min-Distance	Min-Distance(I, J)

Field (cont'd)	Value (cont'd)
Min-Prob-Distance-1	Flight-Position(I, Time-Min-Distance)
Min-Prob-Distance-2	Flight-Position(J, Time-Min-Distance)
Nominal-Distance-1	Nominal-Position(I, Time-Min-Distance)
Nominal-Distance-2	Nominal-Position(J, Time-Min- Distance)
Time-Min-Nom-Distance	Min-Nom-Distance(J, L)
Min-Nominal-Distance-1	Nominal-Position(J, Time-Min-Nom-Distance)
Min-Nominal-Distance-2	Nominal-Position(L, Time-Min-Nom-Distance)
Time-Last-Loss	Last-Infringement(J, L, Airspace-Violated(J), Aircraft-Conflict)
Last-Loss-Position-1	Possible-Position(J, Time-Last-Loss)
Last-Loss-Position-2	Possible-Position(L, Time-Last-Loss)
Last-Nominal-Position-1	Nominal-Position(J, Time-Last-Loss)
Last-Nominal-Position-2	Nominal-Position(L, Time-Last-Loss)

- **Trigger**

There exist I such that for all J, K, L:

Control Mode	Operational	T
State Variables	Flight[I].Detect = Compute	T
	Flight[I].Status = Operational	T
	Trajectory[J].Flight-Id = Flight[I].Flight-Id	T
	Trajectory[J].Status = Operational	T
	Flight[K].Flight-Id = Flight[I].Flight-Id	F
	Flight[K].Status = Operational	T
	Trajectory[L].Flight-Id = Flight[K].Flight-Id	T
	Trajectory[L].Status = Operational	T
	Aircraft-Conflict(J,L)	T
	Not-Detected-Earlier(J,L)	T

3.6 Input messages

Input messages from FDPS to MTCD

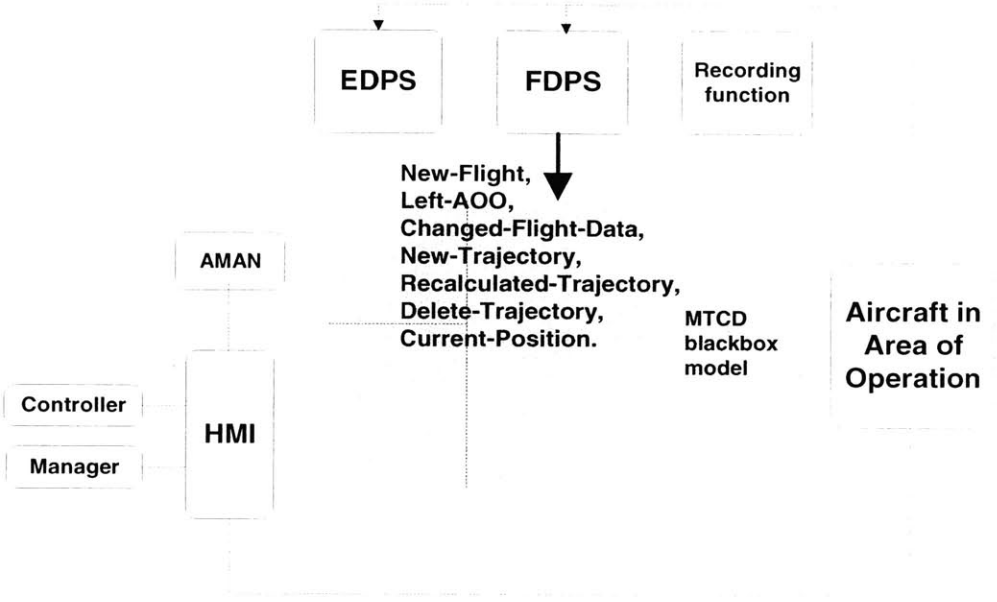

Figure 3-5: Inputs from FDPS to MTCD

NEW-FLIGHT

INPUT MESSAGE

- **Source:** FDPS
- **Description:** When a flight has to be considered for conflict computations, the flight information is sent by FDPS using a NEW-FLIGHT message.
- **References:** Env-As-FDPS-01
  - NEW-FLIGHT(
    - Flight-Id,
    - Aircraft-Type,
    - Navigation-Capabilities,
    - Class-of-Flight,
    - Phase-of-Flight,
    - Current-Position)

LEFT-AOO

INPUT MESSAGE

- **Source:** FDPS
- **Description:** When a flight leaves the Area of Operations of MTCD, FDPS informs MTCD using the LEFT-AOO message.
- **References:** Env-As-FDPS-01
  - LEFT-AOO(
    - Flight-Id)

CHANGED-FLIGHT-DATA

INPUT MESSAGE

- **Source:** FDPS
- **Description:** When the flight data is changed, MTCD is informed by FDPS via the CHANGED-FLIGHT-DATA message
- **References:** Env-As-FDPS-03
  - CHANGED-FLIGHT-DATA(
    - Flight-Id,
    - Aircraft-Type,
    - Navigation-Capabilities,
    - Class-of-Flight,
    - Phase-of-Flight)

NEW-TRAJECTORY

INPUT MESSAGE

- **Source:** FDPS
- **Description:** When a trajectory is created in the MTCD AOO, it is provided to MTCD using the NEW-TRAJECTORY message.
- **References:** Env-As-FDPS-04
  - NEW-TRAJECTORY(
    - Trajectory-Id,
    - Flight-Id,
    - Tentative-Indication,
    - Nominal-Route)

RECALCULATED-TRAJECTORY

INPUT MESSAGE

- **Source:** FDPS
- **Description:** When a trajectory is recomputed in the MTCD AOO, it is provided to MTCD using the RECALCULATED-TRAJECTORY message.
- **References:** Env-As-FDPS-01
  - RECALCULATED-TRAJECTORY(
    - Trajectory-Id,
    - Tentative-Indication,
    - Nominal-Route)

DELETE-TRAJECTORY

INPUT MESSAGE

- **Source:** FDPS
- **Description:** A trajectory is deleted from conflict detection using the DELETE-TRAJECTORY message.
- **References:** Env-As-FDPS-04, Env-As-FDPS-05
  - DELETE-TRAJECTORY(
    - Trajectory-Id)

CURRENT-POSITION

INPUT MESSAGE

- **Source:** FDPS
- **Description:** The position of the aircraft is given to MTCD using the CURRENT-POSITION message
- **References:** Env-As-FDPS-01, [2.8.3]
  - CURRENT-POSITION(
    - Flight-Id,
    - Current-Position)

Input messages from EDPS to MTCD

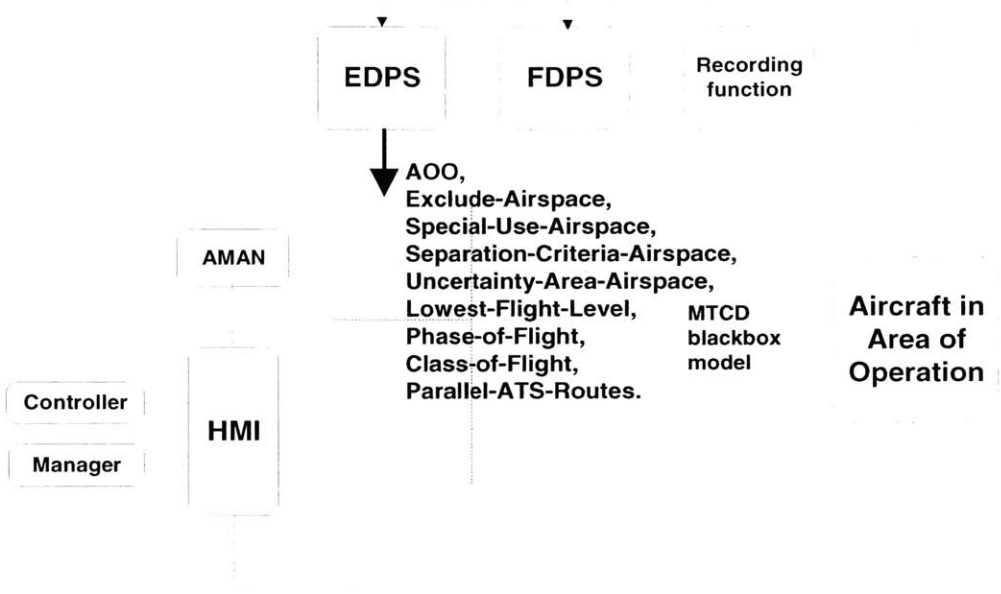

Figure 3-6: Inputs from EDPS to MTCD

AOO

INPUT MESSAGE

- **Source:** EDPS
- **Description:** When MTCD is started up, it receives the Area of Operations with default setting through the AOO message
- **References:** Env-As-EDPS-01.1
  - AOO(
    - Airspace-Id,
    - Airspace-Points
    - Lowest-Airspace-FL
    - Highest-Airspace-FL
    - Default-Separation-Criteria
    - Default-Uncertainty-Area
    - Default-Lowest-Usable-FL
    - List of airspaces in the AOO)

EXCLUDE-AIRSPACE

INPUT MESSAGE

- **Source:** EDPS
- **Description:** An airspace can be excluded from the airspace computation using the EXCLUDE-AIRSPACE message.
- **References:** [2.6.5]
  - EXCLUDE-AIRSPACE(
    - Airspace-Id)

SPECIAL-USE-AIRSPACE

INPUT MESSAGE

- **Source:** EDPS
- **Description:** Special use airspaces are defined for MTCD using the SPECIAL-USE-AIRSPACE message. They can be defined at any time during MTCD operation.
- **References:** Env-As-EDPS-01.1
  - SPECIAL-USE-AIRSPACE(
    - Airspace-Id,
    - Begin-Time-of-Restriction
    - End-Time-of-Restriction)

SEPARATION-CRITERIA-AIRSPACE

INPUT MESSAGE

- **Source:** EDPS
- **Description:** Separation criteria for an airspace is defined using the SEPARATION-CRITERIA-AIRSPACE message. It can be defined/ redefined any time during MTCD operation.
- **References:** [2.12.2.1]
  - SEPARATION-CRITERIA-AIRSPACE(
    - Airspace-Id,
    - Separation-Criteria)

UNCERTAINTY-AREA-AIRSPACE

INPUT MESSAGE

- **Source:** EDPS
- **Description:** Uncertainty parameters for an airspace is defined using the UNCERTAINTY-AREA-AIRSPACE message. It can be defined, redefined any time during MTCD operation
- **References:** [2.10.2]
  - UNCERTAINTY-AREA-AIRSPACE(
    - Airspace-Id,
    - Uncertainty-Parameters)

LOWEST-FLIGHT-LEVEL

INPUT MESSAGE

- **Source:** EDPS
- **Description:** Lowest flight level for an airspace is defined using the LOWEST-FLIGHT-LEVEL message. It can be defined, redefined any time during MTCD operation.
- **References:** Env-As-EDPS-01
  - LOWEST-FLIGHT-LEVEL(
    - Airspace-Id,
    - Lowest-Flight-Level)

PHASE-OF-FLIGHT

INPUT MESSAGE

- **Source:** EDPS
- **Description:** Phase of flight parameters (separation criteria and uncertainty areas) are defined using the PHASE-OF-FLIGHT messages. This is defined at startup and remains static during MTCD operation.
- **References:** [2.10.2], [2.12.2.2]
  - PHASE-OF-FLIGHT(
    - Phase-of-flight,
    - Separation criteria,
    - Uncertainty-Parameters)

CLASS-OF-FLIGHT

INPUT MESSAGE

- **Source:** EDPS
- **Description:** Class-of-flight separation criteria parameters are defined using the CLASS-OF-FLIGHT messages. This is defined at startup and remains static during MTCD operation.
- **References:** [2.12.2.3]
  - CLASS-OF-FLIGHT(
    - Phase-of-flight,
    - Separation criteria)

PARALLEL-ATS-ROUTES

INPUT MESSAGE

- **Source:** EDPS
- **Description:** Parallel ATS routes and the associated separation criteria are defined using the PARALLEL-ATS-ROUTES message. This is defined at startup and remains static during MTCD operation
- **References:** [2.12.2.5]
  - PARALLEL-ATS-ROUTES(
    - Routes,
    - Separation criteria)

Input messages from the HMI to MTCD

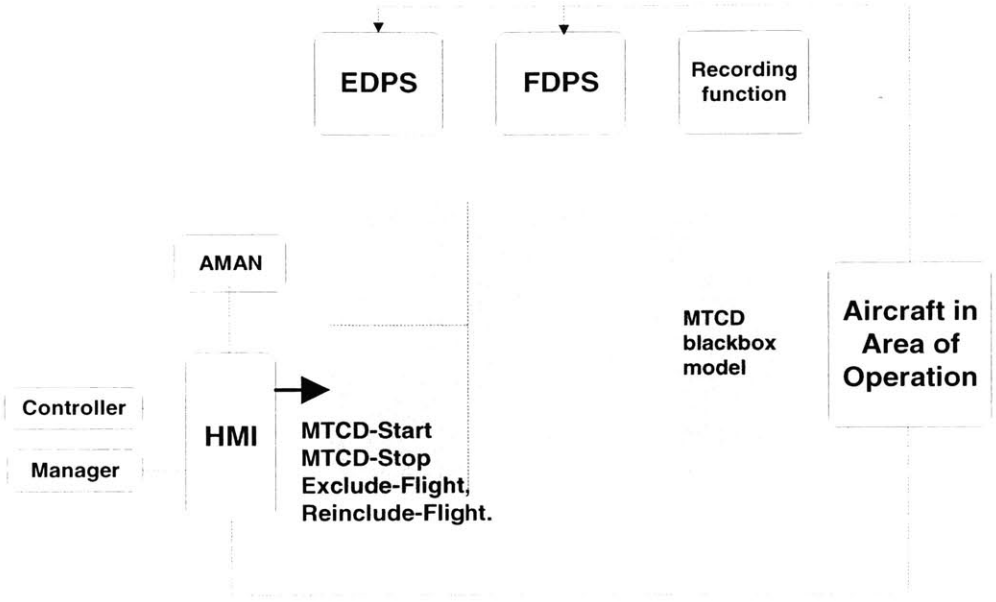

Figure 3-7: Inputs from the HMI to MTCD

MTCD-START

INPUT MESSAGE

- **Source:** HMI
- **Description:** MTCD is started using a MTCD-START message. The message contains the configuration.
- **References:** Supv-01 [2]
  - MTCD-START(
    - Configuration-Id)

MTCD-STOP

INPUT MESSAGE

- **Source:** HMI
- **Description:** MTCD is stopped using the MTCD-STOP message
- **References:** Supv-01
  - MTCD-STOP( )

[2]Here the link should indeed be made with the Task Analysis of level 2 by Daouk [8].

DISPLAY-CONFIGURATION

INPUT MESSAGE

- **Source:** HMI
- **Description:** The controller display is configure using the DISPLAY-CONFIGURATION message.
  - The message contents are TBD

EXCLUDE-FLIGHT

INPUT MESSAGE

- **Source:** HMI
- **Description:** Flights can be excluded from conflict detection using the EXCLUDE-FLIGHT message
- **References:** Op-02
  - EXCLUDE-FLIGHT(
    * Flight-Id)

REINCLUDE-FLIGHT

INPUT MESSAGE

- **Source:** HMI
- **Description:** Flights can be re-included for conflict detection using the REINCLUDE-FLIGHT message
- **References:** Op-03
  - REINCLUDE-FLIGHT(
    * Flight-Id)

3.7 Inputs FDPS -> MTCD

In this section we give the inputs sent from FDPS to MTCD.

Flight[I]-Flight ID

INPUT

- **Source:** FDPS
- **Arrival:** Dynamic based on flights entering the system and controller information needs.
- **Type:** String
- **Expected Range:** Unspecified
- **Granularity:** Unspecified
- **Units:** Unspecified
- **Load:** Unspecified
- **Exception Handling:** Unspecified
- **Obsolescence:** The value becomes obsolete when the Flight leaves the Area of Operations or when MTCD is powered off
- **Description:** The value is sent by the FDPS to MTCD specifying the Flight-ID of the aircraft. This information is provided with the NEW-FLIGHT message
- **References:** Env-As-FDPS-01

Definition

=**FIELD** (Flight ID in NEW FLIGHT message)

Received NEW-FLIGHT	T
Previous Flight ID = Obsolete	T

= **Previous** Flight-ID

Received LEFT-AOO	F	T
LEFT-AOO.Flight-Id = IP-Flight[I].Flight-Id	-	F

= **Obsolete**

Received MTCD-STOP	T	-
Received LEFT-AOO	-	T
LEFT-AOO.Flight-Id = IP-Flight[I].Flight-Id	-	T

Flight[I]-Aircraft type

INPUT

- **Source:** FDPS
- **Arrival:** Dynamic based on flights entering the system and controller information needs.
- **Type:** Enumerated
- **Expected Range:** (A320, A340, B744, B742, B777, B737)
- **Granularity:** N/A
- **Units:** N/A
- **Load:** Unspecified
- **Exception Handling:** Unspecified
- **Obsolescence:** The value becomes obsolete when the Flight leaves the Area of Operations or when MTCD is powered off
- **Description:** The value is sent by the FDPS to MTCD specifying the type of the aircraft. This information is provided with the NEW-FLIGHT message and can be modified by the CHANGE-FLIGHT-DATA
- **References:** Env-As-FDPS-01

Definition

=FIELD (Aircraft-type in NEW-FLIGHT message)

Received NEW-FLIGHT	T
NEW-FLIGHT . Flight-Id = IP-Flight[I].Flight-Id	T

=FIELD (Aircraft-type in CHANGED-FLIGHT-DATA message)

Received CHANGED-FLIGHT-DATA	T
CHANGED-FLIGHT-DATA. Flight-Id = IP-Flight[I].Flight-Id	T

= Previous Aircraft-Type

Received LEFT-AOO	F	T
LEFT-AOO.Flight-Id = IP-Flight[I].Flight-Id	-	F

= Obsolete

Received MTCD-STOP	T	-
Received LEFT-AOO	-	T
LEFT-AOO.Flight-ID = IP-Flight[I].Flight-ID	-	T

Flight[I]-Navigational Capabilities

INPUT

- **Source:** FDPS
- **Arrival:** Dynamic based on flights entering the system and controller information needs.
- **Type:** Enumerated
- **Expected Range:** (INS, GPS, IGS, BAS)
- **Granularity:** N/A
- **Units:** N/A
- **Load:** Unspecified
- **Exception Handling:** Unspecified
- **Obsolescence:** The value becomes obsolete when the Flight leaves the Area of Operations or when MTCD is powered off
- **Description:** The value is sent by the FDPS to MTCD specifying the navigational capabilities of the aircraft. This information is provided with the NEW-FLIGHT message and can be modified by the CHANGE-FLIGHT-DATA
- **Comments::** The value is computed based on the aircraft type and the onboard system. It can be treated as an enumerated set defining Inertial Navigation, Global Positioning, Integrated systems or basic instruments.
- **References:** Env-As-FDPS-01

Definition

=FIELD (Navigational Capabilities in NEW-FLIGHT message)

Received NEW-FLIGHT	T
NEW-FLIGHT . Flight-Id = IP-Flight[I].Flight-Id	T

=FIELD (Navigation-Capabilities in CHANGED-FLIGHT-DATA message)

Received CHANGED-FLIGHT-DATA	T
CHANGED-FLIGHT-DATA. Flight-Id = IP-Flight[I].Flight-Id	T

= Previous Navigation-Capabilities

Received LEFT-AOO	F	T
LEFT-AOO.Flight-Id = IP-Flight[I].Flight-Id	-	F

= Obsolete

Received MTCD-STOP	T	-
Received LEFT-AOO	-	T
LEFT-AOO.Flight-ID = IP-Flight[I].Flight-ID	-	T

Flight[I]-Class of flight

INPUT

- **Source:** FDPS
- **Arrival:** Dynamic based on flights entering the system and controller information needs.
- **Type:** Enumerated
- **Expected Range:** (Large, Medium, Small)(Passenger, Military, Cargo)
- **Granularity:** N/A
- **Units:** N/A
- **Load:** Unspecified
- **Exception Handling:** Unspecified
- **Obsolescence:** The value becomes obsolete when the Flight leaves the Area of Operations or when MTCD is powered off
- **Description:** The value is sent by the FDPS to MTCD specifying the Class-of-Flight of the aircraft. This information is provided with the NEW-FLIGHT message and can be modified by the CHANGE-FLIGHT-DATA
- **Comments::** The classification of class-of-flight is based on the type of aircraft and type of service that the flight provides.
- **References:** Env-As-FDPS-01

Definition

=**FIELD** (Class of flight in NEW-FLIGHT message)

Received NEW-FLIGHT	T
NEW-FLIGHT . Flight-Id = IP-Flight[I].Flight-Id	T

=**FIELD** (Class of flight in CHANGED-FLIGHT-DATA message)

Received CHANGED-FLIGHT-DATA	T
CHANGED-FLIGHT-DATA. Flight-Id = IP-Flight[I].Flight-Id	T

= **Previous** Class-of-flight

Received LEFT-AOO	F	T
LEFT-AOO.Flight-Id = IP-Flight[I].Flight-Id	-	F

= **Obsolete**

Received MTCD-STOP	T	-
Received LEFT-AOO	-	T
LEFT-AOO.Flight-ID = IP-Flight[I].Flight-ID	-	T

Flight[I]-Current position

INPUT

Source: FDPS

Arrival: Dynamic based on flights entering the system and controller information needs.

Type: Real

Expected Range: (0-360, 0-360)

Granularity: 1 minute

Units: Degrees

Load: Unspecified

Exception Handling: Unspecified

Obsolescence: The value becomes obsolete when: the flight leaves the Area of Operations, MTCD is powered off, and the position information is not updated within some time.

Description: The value is sent by the FDPS to MTCD specifying the current position of the aircraft. This information is provided with the NEW-FLIGHT message and can be modified by the CHANGE-FLIGHT-DATA.

Comments:: The position is derived from surveillance information or data link. The granularity is determined by the source of the information and will change in accuracy depending on the user.

References: Env-As-FDPS-01

Definition:

=FIELD (Current position in NEW-FLIGHT message)

Received NEW-FLIGHT	T
NEW-FLIGHT . Flight-Id = IP-Flight[I].Flight-Id	T

=FIELD (Current-position in CURRENT-POSITION message)

Received CURRENT-POSITION	T
CURRENT-POSITION. Flight-Id = IP-Flight[I].Flight-Id	T

= Previous Current-Position

Received LEFT-AOO	F	T	-
LEFT-AOO.Flight-Id = IP-Flight[I].Flight-Id	-	F	-
Time Received CURRENT-POSITION > Time-between-update(TBD)	-	-	T

= Obsolete

Received MTCD-STOP	T	-	-
Received LEFT-AOO	-	T	-
LEFT-AOO.Flight-ID = IP-Flight[I].Flight-ID	-	T	-
Time Received CURRENT-POSITION > Time-between-update(TBD)	-	-	T

Trajectory[I]-Trajectory-Id

INPUT

- **Source:** FDPS
- **Arrival:** Dynamic based on flights entering the system and controller information needs.
- **Type:** String
- **Expected Range:** Unspecified
- **Granularity:** Unspecified
- **Units:** Unspecified
- **Load:** Unspecified
- **Exception Handling:** Unspecified
- **Obsolescence:** The value becomes obsolete when a trajectory is deleted or the flight leaves the Area of Operations
- **Description:** The value is sent by the FDPS to MTCD specifying the trajectory id. This information is provided by the NEW-TRAJECTORY message
- **Comments:** The trajectory id is also unique to the entire system.
- **References:** Env-As-FDPS-01, Env-As-FDPS-04, Env-As-FDPS-05

Definition:
=FIELD (Trajectory-Id in NEW-TRAJECTORY message)

Received NEW-TRAJECTORY	T
Previous Trajectory-Id = Obsolete	T

= Previous Trajectory-Id

Received LEFT-AOO	F	-
Received DELETE-TRAJECTORY	-	T
DELETE-TRAJECTORY.Trajectory-Id = IP-Trajectory[I].Trajectory-Id	-	T

= Obsolete

Received MTCD-STOP	T	-	-
Received LEFT-AOO	-	T	-
LEFT-AOO.Flight-ID = IP-Flight[I].Flight-ID	-	T	-
Received DELETE-TRAJECTORY	-	-	T
DELETE-TRAJECTORY.Trajectory-Id =IP-Trajectory[I].Trajectory-Id	-	-	T

Trajectory[I]-Flight-Id

INPUT

- **Source:** FDPS
- **Arrival:** Dynamic based on flights entering the system and controller information needs.
- **Type:** String
- **Expected Range:** Unspecified
- **Granularity:** Unspecified
- **Units:** Unspecified
- **Load:** Unspecified
- **Exception Handling:** Unspecified
- **Obsolescence:** The value becomes obsolete when a trajectory is deleted or the flight leaves the Area of Operations
- **Description:** The value is sent by the FDPS to MTCD specifying the Flight-Id associated with a trajectory. This information is provided by the NEW-TRAJECTORY message.
- **Comments:** The flight id has to be defined earlier in a NEW-FLIGHT message. We capture that dependency here also.
- **References:** Env-As-FDPS-01, Env-As-FDPS-04, Env-As-FDPS-05

Definition:
=**FIELD** (Flight-Id in NEW-TRAJECTORY message)

Received NEW-TRAJECTORY	T
IP-Trajectory[I].Trajectory-Id = NEW-TRAJECTORY.Trajectory-Id	T

= **Previous** Flight-Id

Received LEFT-AOO	F	T	F
LEFT-AOO.Flight-Id = IP-Trajectory[I].Flight-Id	-	F	-
Received DELETE-TRAJECTORY	F	F	T
DELETE-TRAJECTORY.Trajectory-Id = IP-Trajectory[I].Trajectory-Id	-	-	F

= **Obsolete**

Received MTCD-STOP	T	-	-
Received LEFT-AOO	-	T	-
LEFT-AOO.Flight-ID = IP-Flight[I].Flight-ID	-	T	-
Received DELETE-TRAJECTORY	-	-	T
DELETE-TRAJECTORY.Trajectory-Id =IP-Trajectory[I].Trajectory-Id	-	-	T

Trajectory[I]-Tentative-indication

INPUT

Source: FDPS

Arrival: Dynamic based on flights entering the system and controller information needs.

Type: Enumerated

Expected Range: (Tentative, System)

Granularity: N/A

Units: N/A

Load: Unspecified

Exception Handling: Unspecified

Obsolescence: The value becomes obsolete when a trajectory is deleted or the flight leaves the Area of Operations

Description: The value is sent by the FDPS to MTCD specifying the type of the trajectory. This information is provided by the NEW-TRAJECTORY message. It can be modified using the RECALCULATED-TRAJECTORY.

References: Env-As-FDPS-01

Definition:

=**FIELD** (Tentative in NEW-TRAJECTORY message)

Received NEW-TRAJECTORY	T
IPrevious Trajectory-Id = Obsolete	T

=**FIELD** (Tentative in RECALCULATED-TRAJECTORY message)

Received NEW-TRAJECTORY	T
RECALCULATED-TRAJECTORY.Trajectory-Id = IP-Trajectory[I].Trajectory-Id	T

= **Previous** Tentative-Indication

Received LEFT-AOO	F	T	F
LEFT-AOO.Flight-Id = IP-Trajectory[I].Flight-Id	-	F	-
Received DELETE-TRAJECTORY	F	F	T
DELETE-TRAJECTORY.Trajectory-Id = IP-Trajectory[I].Trajectory-Id	-	-	F

= **Obsolete**

Received MTCD-STOP	T	-	-
Received LEFT-AOO	-	T	-
LEFT-AOO.Flight-ID = IP-Flight[I].Flight-ID	-	T	-
Received DELETE-TRAJECTORY	-	-	T
DELETE-TRAJECTORY.Trajectory-Id =IP-Trajectory[I].Trajectory-Id	-	-	T

TRAJECTORY[I]-Phase-of-flight

INPUT

- **Source:** FDPS
- **Arrival:** Dynamic based on flights entering the system and controller information needs.
- **Type:** Enumerated
- **Expected Range:** (Climb, Descent, Cruise...)
- **Granularity:** N/A
- **Units:** N/A
- **Load:** Unspecified
- **Exception Handling:** Unspecified
- **Obsolescence:** The value becomes obsolete when the Flight leaves the Area of Operations or when MTCD is powered off
- **Description:** The value is sent by the FDPS to MTCD specifying the Phase-of-Flight of the aircraft. This information is provided with the NEW-TRAJECTORY message and can be modified by the CHANGE-TRAJECTORY-DATA
- **References:** Env-As-FDPS-01

Definition:

=FIELD (Phase of flight in NEW-FLIGHT message)

Received NEW-FLIGHT	T
NEW-FLIGHT . Flight-Id = IP-Flight[I].Flight-Id	T

=FIELD (Phase of flight in CHANGED-FLIGHT-DATA message)

Received CHANGED-FLIGHT-DATA	T
CHANGED-FLIGHT-DATA. Flight-Id = IP-Flight[I].Flight-Id	T

= Previous Phase-of-flight

Received LEFT-AOO	F	T
LEFT-AOO.Flight-Id = IP-Flight[I].Flight-Id	-	F

= Obsolete

Received MTCD-STOP	T	-
Received LEFT-AOO	-	T
LEFT-AOO.Flight-ID = IP-Flight[I].Flight-ID	-	T

Trajectory[I]-Nominal-Route

INPUT

Source: FDPS

Arrival: Dynamic based on flights entering the system and controller information needs.

Type: Enumerated

Expected Range: (????)

Granularity: N/A

Units: N/A

Load: Unspecified

Exception Handling: Unspecified

Obsolescence: The value becomes obsolete when a trajectory is deleted or the flight leaves the Area of Operations

Description: The value is sent by the FDPS to MTCD specifying the type of the trajectory. This information is provided by the NEW-TRAJECTORY message. It can be modified using the RECALCULATED-TRAJECTORY.

Definition:

=**FIELD** (Nominal-Route in NEW-TRAJECTORY message)

Received NEW-TRAJECTORY	T
IPrevious Trajectory-Id = Obsolete	T

=**FIELD** (Nominal-Route in RECALCULATED-TRAJECTORY message)

Received NEW-TRAJECTORY	T
RECALCULATED-TRAJECTORY.Trajectory-Id = IP-Trajectory[I].Trajectory-Id	T

= **Previous** Nominal-Route

Received LEFT-AOO	F	T	F
LEFT-AOO.Flight-Id = IP-Trajectory[I].Flight-Id	-	F	-
Received DELETE-TRAJECTORY	F	F	T
DELETE-TRAJECTORY.Trajectory-Id = IP-Trajectory[I].Trajectory-Id	-	-	F

= **Obsolete**

Received MTCD-STOP	T	-	-
Received LEFT-AOO	-	T	-
LEFT-AOO.Flight-ID = IP-Flight[I].Flight-ID	-	T	-
Received DELETE-TRAJECTORY	-	-	T
DELETE-TRAJECTORY.Trajectory-Id =IP-Trajectory[I].Trajectory-Id	-	-	T

Left-AOO[I]-Flight-Id

INPUT

- **Source:** FDPS
- **Arrival:** Dynamic depends on the flight location.
- **Type:** String
- **Expected Range:** Unspecified
- **Granularity:** Unspecified
- **Units:** Unspecified
- **Load:** Unspecified
- **Exception Handling:** Unspecified
- **Obsolescence:** The value becomes obsolete when the operation times out.
- **Description:** The value is sent by the FDPS to MTCD specifying the flight-id of the aircraft that left the area of operations
- **References:** Env-As-FDPS-01

Definition

=**FIELD** (Flight-Id in Left-AOO message)

Received Left-AOO	T
Previous Flight ID = Obsolete	T

= **Previous** Flight-ID

Received STOP-MTCD	F
Time Received Left-AOO > 800 milliseconds	F

= **Obsolete**

Received MTCD-STOP	T	-
Time Received Left-AOO > 800 milliseconds	-	T

Delete-Trajectory[I]-Trajectory-Id

INPUT

- **Source:** FDPS
- **Arrival:** Dynamic depends on the controller workload and information needs.
- **Type:** String
- **Expected Range:** Unspecified
- **Granularity:** Unspecified
- **Units:** Unspecified
- **Load:** Unspecified
- **Exception Handling:** Unspecified
- **Obsolescence:** The value becomes obsolete when the operation times out
- **Description:** The value is sent by the FDPS to MTCD specifying the trajectory-id of the trajectory to be deleted
- **References:** Env-As-FDPS-04

Definition

=**FIELD** (Trajectory-Id in Delete-Trajectory message)

Received Delete-Trajectory	T
Previous Trajectory-Id = Obsolete	T

= **Previous** Trajectory-Id

Received MTCD-STOP	F
Time Received Delete-Trajectory > 800 milliseconds	F

= **Obsolete**

Received MTCD-STOP	T	-
Time Received Delete-Trajectory > 800 milliseconds	-	T

3.8 Inputs EDPS -> MTCD

In this section we give the inputs from the EDPS sent to MTCD.

AOO-Airspace

INPUT

- **Source:** EDPS
- **Arrival:** Defined at MTCD configuration time
- **Type:** List of Real
- **Expected Range:** (0-359deg 59', 0-359deg 59')
- **Granularity:** 1 minute
- **Units:** Unspecified
- **Load:** Unspecified
- **Exception Handling:** Unspecified
- **Obsolescence:** The airspace becomes obsolete only when MTCD is stopped.
- **Description:** The value is sent by the EDPS to MTCD specifying the list of airspaces in the AOO.
- **Comments:** The airspace list consists of a list of points defining the area of operations.
- **References:** Env-As-EDPS-01

Definition

=**FIELD** (Airspace in Area-Of-Operations message)

Received Area-of-Operations	T

= **Previous** AOO-Airspace

Received MTCD-STOP	F

= **Obsolete**

Received MTCD-STOP	T

AOO-Default-Separation

INPUT

- **Source:** EDPS
- **Arrival:** Defined at MTCD configuration time
- **Type:** Unspecified
- **Expected Range:** Unspecified
- **Granularity:** Unspecified
- **Units:** Unspecified
- **Load:** Unspecified
- **Exception Handling:** Unspecified
- **Obsolescence:** The default separation criteria becomes obsolete only when MTCD is stopped.
- **Description:** The value is sent by the EDPS to MTCD specifying the default separation criteria to use in the area of operations.
- **References:** Env-As-EDPS-01

Definition

=**FIELD** (Default-Separation in Area-Of-Operations message)

Received Area-of-Operations	T

= **Previous** IP-AOO-Default-Separation

Received MTCD-STOP	F

= **Obsolete**

Received MTCD-STOP	T

AOO-Default-Uncertainty

INPUT

- **Source:** EDPS
- **Arrival:** Defined at MTCD configuration time
- **Type:** Unspecified
- **Expected Range:** Unspecified
- **Granularity:** Unspecified
- **Units:** Unspecified
- **Load:** Unspecified
- **Exception Handling:** Unspecified
- **Obsolescence:** The default uncertainty criteria becomes obsolete only when MTCD is stopped.
- **Description:** The value is sent by the EDPS to MTCD specifying the default uncertainty area to use in the area of operations
- **References:** Env-As-EDPS-01

Definition

=**FIELD** (Default-Uncertainty in Area-Of-Operations message)

Received Area-of-Operations	T

= **Previous** IP-AOO- Default-Uncertainty

Received MTCD-STOP	F

= **Obsolete**

Received MTCD-STOP	T

AOO-Default-Lowest-Level

INPUT

- **Source:** EDPS
- **Arrival:** Defined at MTCD configuration time
- **Type:** Unspecified
- **Expected Range:** Unspecified
- **Granularity:** Unspecified
- **Units:** Unspecified
- **Load:** Unspecified
- **Exception Handling:** Unspecified
- **Obsolescence:** The default Lowest-Level criteria becomes obsolete only when MTCD is stopped
- **Description:** The value is sent by the EDPS to MTCD specifying the default lowest level to use in the area of operations
- **References:** Env-As-EDPS-01

Definition

=**FIELD** (Default-Lowest-Level in Area-Of-Operations message)

Received Area-of-Operations	T

= **Previous** AOO-Default-Lowest-Level

Received MTCD-STOP	F

= **Obsolete**

Received MTCD-STOP	T

AOO-Airspace[I].Airspace-Id

INPUT

- **Source:** EDPS
- **Arrival:** Defined at MTCD configuration time
- **Type:** Unspecified
- **Expected Range:** Unspecified
- **Granularity:** Unspecified
- **Units:** Unspecified
- **Load:** Unspecified
- **Exception Handling:** Unspecified
- **Obsolescence:** The Airspace id becomes obsolete only when MTCD is stopped.
- **Description:** The value is sent by the EDPS to MTCD specifying the airspace id when the area of operation is specified.
- **References:** Env-As-EDPS-01

Definition

=**FIELD** (Airspace-Id in Airspace-List of Area-Of-Operations message)

Received Area-of-Operations	T
Airspace[I].Airspace-Id = obsolete	T

= Obsolete

Received MTCD-STOP	T

AOO-Airspace[I].Airspace-points

INPUT

- **Source:** EDPS
- **Arrival:** Defined at MTCD configuration time
- **Type:** List of Real
- **Expected Range:** (0-359deg 59', 0-359deg 59')
- **Granularity:** 1 minute
- **Units:** Degrees
- **Load:** Unspecified
- **Exception Handling:** Unspecified
- **Obsolescence:** The Airspace-points are obsolete only when MTCD is stopped
- **Description:** The value is sent by the EDPD to MTCD specifying the list of points specifying the airspace in the list of airspaces, when the area of operations is specified.
- **Comments:** The points are specified as a list of latitude, longitude pairs.
- **References:** Env-As-EDPS-01

Definition

=FIELD (Airspace-Points in Airspace-List of Area-Of-Operations message)

Received Area-of-Operations	T
Airspace[I].Airspace-Id = obsolete	T

= Obsolete

Received MTCD-STOP	T

AOO-Airspace[I].SUA-Start

INPUT

- **Source:** EDPS
- **Arrival:** Defined at MTCD configuration time or dynamically by the EDPS
- **Type:** Real
- **Expected Range:** (00:00:00 - 23:59:59)
- **Granularity:** 1 second
- **Units:** hour, minute, second
- **Load:** Unspecified
- **Exception Handling:** Unspecified
- **Obsolescence:** The Special Use airspace criteria is obsolete only when MTCD is stopped.
- **Description:** The value is sent by the EDPD to MTCD specifying the start time of special use airspace criteria.
- **References:** Env-As-EDPS-01

Definition

=FIELD (SUA-Start in Special-Use-Airspace message)

Received Area-of-Operations	T
Airspace[I].Airspace-Id = obsolete	T

= Previous SUA-Start

Received MTCD-STOP	F

= Obsolete

Received MTCD-STOP	T

AOO-Airspace[I].SUA-Stop

INPUT

- **Source:** EDPS
- **Arrival:** Defined at MTCD configuration time or dynamically by the EDPS
- **Type:** Real
- **Expected Range:** (00:00:00 - 23:59:59)
- **Granularity:** 1 second
- **Units:** hour, minute, second
- **Load:** Unspecified
- **Exception Handling:** Unspecified
- **Obsolescence:** The Special Use airspace criteria is obsolete only when MTCD is stopped.
- **Description:** The value is sent by the EDPD to MTCD specifying the stop time of special use airspace criteria.
- **References:** Env-As-EDPS-01

Definition

=**FIELD** (SUA-Stop in Special-Use-Airspace message)

Received Area-of-Operations	T
Airspace[I].Airspace-Id = obsolete	T

= **Previous** SUA-Start

Received MTCD-STOP	F

= **Obsolete**

Received MTCD-STOP	T

AOO-Airspace[I].Separation-Criteria

INPUT

- **Source:** EDPS
- **Arrival:** Defined at MTCD configuration time or dynamically by the EDPS
- **Type:** Unspecified
- **Expected Range:** Unspecified
- **Granularity:** Unspecified
- **Units:** Unspecified
- **Load:** Unspecified
- **Exception Handling:** Unspecified
- **Obsolescence:** The separation criteria airspace becomes obsolete only when MTCD is stopped.
- **Description:** The value is sent by the EDPD to MTCD specifying the separation criteria for the airspace, which has been defined.
- **References:** Env-As-EDPS-01

Definition

=**FIELD** (Separation-Criteria in Separation-Criteria message)

Received Separation-Criteria	T
Airspace[I].Airspace-Id = Separation-Criteria.Airspace-Id	T

= **Previous** Separation-Criteria

Received MTCD-STOP	F

= **Obsolete**

Received MTCD-STOP	T

AOO-Airspace[I].Uncertainty-Criteria

INPUT

- **Source:** EDPS
- **Arrival:** Defined at MTCD configuration time or dynamically by the EDPS
- **Type:** Unspecified
- **Expected Range:** Unspecified
- **Granularity:** Unspecified
- **Units:** Unspecified
- **Load:** Unspecified
- **Exception Handling:** Unspecified
- **Obsolescence:** The uncertainty airspace becomes obsolete only when MTCD is stopped.
- **Description:** The value is sent by the EDPD to MTCD specifying the uncertainty parameters for the airspace, which has been defined.
- **References:** Env-As-EDPS-01

Definition

=**FIELD** (Uncertainty-Criteria in Separation-Criteria message)

Received Uncertainty-Airspace	T
Airspace[I].Airspace-Id = Uncertainty-Airspace.Airspace-Id	T

= **Previous** Uncertainty-Criteria

Received MTCD-STOP	F

= **Obsolete**

Received MTCD-STOP	T

AOO-Airspace[I].Lowest-Flight-Level

INPUT

- **Source:** EDPS
- **Arrival:** Defined at MTCD configuration time or dynamically by the EDPS
- **Type:** Integer
- **Expected Range:** 0-36000
- **Granularity:** 1 foot
- **Units:** Unspecified
- **Load:** Unspecified
- **Exception Handling:** Unspecified
- **Obsolescence:** The Lowest-Flight-Level criterion airspace becomes obsolete only when MTCD is stopped.
- **Description:** The value is sent by the EDPD to MTCD specifying the lowest flight level for a defined airspace
- **References:** Env-As-EDPS-01

Definition

=**FIELD** (Level in Lowest-Flight-Level message)

Received Lowest-Flight-Level	T
Airspace[I].Airspace-Id = Lowest-Flight-Level.Airspace-Id	T

= **Previous** Lowest-Flight-Level

Received MTCD-STOP	F

= **Obsolete**

Received MTCD-STOP	T

POF-Phase

INPUT

- **Source:** EDPS
- **Arrival:** Defined at MTCD configuration time.
- **Type:** Enumerated
- **Expected Range:** (Climb, Descent, Cruise...)
- **Granularity:** N/A
- **Units:** N/A
- **Load:** Unspecified
- **Exception Handling:** Unspecified
- **Obsolescence:** The Phase of Flight criteria becomes obsolete only when MTCD is stopped.
- **Description:** The value is sent by the EDPS to MTCD specifying the Phase for which the separation criteria and uncertainty parameters have to be defined
- **Comments:**The classification is the same as used for the class of flight for an aircraft. During conflict detection, the separation criteria and uncertainty criteria used is determined by the phase of flight
- **References:** Env-As-EDPS-01, Env-As-EDPS-04

Definition

=**FIELD** (Phase in Phase-Of-Flight message)

Received Phase-Of-Flight	T

= **Previous** Phase

Received MTCD-STOP	F

= **Obsolete**

Received MTCD-STOP	T

POF-Criteria

INPUT

- **Source:** EDPS
- **Arrival:** Defined at MTCD configuration time.
- **Type:** Unspecified
- **Expected Range:** Unspecified
- **Granularity:** Unspecified
- **Units:** Unspecified
- **Load:** Unspecified
- **Exception Handling:** Unspecified
- **Obsolescence:** The Phase of Flight criteria becomes obsolete only when MTCD is stopped.
- **Description:** The value is sent by the EDPD to MTCD specifying the separation criteria for the phase of flight defined
- **References:** Env-As-EDPS-01

Definition

=**FIELD** (Criteria in Phase-Of-Flight message)

Received Phase-Of-Flight	T

= **Previous** Criteria

Received MTCD-STOP	F

= **Obsolete**

Received MTCD-STOP	T

POF-Parameters

INPUT

- **Source:** EDPS
- **Arrival:** Defined at MTCD configuration time.
- **Type:** Unspecified
- **Expected Range:** Unspecified
- **Granularity:** Unspecified
- **Units:** Unspecified
- **Load:** Unspecified
- **Exception Handling:** Unspecified
- **Obsolescence:** The Phase of Flight criteria becomes obsolete only when MTCD is stopped.
- **Description:** The value is sent by the EDPD to MTCD specifying the Uncertainty parameters for the phase of flight defined.
- **References:** Env-As-EDPS-01

Definition

=**FIELD** (Parameters in Phase-Of-Flight message)

Received Phase-Of-Flight	T

= **Previous** Parameters

Received MTCD-STOP	F

= **Obsolete**

Received MTCD-STOP	T

COF-Class

INPUT

- **Source:** EDPS
- **Arrival:** Defined at MTCD configuration time.
- **Type:** Enumerated
- **Expected Range:** (Large, Medium, Small)(Passenger, Military, Cargo)
- **Granularity:** N/A
- **Units:** N/A
- **Load:** Unspecified
- **Exception Handling:** Unspecified
- **Obsolescence:** The Class of Flight criteria becomes obsolete only when MTCD is stopped.
- **Description:** The value is sent by the EDPD to MTCD specifying the class of flight for which the separation criteria has to be defined.
- **Comments:** The classification is the same as used for the class of flight for an aircraft. During conflict detection, the separation criteria used is determined by both the class of flight and phase of flight.
- **References:** Env-As-EDPS-01

Definition

=**FIELD** (Class in Class-Of-Flight message)

Received Class-Of-Flight	T

= **Previous** Class

Received MTCD-STOP	F

= **Obsolete**

Received MTCD-STOP	T

COF-Criteria

INPUT

- **Source:** EDPS
- **Arrival:** Class of flight parameters are specified during MTCD configuration phase.
- **Type:** Unspecified
- **Expected Range:** Unspecified
- **Granularity:** Unspecified
- **Units:** Unspecified
- **Load:** Unspecified
- **Exception Handling:** Unspecified
- **Obsolescence:** The Class of Flight criteria becomes obsolete only when MTCD is stopped.
- **Description:** The value is sent by the EDPS to MTCD specifying the separation criteria for the class of flight defined.
- **References:** Env-As-EDPS-01

Definition

=**FIELD** (Criteria in Class-Of-Flight message)

Received Class-Of-Flight	T

= **Previous** IP-COF-Criteria

Received MTCD-STOP	F

= **Obsolete**

Received MTCD-STOP	T

NAV-Equipment

INPUT

- **Source:** EDPS
- **Arrival:** Navigation equipment is specified at MTCD configuration phase.
- **Type:** Enumerated
- **Expected Range:** (INS, GPS, IGS, BAS...)
- **Granularity:** N/A
- **Units:** N/A
- **Load:** Unspecified
- **Exception Handling:** Unspecified
- **Obsolescence:** It becomes obsolete only when MTCD is stopped.
- **Description:** The value is sent by the EDPS to MTCD specifying the on aircraft equipment for which the uncertainty parameters have to be defined.
- **Comments:** The classification is the same as that used for Navigation-Capabilities of a flight.
- **References:** Env-As-EDPS-01

Definition

=**FIELD** (Equipment in Navigation-Capabilities message)

Received Navigation-Capabilities	T

= **Previous** NAV-Equipment

Received MTCD-STOP	F

= **Obsolete**

Received MTCD-STOP	T

NAV-Parameters

INPUT

- **Source:** EDPS
- **Arrival:** Navigation parameters are specified at MTCD configuration phase.
- **Type:** Unspecified
- **Expected Range:** Unspecified
- **Granularity:** Unspecified
- **Units:** Unspecified
- **Load:** Unspecified
- **Exception Handling:** Unspecified
- **Obsolescence:** TThe uncertainty parameters associated with navigation, become obsolete only when MTCD is stopped.
- **Description:** The value is sent by the EDPS to MTCD specifying the Uncertainty parameters for the navigation capability defined.
- **References:** Env-As-EDPS-01

Definition

=**FIELD** (Parameters in Navigation-Capabilities message)

Received Navigation-Capabilities	T

= **Previous** Parameters

Received MTCD-STOP	F

= **Obsolete**

Received MTCD-STOP	T

PAR-Route

INPUT

- **Source:** EDPS
- **Arrival:** Specified at MTCD configuration phase.
- **Type:** List of Real points
- **Expected Range:** Unspecified
- **Granularity:** 1 minute
- **Units:** Degree
- **Load:** Unspecified
- **Exception Handling:** Unspecified
- **Description:** The value is sent by the EDPS to MTCD specifying the parallel ATS routes for which the separation criteria have to be defined.
- **Comments:** The routes are specified as lists of latitude and longitude points in the Area of Operations.
- **References:** Env-As-EDPS-01

Definition

=**FIELD** (Route in Parallel-ATS-Route message)

Received Parallel-ATS-Route	T

= **Previous** PAR-Route

Received MTCD-STOP	F

= **Obsolete**

Received MTCD-STOP	T

PAR-Criteria

INPUT

- **Source:** EDPS
- **Arrival:** Specified at MTCD configuration phase.
- **Type:** Real
- **Expected Range:** Unspecified
- **Granularity:** Unspecified
- **Units:** Unspecified
- **Load:** Unspecified
- **Exception Handling:** Unspecified
- **Description:** The value is sent by the EDPS to MTCD specifying the separation criteria for the parallel ATS route defined.
- **References:** Env-As-EDPS-01

Definition

=**FIELD** (Criteria in Parallel-ATS-Route message)

Received Parallel-ATS-Route	T

= **Previous** PAR-Criteria

Received MTCD-STOP	F

= **Obsolete**

Received MTCD-STOP	T

Exclude-Airspace[I]-Airspace-Id

INPUT

- **Source:** EDPS
- **Arrival:** Specified at MTCD configuration phase.
- **Type:** String
- **Expected Range:** Unspecified
- **Granularity:** Unspecified
- **Units:** Unspecified
- **Load:** Unspecified
- **Exception Handling:** Unspecified
- Obsolescence The value becomes obsolete when MTCD-STOP is received.
- Description The value is sent by the EDPD to MTCD specifying the airspace in which conflict detection is not performed.
- **References:** Env-As-EDPS-01

Definition

=**FIELD** (Airspace in Exclude-Airspace message)

Received Exclude-Airspace	T
Previous Airspace = Obsolete	T

= **Previous** Airspace

Received MTCD-STOP	F

= **Obsolete**

Received MTCD-STOP	T

3.9 Inputs HMI –> MTCD

In this section we give the inputs from the HMI sent to MTCD.

Configuration

INPUT

- **Source:** HMI
- **Arrival:** Specified at MTCD configuration phase.
- **Type:** Integer
- **Expected Range:** Unspecified
- **Granularity:** N/A
- **Units:** N/A
- **Load:** Unspecified
- **Exception Handling:** Unspecified
- Obsolescence The value becomes obsolete when MTCD stops.
- Description The value is sent by the HMI to MTCD specifying the configuration.
- **References:** [2.18.2]

Definition

=**FIELD** (Configuration in MTCD-START message)

Received MTCD-START	T

= **Previous** Configuration

Received MTCD-STOP	F

= **Obsolete**

Received MTCD-STOP	T

Exclude-Flight[I]-Flight-Id

INPUT

- **Source:** HMI
- **Arrival:** Specified at MTCD configuration phase.
- **Type:** String
- **Expected Range:** Unspecified
- **Granularity:** Unspecified
- **Units:** Unspecified
- **Load:** Unspecified
- **Exception Handling:** Unspecified
- Obsolescence The value becomes obsolete when it exceeds the response time.
- Description The value is sent by the HMI to MTCD specifying which flight has to be excluded from conflict detection.
- **References:** [2.6.6]

Definition

=**FIELD** (Flight-Id in Exclude-Flight message)

Received Exclude-Flight	T
Previous Flight-Id = Obsolete	T

= **Previous** Flight-Id

Time Previous Flight-Id > 800 milliseconds	F

= **Obsolete**

Time Previous Flight-Id > 800 milliseconds	T

Reinclude-Flight[I]-Flight-Id

INPUT

- **Source:** HMI
- **Type:** String
- **Expected Range:** Unspecified
- **Granularity:** Unspecified
- **Units:** Unspecified
- **Load:** Unspecified
- **Exception Handling:** Unspecified
- Obsolescence The value becomes obsolete when it exceeds the response time.
- Description The value is sent by the HMI to MTCD specifying which flight has to be reincluded from conflict detection.
- **References:** [2.6.6]

Definition

=**FIELD** (Flight-Id in Reinclude-Flight message)

Received Reinclude-Flight	T
Previous Flight-Id = Obsolete	T

= **Previous** Flight-Id

Time Previous Flight-Id > 800 milliseconds	F

= **Obsolete**

Time Previous Flight-Id > 800 milliseconds	T

3.10 Inferred Airspace States

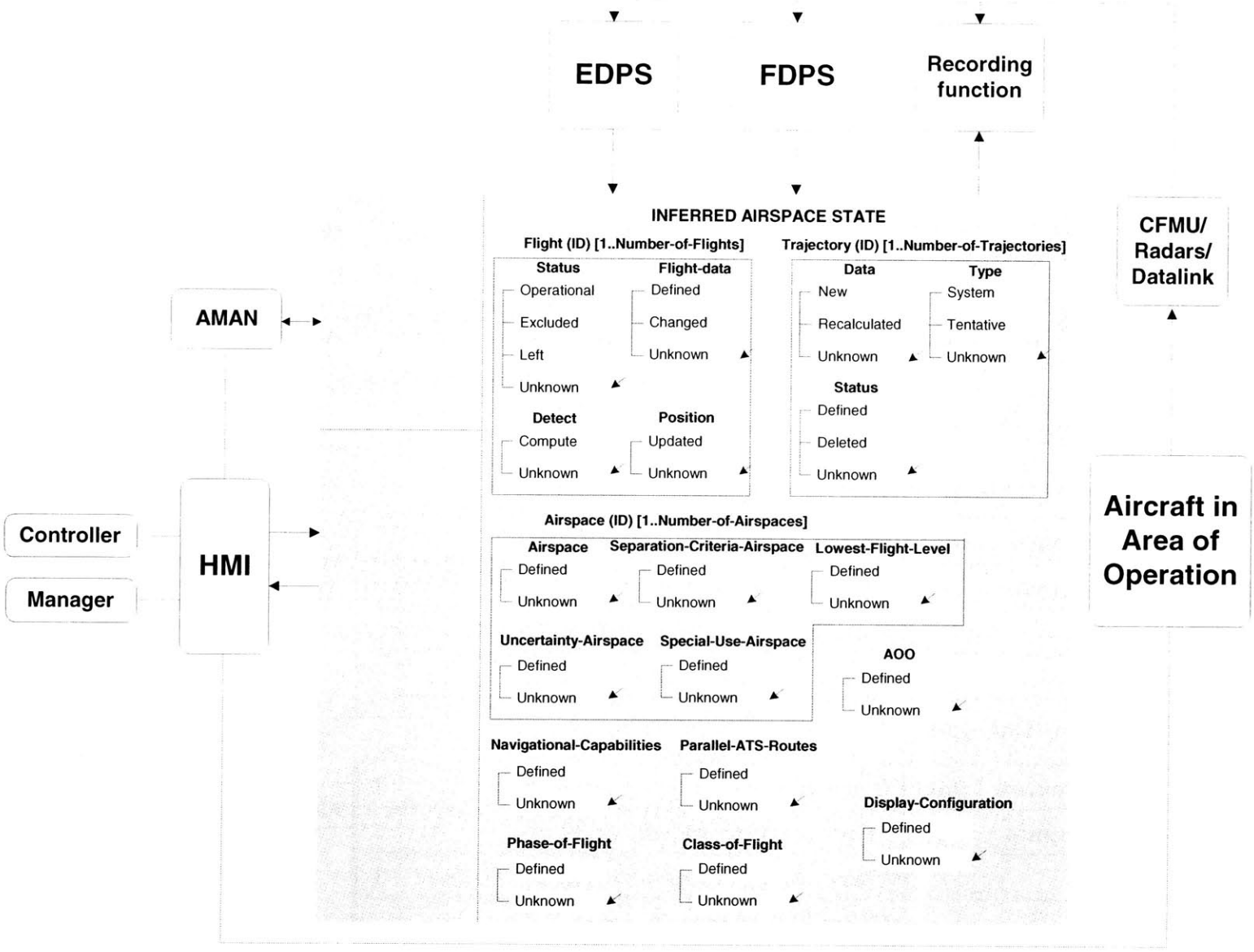

Figure 3-8: MTCD Inferred System States

Flight[I]-Flight Data

STATE

- **Description:** The Flight-Data state variable specifies whether data for the flight is specified.
- **Obsolesence:** It becomes obsolete when the flight leaves the area of operations. It starts in the unknown state and transitions into the unknown state when it becomes obsolete.
- **Exception Handling:** Not defined.
- **Comments:** Flight-Data can be updated by either the New-Flight or the Change-Flight-Data messages.
- **Definition:**

= Defined

Control Mode	Operational	T
Inputs	Received New-Flight	T
	New-Flight.Flight-Id = Flight[I].Flight-Id	T
	Flight[I].Flight-Id = Obsolete	F

= Changed

Control Mode	Operational	T
Inputs	Received Changed-Flight	T
	Changed-Flight.Flight-Id = Flight[I].Flight-Id	T
	Flight[I].Flight-Id = Obsolete	F

= Unknown

Control Mode	Operational	T
Inputs	Flight[I].Flight-Id = Obsolete	T

Flight[I]-Position

STATE

- **Description:** The Position state variable specifies whether current position for the flight has been updated.
- **Obsolesence:** It becomes obsolete when the current-position input becomes obsolete.
- **Exception Handling:** Not defined.
- **Comments:** Flight-Data can be updated by either the New-Flight or the Change-Flight-Data messages.
- **Comments:** The Position state variable is initialized via the New-Flight message and is updated by the Current-Position message.
- **Definition:**

= Defined

Control Mode	Operational	T
Inputs	Received New-Flight	T
	New-Flight.Flight-Id = Flight[I].Flight-Id	T
	Flight[I].Flight-Id = Obsolete	F

= Updated

Control Mode	Operational	T
Inputs	Received Current-Position	T
	Current-Position.Flight-Id = Flight[I].Flight-Id	T
	Time Received Current Position < UPDATE-TIME$_{TBD}$	T
	Flight[I].Flight-Id = Obsolete	F

= Unknown

Control Mode	Operational	T	T
Inputs	Time Received Current Position < UPDATE-TIME$_{TBD}$	F	-
	Flight[I].Flight-Id = Obsolete	-	T

Flight[I]-Status

STATE

- **Description:** The Status state variable specifies the status of the flight.
- **Obsolesence:** It becomes obsolete 500 milliseconds after the flight leaves the area of operations.
- **Exception Handling:** Not defined.
- **Comments:** It is determined by the New-Flight, Reinclude-Flight, Exclude-Flight and the Left-AOO messages.
- **Definition:**

= Operational

Control Mode	Operational	T	T
Inputs	Received New-Flight	T	-
	New-Flight.Flight-Id = Flight[I].Flight-Id	T	-
	Flight[I].Flight-Id = Obsolete	F	F
	Received Reinclude-Flight	-	T
	Reinclude-Flight.Flight-Id = Flight[I].Flight-Id	-	T

= Excluded

Control Mode	Operational	T
Inputs	Received Exclude-Flight	T
	Exclude-Flight.Flight-Id = Flight[I].Flight-Id	T
	Flight[I].Flight-Id = Obsolete	F

= Left

Control Mode	Operational	T
Inputs	Received Left-AOO	T
	Left-AOO.Flight-Id = Flight[I].Flight-Id	T
	Flight[I].Flight-Id = Obsolete	T
	Time Received Left-AOO < 500 milliseconds	T

= Unknown

Control Mode	Operational	T
Inputs	Time Received Left-AOO < 500 milliseconds	F

Flight[I]-Detect

STATE

- **Description:** The Detect state variable specifies whether all conflicts computations must be carried out for the flight.
- **Obsolesence:** It becomes obsolete when the flight leaves the area of operations or if a conflict has been detected and the recomputed timer has not timed out.
- **Exception Handling:** Not defined.
- **Definition:**

= Compute

Control Mode	Operational	T	T	T	T	T
Inputs	Received New-Flight	T	-	-	-	-
	New-Flight.Flight-Id = Flight[I].Flight-Id	T	-	-	-	-
	Flight[I].Flight-Id = Obsolete	F	F	F	F	-
	Received Changed-Flight	-	T	-	-	-
	Changed-Flight.Flight-Id = Flight[I].Flight-Id	-	T	-	-	-
	Received Current-Position	-	-	T	-	-
	Current-Position.Flight-Id = Flight[I].Flight-Id	-	-	T	-	-
	Received Reinclude-Flight	-	-	-	T	-
	Reinclude-Flight.Flight-Id = Flight[I].Flight-Id	-	-	-	T	-
	Time-Last-Computed > RECOMPUTE-TIME$_{TBD}$	-	-	-	-	T

= Unknown

Control Mode	Operational	T	T	-
Inputs	Received Exclude-Flight	T	-	-
	Exclude-Flight.Flight-Id = Flight[I].Flight-Id	T	-	-
	Flight[I].Flight-Id = Obsolete	F	T	-
	Time-Last-Computed > RECOMPUTE-TIME$_{TBD}$	-	-	F

Trajectory[I]-Data

STATE

- **Description:** The Data state variable specifies whether the data associated with the trajectory of a flight is new or recalculated.
- **Obsolesence:** It becomes obsolete when the flight leaves the area of operations or when the trajectory is deleted.
- **Exception Handling:** Not defined.
- **Definition:**

= New

Control Mode	Operational	T
Inputs	Received New-Trajectory	T
	New-Trajectory.Trajectory-Id = Trajectory[I].Trajectory-Id	T
	Trajectory[I].Trajectory-Id = Obsolete	F

= Recalculated

Control Mode	Operational	T
Inputs	Received Recalculated-Trajectory	T
	Recalculated-Trajectory.Trajectory-Id = Trajectory[I].Trajectory-Id	T
	Trajectory[I].Trajectory-Id = Obsolete	F

= Unknown

Control Mode	Operational	T
Inputs	Trajectory[I].Trajectory-Id = Obsolete	T

Trajectory[I]-Type

STATE

- **Description:** The Type state variable specifies whether the trajectory is a system trajectory or a tentative trajectory.
- **Obsolesence:** It becomes obsolete when the flight leaves the area of operations or when the trajectory is deleted.
- **Exception Handling:** Not defined.
- **Definition:**

= System

Control Mode	Operational	T	T
Inputs	Received New-Trajectory	T	-
	New-Trajectory.Tentative = System	T	-
	Trajectory[I].Trajectory-Id = Obsolete	F	F
	Received Recalculated-Trajectory	-	T
	Recalculated-Trajectory.Tentative = System	-	T

= Tentative

Control Mode	Operational	T	T
Inputs	Received New-Trajectory	T	-
	New-Trajectory.Tentative = Tentative	T	-
	Trajectory[I].Trajectory-Id = Obsolete	F	F
	Received Recalculated-Trajectory	-	T
	Recalculated-Trajectory.Tentative = Tentative	-	T

= Unknown

Control Mode	Operational	T
Inputs	Trajectory[I].Trajectory-Id = Obsolete	T

Trajectory[I]-Status

STATE

- **Description:** The Status state variable specifies whether the trajectory is defined, deleted or unknown.
- **Obsolesence:** It becomes obsolete when the flight leaves the area of operations or when the trajectory is deleted.
- **Exception Handling:** Not defined.
- **Definition:**

= Defined

Control Mode	Operational	T	T
Inputs	Trajectory[I].Data = New	T	-
	Trajectory[I].Data = Recalculated	-	T

= Deleted

Control Mode	Operational	T
Inputs	Received DELETE-TRAJECTORY	T
	DELETE-TRAJECTORY = Trajectory.Trajectory-Id	T
	Trajectory[I].Trajectory-Id = Obsolete	F

= Unknown

Control Mode	Operational	T
Inputs	Trajectory[I].Trajectory-Id = Obsolete	T

Airspace[I]-Status

STATE

- **Description:** The Status state variable specifies whether the airspace is defined, excluded or unknown.
- **Obsolesence:** It becomes obsolete when MTCD is powered down or when a delete airspace message is received.
- **Exception Handling:** Not defined.
- **Definition:**

= Defined

Control Mode	Operational	T
Inputs	Received Area-of-Operations	T
	Airspace[I].Airspace-Id = Obsolete	F

= Excluded

Control Mode	Operational	T
Inputs	Received Exclude-Airspace	T
	Exclude-Airspace.Airspace-Id = Airspace[I].Airspace-Id	T
	Airspace[I].Airspace-Id = Obsolete	F

= Unknown

Inputs	Airspace[I].Airspace-Id = Obsolete	F

Airspace[I]-Special-Use-Airspace

STATE

- **Description:** The Special-Use-Airspace state variable specifies whether there are any special use airspace criteria associated with the airspace.
- **Obsolesence:** It becomes obsolete when MTCD is powered down or when a delete airspace message is received.
- **Exception Handling:** Not defined.
- **Definition:**

= Defined

Control Mode	Operational	T
Inputs	Received Special-Use-Airspace	T
	Special-Use-Airspace.Airspace-Id = Airspace[I].Airspace-Id	T

= Unknown

Control Mode	Operational	T	T
Inputs	Airspace[I].State = Unknown	T	F
	Received Special-Use-Airspace	-	F
	Previous Special-Use-Airspace = Unknown	-	T

Airspace[I]-Uncertainty-Airspace

STATE

- **Description:** The Uncertainty-Airspace state variable specifies whether there are any uncertainty criteria associated with the airspace.
- **Obsolesence:** It becomes obsolete when MTCD is powered down or when a delete airspace message is received.
- **Exception Handling:** Not defined.
- **Definition:**

= Defined

Control Mode	Operational	T
Inputs	Received Uncertainty-Airspace	T
	Uncertainty-Airspace.Airspace-Id = Airspace[I].Airspace-Id	T

= Unknown

Control Mode	Operational	T	T
Inputs	Airspace[I].State = Unknown	T	F
	Received Uncertainty-Airspace	-	F
	Previous Uncertainty-Airspace = Unknown	-	T

Airspace[I]-Separation-Criteria

STATE

- **Description:** The Separation-Criteria state variable specifies whether there are any separation criteria associated with the airspace.
- **Obsolesence:** It becomes obsolete when MTCD is powered down or when a delete airspace message is received.
- **Exception Handling:** Not defined.
- **Definition:**

= Defined

Control Mode	Operational	T
Inputs	Received Separation-Criteria	T
	Separation-Criteria.Airspace-Id = Airspace[I].Airspace-Id	T

= Unknown

Control Mode	Operational	T	T
Inputs	Airspace[I].State = Unknown	T	F
	Received Separation-Criteria	-	F
	Previous Separation-Criteria = Unknown	-	T

Airspace[I]-Lowest-Flight-Level

STATE

- **Description:** The Lowest-Flight-Level state variable specifies whether there is a lowest flight level criteria associated with the airspace.
- **Obsolesence:** It becomes obsolete when MTCD is powered down or when a delete airspace message is received.
- **Exception Handling:** Not defined.
- **Definition:**

= Defined

Control Mode	Operational	T
Inputs	Received Lowest-Flight-Level	T
	Lowest-Flight-Level.Airspace-Id = Airspace[I].Airspace-Id	T

= Unknown

Control Mode	Operational	T	T
Inputs	Airspace[I].State = Unknown	T	F
	Received Lowest-Flight-Level	-	F
	Previous Lowest-Flight-Level = Unknown	-	T

AOO

STATE

- **Description:** The AOO state variable specifies whether the Area of Operations for MTCD has been specified.
- **Obsolesence:** It becomes obsolete when MTCD is powered down.
- **Exception Handling:** Not defined.
- **Definition:**

= Defined

Control Mode	Startup	T
Inputs	Received Area-of-Operations	T
	Previous AOO = Unknown	T

= Unknown

Inputs	AOO.Aoo-Airspace = Obsolete	T

Class-of-Flight

STATE

- **Description:** The Class-of-Flight state variable specifies whether the separation criteria for various classes of flight for MTCD have been specified.
- **Obsolesence:** It becomes obsolete when MTCD is powered down or when a delete airspace message is received.
- **Exception Handling:** Not defined.
- **Definition:**

= Defined

Control Mode	Startup	T
Inputs	Received Class-of-Flight	T
	Previous Class-of-Flight = Unknown	T

= Unknown

Inputs	COF.Class-of-Flight = Obsolete	T

Phase-of-Flight

STATE

- **Description:** The Phase-of-Flight state variable specifies whether the uncertainty criteria and separation criteria for various phases of flight for MTCD have been specified.
- **Obsolesence:** It becomes obsolete when MTCD is powered down or when a delete airspace message is received.
- **Exception Handling:** Not defined.
- **Definition:**

= Defined

Control Mode	Startup	T
Inputs	Received Phase-of-Flight	T
	Previous Phase-of-Flight = Unknown	T

= Unknown

Inputs	POF.Phase = Obsolete	T

Navigational-Capabilities

STATE

- **Description:** The Navigational-Capabilities state variable specifies whether the uncertainty criteria for various flight for MTCD have been specified. The criteria are assigned based on the equipment on board the aircraft.
- **Obsolesence:** It becomes obsolete when MTCD is powered down or when a delete airspace message is received.
- **Exception Handling:** Not defined.
- **Definition:**

= Defined

Control Mode	Startup	T
Inputs	Received Navigational-Capabilities	T
	Previous Navigational-Capabilities = Unknown	T

= Unknown

Inputs	NAV.Parameters = Obsolete	T

Parallel-ATS-Routes

STATE

- **Description:** The Parallel-ATS-Routes state variable specifies whether parallel ATS routes are present and the associated separation criteria.
- **Obsolesence:** It becomes obsolete when MTCD is powered down or when a delete airspace message is received.
- **Exception Handling:** Not defined.
- **Definition:**

= Defined

Control Mode	Startup	T
Inputs	Received Parallel-ATS-Routes	T
	Previous Parallel-ATS-Routes = Unknown	T

= Unknown

Inputs	PAR.Routes = Obsolete	T

3.11 Functions

Functions may be specified separately and referenced in the tables rather than including complex mathematical functions directly in the transition tables.

Airspace-Violated

FUNCTION

- **Description:** The function is called to determine the airspace in which the trajectory violated the lowest level criteria.
- **References:**
- **Appears in::** Descent-Below-Lowest-Level
- **Definition:** The function accepts the trajectory index and violation type and determines the airspace in which the violation takes place.

First-Infringement

FUNCTION

- **Description:** The function is called to determine the time at which the separation between trajectories or trajectories and airspace is first violated.
- **References:**
- **Appears in::** Descent-Below-Lowest-Level
- **Definition:** The function accepts the trajectories and type of conflict or the trajectory, airspace and type of conflict, to return the time at which the infringement occurs.

Possible-Position

FUNCTION

- **Description:** The function is called to determine the position of the aircraft in a trajectory at a given time, taking into consideration uncertainty information.
- **References:**
- **Appears in::** Descent-Below-Lowest-Level
- **Definition:** The function accepts the trajectory, time and airspace in which the position is needed and returns the position. It uses the uncertainty information from airspace and class-of-flight to determine the position.

Nominal-Position

FUNCTION

- **Description:** The function is called to determine the position of the aircraft in a trajectory at a given time, without taking into consideration uncertainty information.
- **References:**
- **Appears in::** Descent-Below-Lowest-Level
- **Definition:** The function accepts the trajectory, time and airspace in which the position is needed and returns the position. It uses only the nominal route informationt to compute the position.

LFLV

FUNCTION

- **Description:** The function is called to determine if a given trajectory violates the lowest flight level specified.
- **References:**
- **Appears in::** Descent-Below-Lowest-Level
- **Definition:** The function accepts the trajectory and uses the configuration parameters of MTCD to determine the lookup horizon. Returns ture if there is a violation within the lookup horizon. Uses the Airspace-Violated and First-Infringement functions to determine if there is a violation and whether it lies within the lookup horizon.

Determine-Conflict-Id

FUNCTION

- **Description:** The function is called to generate a unique conflict id.
- **References:**
- **Appears in::** Descent-Below-Lowest-Level, Nominal-Routes-Overlap, Aircraft-Conflict, Special-Use-Airspace.
- **Definition:** The function accepts the flight and the trajectory id of both aircraft involved in a conflict and determines the conflict id. For a Descent Below Lowest Level violation or a Special Use Airspace violation, the second flight and trajectory id are specified as zero.

Min-Nominal-Distance

FUNCTION

- **Description:** The function is called to determine the time at which the nominal distance between the airspace and the trajectory or the distance between two trajectories is the smallest.
- **References:**
- **Appears in::** Nominal-Routes-Overlap, Aircraft-Conflict, Special-Use-Airspace.
- **Definition:** The function accepts the trajectories and type of conflict or the trajectory, the airspace and type of conflict and determines the time at which the nominal distance is minimum.

3.12 Macros

Macros are simply named pieces of AND/OR tables that can be referenced from within another table. Its use is not necessary but simplifies the specification and makes it easier to understand. For complex models, macros are almost required for humans to be able to handle the complexity involved in constructing the specification.

MTCD-Configured

MACRO

- **Description:** The macro is invoked to check if MTCD is configured.
- **Comments:**
- **References:**
- **Appears in::** Control-Modes
- **Definition:**

State	AOO = Defined	T
	Phase-Of-Flight = Defined	T
	Class-Of-Flight = Defined	T
	Flight-DefinedM	T

Flight-Defined

MACRO

- **Description:** The macro is invoked to check if there is at least one flight defined for the system.
- **Comments:**
- **References:**
- **Appears in::** Control-Modes
- **Definition:**

There exists I such that

State	Flight[I].Flight-Data = Defined	T

Trajectory-Defined

MACRO

- **Description:** The macro is invoked to check if there is at least one trajectory available for conflict detection computations.
- **Comments:**
- **References:**
- **Appears in::** Control-Modes
- **Definition:**

There exists I such that

State	Trajectory[I].Data = Defined	T

Chapter 4

Conclusions

4.1 Achievements

In this thesis, we gave a demonstration of a safety-centered methodology to developing new air traffic management tools. We based our demonstration on a Medium Term Conflict Detection new automated tool developed by Eurocontrol for the Air Traffic Control systems in Europe.

The methodology we used was supported by a new specification structuring approach, developed by Prof Leveson and her students in the Software Engineering Research Laboratory at the Massachussets Institute of Techonology.
It is called Intent Specifications and supports traceability and documentation of design rationale as the development process proceeds. In this thesis we focused on the first three levels of this approach.

The **first level** gave the historical context, the system functional goals, and the description of the environment, and of the assumptions and constraints of this environment on the system. It also contained the results of a preliminary hazard and task analyses that conducted to the system requirements, constraints and limitations.

The **second level** of the specification contained the basic system scientific and engineering design principles needed to achieve the behavior specified in the first level. It answered the question "why" for the design decisions and describes any basic principles or assumptions upon which the system design depends.

We gave the general definitions we used for the different types of conflicts to be detected by MTCD and we described in more details the system design principles that we have linked to the system requirements. We also highlighted the tradeoffs and the rationale for the design decisions that have been made for MTCD.

The **third level** described the blackbox behavior of the system components, including humans, the logical aspects of the interfaces between the components and any assumed relevant environment behavior.

We used a specification language called SpecTRM-RL (Specification Tools and Requirements Methodology-Requirements Language) for modeling the blackbox software behavior using our system requirements.

This level had to satisfy both objectives: to be easily readable to serve as the official system specification of the behavioral requirements and to have an underlying formal model that can be executed and subjected to mathematical analysis.

The SpecTRM-RL specification was composed of four main parts: a specification of the supervisory modes, a specification of the operating modes, a model of the controlled process that includes the inferred system states, and a specification of the inputs and outputs.

Finally, the Medium Term Conflict Detection automated tool developed by Eurocontrol was found to be a very good example for our demonstration because it is highly interactive with other software components and with a human controller and also because of the actual urge in the industry to find automated solutions to the current tremendous increase of traffic implying important congestion and delays in the air travel industry.
It thus highlights all the challenges faced today in the industry to build such complex, safe and reliable systems.

4.2 Further work

For our safety-centered approach to developing new Air Traffic Management tools to be complete, additional work should be done:

- To integrate the human-centered approach now under development in the Software Engineering Research Laboratory at M.I.T. by Mirna [8] with for example operator task, mode confusion, human error, and usability analyses.

- And to actually use our executable SpecTRM-RL model we built for MTCD to lead for example the following system hazard analyses described in Leveson's previous work on automated ATC systems [28]: completeness, consistency, state machine hazard, and deviation analyses.

Finally this work could also be completed with the level 4 and 5 of the Intent Specifications approach described in [23] and already applied on another Air Traffic Control project in [28].

The fourth level of an Intent Specification would contain the normal physical design representation with links to the levels above.

The fifth level would contain the actual software, the hardware assembly, the installation instructions, and the training requirements.

Acronyms

Acronym	Definition
ACAS	Airborne Collision Avoidance System
AOI	Area of Interest
AOO	Area of Operation
AMAN	Arrival Manager
ASAS	Airborne Separation Assurance
ATC	Air Traffic Control
ATM	Air Traffic Management
ATS	Air Traffic Service
B757	Boeing 757
CFMU	Control Flow Management Unit
COF	Class of Flight
CTAS	Center-TRACON Automation System
DMAN	Departure Manager
ECAC	European Civil Aviation Community
EDPS	Environment Data Processing System
FAA	Federal Aviation Administration
FDPS	Flight Data Processing System
FL	Flight Level
HMI	Human Machine Interface
ICAO	International Civil Aviation Organization
MTCD	Medium Term Conflict Detection
MONA	Monitoring Aids
NM	Nautical Miles
PC	Planning Controller
PHA	Preliminary Hazard Analysis
POF	Phase of Flight
PTA	Preliminary Task Analysis
RPVD	Radar Plan View Display
SIL	Sectory Inbound List
SpecTRM-RL	Specification Tools and Requirements Methodology-Requirements Language
TBC	To Be Confirmed
TBD	To Be Defined
TC	Tactical Controller
TRACON	Terminal Radar Approach CONtrol

Bibliography

[1] N. Leveson web page: http://sunnyday.mit.edu, 2001.

[2] Eurocontrol Experimental Center web page: http://www.eurocontrol.fr/, 2001.

[3] ATM FAA/Eurocontrol Seminar 2001 web page: http://www.atm2001.eurocontrol.fr/.

[4] C.E. Billings. *Aviation Automation: The Search for Human-Centered Approach.* Lawrence Erlbaum Associates, 1997.

[5] Eurocontrol Experimental Center. Validation of a Methodology for Predicting Performance and Workload, june 1999. EEC note No 7/99.

[6] Eurocontrol Experimental Center. Expression of Requirements for HMI Specifications for ATC/CWP, March 2001. EEC Note No. 06/01, Project HRS/HSP-006 (Core Requirements for ATM Working Positions).

[7] D. Javaux, Work Psychology Department, University of Liège, Belgium. The Prediction of Pilot-Mode Interaction Difficulties, Spreading Activation Networks as An Explanation of Frequential and Inferential Simplifications.

[8] M. Daouk. A Human-Centered Approach to Developing New Air Traffic Management Tools. Master's thesis, Aeronautics and Astronautics Department, Massachussets Institute of Technology. In Preparation for Fall 2001.

[9] Eurocontrol. Functional Specification for EATCHIP Phase III - Medium Term Conflict Detection, Sept 1997. Draft DPS.ET1.ST06.DEL01.02.5.

[10] Eurocontrol. Operational Requirements Document for EATCHIP Phase III - ATM Added Functions - Volume 5 - Medium Term Conflict Detection, January 1999. European Air Traffic Management Programme, OPR.ET1.ST04.DEL01.5, Edition 2.0.

[11] Eurocontrol. Operational Requirements Document for EATCHIP Phase III - ATM Added Functions - Volume 1 - Monitoring Aids, January 1999. European Air Traffic Management Programme, OPR.ET1.ST04.DEL01.1, Edition 2.0.

[12] Eurocontrol. Operational Requirements Document for EATCHIP Phase III - ATM Added Functions - Volume 2 - Safety Nets, January 1999. European Air Traffic Management Programme, OPR.ET1.ST04.DEL01.2, Edition 2.0.

[13] Eurocontrol. Operational Requirements Document for EATCHIP Phase III - ATM Added Functions - Volume 3 - Arrival Manager, January 1999. European Air Traffic Management Programme, OPR.ET1.ST04.DEL01.3, Edition 2.0.

[14] Eurocontrol. Control Flow Management Unit Operations, Executive Summary, 2000.

[15] Eurocontrol. Eurocontrol Safety Regulatory Requirement 3 (ESARR): Use of Safety Management Systems by ATM Service Providers, July 2000.

[16] Eurocontrol. Eurocontrol Safety Regulatory Requirement 4 (ESARR): Risk Assessment and Mitigation in ATM , July 2000.

[17] Eurocontrol. MTCD Algorithmic Overview, June 2000. EATCHIP Escape MTCD Algorithms.

[18] G. Glynn Eurocontrol Experimental Center. Romania 99 Real-Time Simulation Controller Information, May 1999. EEC-RTO.

[19] G. Glynn Eurocontrol Experimental Center, M. Bonnier. EATCHIP III Evaluation and Demonstration, Phase III Project, Experiment 3A Bis: MTCD, Final Report, October 2000.

[20] G. Glynn Eurocontrol Experimental Center, M. Bonnier. Romania 99 Real-Time Simulation System Handbook, March 2001. 5.0.EEC-OPS.

[21] M. Heimdahl. Intent Specification for the Altitude Switch. Department of Computer Science and Engineering, University of Minnesota.

[22] Nancy G. Leveson. *Safeware: System Safety and Computers.* Addison-Wesley, 1995.

[23] Nancy G. Leveson. Draft Intent Specifications (including SpecTRM-RL) user manual. M.I.T., Safeware Engineering Corporation, August 1999.

[24] Nancy G. Leveson. Evaluating Accident Models using Recent Aerospace Accidents. March 2001. Software Engineering Research Laboratory, Aeronautics and Astronautics Dept. M.I.T.

[25] M. de Villepin, M. Katahira, M. Rodriguez, M. Zimmerman, B. Ingram, N. Leveson, Dept of Aeronautics and Astronautics, M.I.T. Identifying Mode Confusion Potential in Software Design.

[26] M. Heimdahl, Department of Computer Science, Michigan State University; N. Leveson, Computer Science and Engineering, University of Washington. Completeness and Consistency in Hierarchical State-Based Requirements.

[27] Computer Science N. Leveson and NASA Ames Research Center Engineering; E. Palmer. Designing Automation to Reduce Operator Errors. Oct 1997.

[28] N. Leveson, L. Alfaro, C. Alvarado, M. Brown, E.B. Hunt, M. Jaffe, S. Joslyn, D. Pinnel, J. Reese, J. Samarziya, S. Sandys, A. Shaw, Z. Zabinsky. A Demonstration Safety Analysis of Air Traffic Control Software. 1997. University of Washington, Seattle Wa.

[29] N. Leveson, M. de Villepin, M. Daouk, J. Bellingham, J. Srinivasan, N. Neogi, E. Bachelder (M.I.T); N. Pilon, G. Flynn (Eurocontrol Experimental Center). A Safety and Human-Centered Approach to Developing New Air Traffic Management Tools (Extended Abstract). *ATM 2001 seminar*, 2001.

[30] Jon Damon Reese Nancy Leveson, Mats P.E Heimdahl. Designing Specification Languages for Process Control Systems: Lessons Learned and Steps to the Future. 2001. M.I.T. Aeronautics and Astronautics Department, University of Minnesota Computer Science and Engineering Department.

[31] Jens Rasmussen. Risk Management in a Dynamic Society: a Modelling Problem. *Safety Science Vol.27 No 2/3, pp 183-213*, 1997. Hurecon, Smorun, Denmark.

[32] Alex Vink. EATCHIP Medium Term Conflict Detection. June 1997. Part 1 - EATCHIP Context - ATM Seminar 1997.

[33] Alex Vink. What is MTCD? 1997. http://atm-seminar-97.eurocontrol.fr/vink.htm.