Dependability analyses and assessment of ARTAS: ATM SURVEILLANCE TRACKER AND SERVER

Sylvain METGE

SRTI SYSTEM
105, avenue du Général Eisenhower
BP 1186, 31037 Toulouse Cedex 1 - FRANCE

Alain PEYTAVIN

CENA
7, avenue Edouard Belin
BP 4005, 31055 Toulouse Cedex - FRANCE

Abstract

ARTAS is a Surveillance Data Processing System designed to provide a single, 'seamless' Air Situation Picture throughout ECAC. This paper describes the rigorous, but pragmatic, dependability approach which provides End Users of ARTAS with evidence and supporting arguments such that they can rely on the system. This paper presents relevant safety analyses and concrete result. In the conclusion, it is explained that the dependability activity may form part of a wider « safety case[1] » by addressing significant safety issue. A follow-up of the dependability approach on future ARTAS development is mentioned as well.

Foreword

This paper essentially deals with a dependability activity achieved on ARTAS, a European Surveillance Data Processing System [1], in order for the most part, to verify whether initial dependability objectives are met. Besides the relevance of the applied dependability process to verify such requirements [2], advantage of such an approach can also be taken to make sure that, even in a degraded mode of operation, radar data accuracy remain quite acceptable with regard to a specified separation minima. Indeed, from the french CAA point of view, the intrinsic accuracy offered by the radar data processing systems currently in use, may not be in accordance, as it is, with reduced separation minima (*e.g. 5NM*). This subject is briefly discussed in the last part of this paper.

Introduction

Modern Air Traffic Management relies on timely and accurate surveillance data. Loss or degradation of such data in one part of the airspace can increase risk in the short-term and result in a temporary reduction in airspace capacity.

Nowadays, many different types of Radar Data Processing (*RDP*) Systems are implemented in ECAC. They apply different radar data processing techniques with a large variation in the process sophistication and level of performance. These systems often operate autonomously, in that they do not liaise in any way with neighboring systems. One consequence of the current incompatibility of these systems is that it restricts the number of aircraft that can be transferred from one ATC system to the next one. This is partly due to the increased separation minima which have to be applied at transfers.

The introduction of a seamless system, in which, the same radar minima as for intra-center operation could be applied at transfers, would contribute to an overall increase in airspace capacity. This is why, as part of EATCHIP[2], the development of a Europe

[1] The « safety case » is a global safety approach whose objective is to present the evidence, arguments and assumptions to show that system risks have been identified and controlled throughout the life of the system.

[2] European Air Traffic Control Harmonization and Integration Programme

wide Surveillance Data Processing and Distribution System, named ARTAS[3] [3] was undertaken by EUROCONTROL in cooperation with National ATM administrations and industrial partners. The goals of ARTAS are to overcome all present RDPS shortcomings and to provide a basis for a harmonized and integrated SDPD system in ECAC.

The ARTAS system must be highly dependable (*i.e. available, reliable and risk-free*) and provide users with high integrity surveillance data. In order to verify whether ARTAS has achieved this global objective, a detailed dependability study has been performed by CENA[4] under contract from EUROCONTROL. The dependability study was performed on ARTAS system which became operational June 98. Given that when the study started, this version of ARTAS was in an advanced state of design, the main purpose of the study was to confirm that this system had reached a satisfactory level of safety, compliant with initial requirements.

Presentation of ARTAS

ARTAS Environment

The ARTAS system is a distributed system consisting of a number of identical co-operating ARTAS units. Each ARTAS unit processes surveillance data, exchanging and co-ordinating its own track information with adjacent ARTAS units in order to build a unique, coherent and continuous Air Situation Picture for the complete airspace. Figure 1 presents an example of the ARTAS 1 environment.

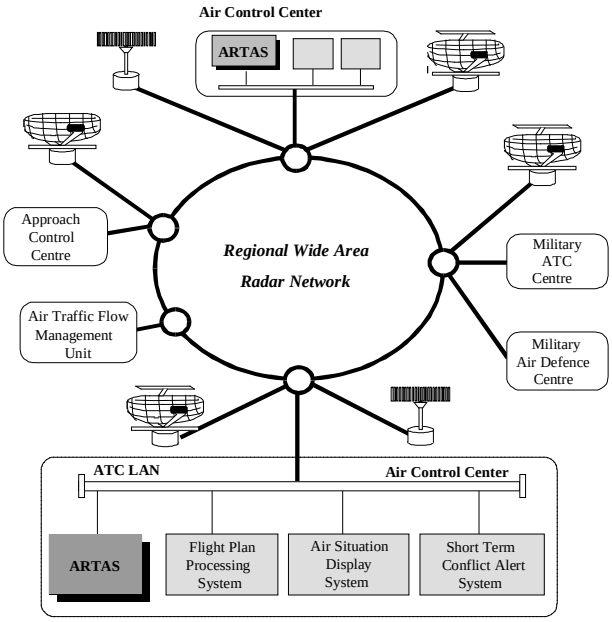

Fig. 1 Example of ARTAS 1 environment.

Two categories of systems are part of the ARTAS operational environment, these are:

- Data sources (*e.g. radars, including Primary, Secondary Surveillance radars or combined*) which provide ARTAS with the information necessary to elaborate the Air situation Picture;
- Users which are any ATC subsystems having a requirement to receive, at defined instants, the best and most up-to-date estimate of all or selected aircraft state vector elements for all air traffic of interest to these users.

Some systems (*e.g. Flight Plan Processing System*) are both Data Source and Data User to ARTAS.

Functional architecture

An ARTAS unit can be divided into three main functional components which are:

- The **Tracker** function which processes the radar input data and maintains the latest air situation;
- the **Server** function which performs the track information Service (*i.e. distributing track information to the Users*) and ensures the inter-ARTAS cooperation;
- the **System Manager** function which carries out the functions related to the supervision and management of the ARTAS unit.

---

[3] ARTAS is the acronym of ATM SURVEILLANCE TRACKER AND SERVER

[4] CENA (*Centre d'Etudes de la Navigation Aérienne*) belongs to the french Civil Aviation Authorithy. CENA carries out research and studies on ATM and related topics.

The information shared between the functional elements are stored within the ARTAS data bases:

- the Track database which contains data involved in the track information service;
- the Radar database which gathers the characteristics of the radars and associated relevant information;
- the Geographical database which contains information related to the airspace organization and map information needed by the Tracker and the Server.

Physical architecture

The hardware platform (*ARTAS unit*) has a fully duplicated, hot stand-by architecture, in order to meet high level of availability (*the unavailability shall not exceed 5 minutes a year*). An ARTAS unit is composed of interconnected equipment communicating via an internal FDDI network.
Hardware elements (*called **nodes***) act as master and slave for redundancy and fallback purpose. The switch-over is automatically triggered, element by element, independently, the twin node being replaced by its dual hot stand-by node.
Hardware elements are:

- the Tracker Station which provides the computation capacity for the Tracker function;
- the Server Station which provides the computation capacity for the Server function;
- the System Manager Station which provides the computation capacity and the peripherals required by the function related to the Supervision and the management of the hardware components of an ARTAS unit;
- the Router Bridge Station which handles the external interfaces of the ARTAS unit with the Local Area Network for transmission / reception of all information exchanged with radars, FPPS, and Users and the master timing source;
- the Data Analysis Stations which provides the computation capacity and the peripherals required by, in particular, the recording function.

Figure 2 depicts the redundant architecture of an ARTAS unit, where the shaded rectangles represent the « slave » HW elements.

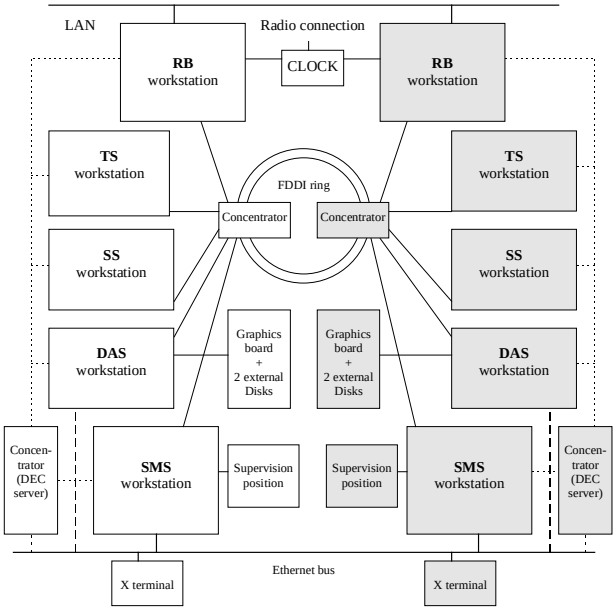

Fig. 2 ARTAS unit HW architecture.

External Interfaces

ARTAS operates in a communication environment comprising both Wide Area Networks (WANs) for inter-center communications and LANs. Inter-center communications are normally carried out over Regional Radar WANs (*e.g. RADNET in Germany/Benelux and RENAR in France*). For regions where such networks are not available, provision has been made to handle communications on a locally installed network. The connection of radar using dedicated X.25 lines is then done through hardware equipment.

ARTAS system availability requirements

ARTAS is a system designed to operate 24 hours per day, seven days per week. Should occasion arise to re-introduce a component of an ARTAS Unit into operation (*e.g. software upgrade*) then the ARTAS Unit is capable of managing and completing the ARTAS start-up process within 15 minutes.
Any SW/HW failures leading to a reconfiguration process will be detected within 1 second. The subsequent switchover to the hot-standby processing equipment is transparent to ARTAS Users (*i.e. track numbers do not change*) and takes no longer than 3 seconds. All these requirements have been tested according to inspection, analysis or demonstration methods, depending upon relevant appropriateness.

ARTAS dependability activity

Dependability can be viewed as the property of a system such that reliance can justifiably be placed on the **capabilities** it provided to its **users**.

A capability is the result of activities carried out by a number of system functions, each function contributing to provide the capabilities through the outputs they produce. Unlike capabilities, functions are internal to the system; the user is not necessarily aware of them.

ARTAS users are any ATC subsystems such as Flow Management Control Systems, remote TMA's, Military units, STCA functions, Controller Display System, FPPS, etc. Note that the role played by the ARTAS Operator, in charge of supervising ARTAS system and acting on behalf on « broadcast users[5] », was considered in the dependability study.

Methodology

Three main dependability activities were performed:

1. technical and functional analysis which provided some preliminary results necessary for achieving the next two activities efficiently;
2. qualitative-oriented risk analyses for both hardware and software architectures;
3. « Feared events » assessment using the Fault-tree technique. Feared events are highly undesired operational situations due to external failures or system outages.

These activities are further detailed in this paper.

Since the rigid DoD-STD-2167A[6] life-cycle model was applied during the development of ARTAS, the dependability study has made the most of the deliverables products. In particular, documentation (*e.g. the System/SegmentSpecification, Interface Requirements Specification, Software Requirements Specification and System/Segment Design Document*) have been especially used as inputs for carrying out the dependability activities.

Technical and functional risk analysis

The first steps of the dependability study consisted of identifying the different categories of users, and identifying a set of twenty capabilities provided by ARTAS to those users. Capabilities were then assigned a level of importance according to the impact of the loss or degradation of that capability based on different categories of user. This permitted further assessment of the impact of failures towards the users, taking into account their specific expectations or requirements. Three levels of importance were defined (*routine, important, critical*). Additional criteria were considered such as the interactive characteristics (on-line / off-line mode of operation) as well as the time-limit tolerated by the user (*short-term, medium-term, long-term*). Figure 3 gives an example of ARTAS capability with associated characteristics.

In addition to these tasks, the most important work of activity #1 was a functional risk analysis. The aim of such an activity was to clearly identify the interactions between system functions and the data exchanged between the functions. ARTAS functional analysis was performed using a SADT[7].

This analysis was also used to map capabilities and functions and then link functions to software components, called CSC (*in accordance with the DOD-STD-2167A terminology*), in order to further perform the risk analysis on ARTAS software architecture.

For information, ARTAS software which is constituted of 85 CSCs, has been split-up into 65 functions. The overall volume of the application software is approximately 410,000 lines of Ada and 265,000 lines of C++.

---

[5] For this category of users, no direct dialogue is set-up with ARTAS. Common standard services are broadcast over the network, whether users are effectively connected or not.

[6] Military Standard approved for use by US Dpt of Defense, Washington, D.C.

[7] SADT is the acronym of Structured Analysis and Design Technique

Ident. C2	**Name**: To enable service update					
Description						
This capability permits any User to modify certain parameters of a service request during the period of time the service is provided. This capability is used to refine the Domain of Interest resulting from the track selector(s) or the Item Selection. The user can also request specific tracks.						
End User(s)	**Level of importance**					
ARTAS point-to-point users	Routine	☐	Important	☐	Critical	☒
Operator (broadcast users)	Routine	☐	Important	☒	Critical	☐
Tolerated time-limit	short-term	☐	medium-term	☒	long-term	☐

Fig. 3 Example of capability (C2) and associated characteristics

Risk analyses

FMEA technique

Failure to meet the dependability requirements may originate from a number of sources including failures of external systems, internal hardware or software components.
Therefore the Failure Modes Effects Analysis (*FMEA*) technique was used to conduct the risk analyses both on the software and hardware architectures. Note that a risk analysis when applied to the software is slightly different and is generally known as *Software Error Effect Analysis*. A large literature base details the principle of these techniques; refer, for example, to the US Aerospace Recommended Practice from the Society of Automotive Engineers (*SAE*), the US DoD MIL-STD-785, or the IEC standard on Analysis techniques for system reliability. The use of tabulated forms have facilitated these analyses.

The main objectives of the FMEA performed on ARTAS were:

- to assess the effects of HW and/or SW failures on an ARTAS unit, and consequently, impacts on the Users;
- to clearly indicate whether existing detection mechanisms and associated fallback actions were implemented or not;
- to set the appropriate severity level;
- according to the severities, to propose additional mitigation means, when necessary.

Scope of the system

In order to propose adequate mitigation means for each failure mode, it was necessary to clearly agree on the systems that should be considered as « external systems » to an ARTAS unit (*e.g. FPPS or radar stations*) and subsystems that are elements of the ARTAS unit. A useful result of this task was the identification of those internal ARTAS functions requiring external inputs and which were therefore dependent on the external sub-systems for correct functioning.

About severity levels

Failure modes were ranked into severity levels. Definitions can be found in standards such as JAR-25-1309, NAS-SR-138 from the US F.A.A. but they are airborne oriented. Regarding to severity levels as proposed in EATCHIP Safety Assessment Methodology [4], they are strongly Air Traffic Controller Officers oriented. ARTAS severity level definitions were adapted taking into account the specificity of ARTAS system and the category of its direct users that are not necessarily ATCOs.

Three severity levels were defined in accordance with ARTAS context:

- **minor** (m): failure which leads to the loss or degradation of a routine capability. Nominal mode shall restored with no impact on important or critical capabilities;
- **Major** (M): failure that leads to the degradation of an important capability, where the duration of the failure does not exceed user defined limits (*specified as either absolute time - e.g. 15 minutes per occurrence - or cumulated time - e.g. 15 minutes per year*);
- **Catastrophic** (C): failure resulting in the degradation of a critical capability or, the loss of either a critical or important capability.

ARTAS failure detection mechanisms

The ARTAS middleware (UBSS[8] - *Unix Based System Software*) performs process management and failure detection / recovery mechanisms (*switch-overs, start-overs*) following SW or HW failures. UBSS has been designed according to DOD 2167A US standard with respect to ISO 9001 and SPICE quality process. Dedicated UBSS processes and libraries are in charge of handling failure detection and abnormal situations such as an isolate node in dual configuration. These processes use, for example, polling and watchdog mechanisms in order to ensure that a node is still alive. Some HW failures are indirectly detected by their impact on a particular software process. As an example, « I/O exit codes » are generated and captured by UBSS when a disk is unavailable (*due to e.g .a defective disk controller*). In most cases failures will lead to an automatic switch-over.

FMEA on hardware architecture

ARTAS HWCIs were decomposed into smaller components such as CPU, internal disk or LAN controllers. The following generic failure modes have been considered:

- loss / stop of the HW component due to, for example, defective power supply or inadvertent resets;
- erroneous outputs due to defective memory (*corruption of disk track, memory board outage,...*);
- delayed outputs, originating in a software fault such as a failed queue management.

As a result of this FMEA, a summary document was produced identifying the most a set of critical areas that needed to be amended. Concrete examples are :

- the FDDI network initial architecture, for which the hub did not enable a « Double Attachement Station » configuration requiring a double optic fiber ring, owing to lack of available slots;
- the Ethernet LAN controller configuration, for which a dupliqued LAN would have improved the reliability of the communication exchanges between the Router Bridge and the radar stations;
- the clock synchronization drifts. They may have resulted in an ARTAS unit potentially rejecting some radar data as well as tracks from Adjacent units.

FMEA on software architecture

The software risk analysis has been conducted using two complementary viewpoints:

- the *static organization* of ARTAS software architecture based on a Computer Software Component (CSC) decomposition;
- the *dynamic organization* where processes were analyzed through the different UBSS CSCIs (*Inter Process Communication, Consistent Datastore Copies, FIFOs,etc.*).

In addition, a FMEA was also performed on UBSS in order to ensure that this middleware was robust regarding internal failures. The next sub-paragraphs deal with these three topics.

Static approach

Two generic failure modes were considered when the function is directly affected by a CSC:

- **no result**; results are no longer provided by the failed CSC or are not delivered in time;
- **erroneous result**; inconsistent or inadequate results might be either provided by the failed CSC or unjustifiably delivered.

Indirect effects of failures occurring on other software components, belonging or not to the same CSCI, were also considered. Direct and indirect effects of failed CSCIs on ARTAS functions are depicted in fig. 4. Indirect effects have been expressed as *input missing* and *erroneous input* regarding to the involved function. Corrupted / missing data is explicitly indicated in the FMEA forms as shown in fig. 5 (*see next page - in this example, capabilities C1 «**To provide track information service** » and C9 «**To inform the Operator on the ARTAS system state** » are affected by the Router Bridge failures*).

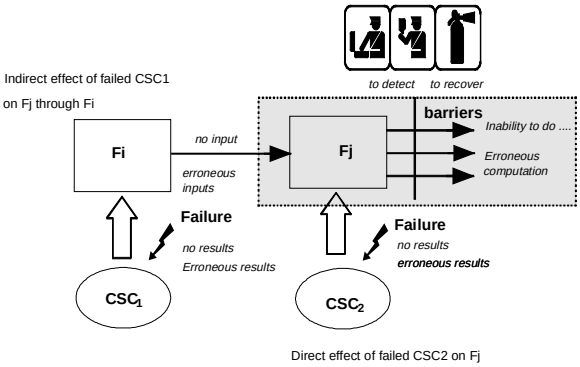

Fig.4áDirect and indirect effects of CSC failures on a ARTAS function.

[8] UBSS « UNIX Based System Software » is a © Thomson AIRSYS-ATM software product used in 25 different ATC Centers over the world.

Effects of the CSCI's failures have been assessed regarding to the functions that are affected by the failure. The effects have been expressed with respect to design requirements from the CSCIs SRS (*Software Requirements Specifications*). Failure modes have been assigned a severity level according to their impacts on the capabilities. These impacts are indicated in the FMEA worksheets.

Severity levels have been set assuming a worst case situation (*i.e. UBSS failure detection / recovery is inoperative*). On the other hand, off-line / on-line mode of operation have been taken into account; it has been considered for example, that database updates were not « critical » given that they are only authorized in off-line mode. However parameters which are modified shall thoroughly be tested with regard to bounds; moreover, prior to use the updated databases, non-regression tests must absolutely be run to show that the fault corrections, to a reasonable degree of confidence, will not adversely affect other data or functions. Severity level ranking permitted to identify and then focus on the most critical cases that should thoroughly be analyzed.

The main results of this FMEA are the following:

- identification of the CSCI's whose weight is the most important regarding to the loss of capabilities. As an example, some CSCIs can independently lead to the loss of a majority of critical capabilities, while others would only affect a few of the non-critical capabilities. This result highlighted software components that have the most impact towards the users if they failed; these components need the most attention. The FMEA performed on ARTAS sofware components provides thus useful indication concerning the test effort required in proportion to the software components reliability;

- identification of the capabilities which are the most sensitive to CSCI failures, since some of them are supported by a great number of CSCs and are thus more vulnerable than others.

In addition, a complementary analysis was performed with respect to the qualification tests applied to the CSCIs that are prone to failure and whose failure modes have been assigned a « catastrophic » severity level.

With regard to the « critical » capabilities, the FMEA enabled the mapping of the software design requirements associated to all CSCIs whose failure mode severity levels were « catastrophic » with the higher requirements expressed in the System/Segment Specification.
This mapping may facilitate the verification of the qualification tests at system level.

failedCSC	effect	impact on the outputs of the function	effects on capabilities	severity
UBSS_ output.2	Erroneous outputs	1. Risk to provide SRV with non-aircraft tracks or to omit the delivery of some aircraft tracks 2. Inability to maintain a consistent track numbering between Master and Slave	1: Loss of C1 2: Loss of C1 Only in case of switch-over. This is a very critical situation that shows the limit and weakness of the switch-over mechanism	C
RBR-CSC3.2 & RBR-CSC7.2 & RBR-CSC8.2	Input missing from **RBR-*Filter plots funct.***: *overload/ filtering alarm-tr* Input missing from **RBR-*Transform Radar Data funct.***: *cpu_overload, filtering_alarm_trk*	• Inability to handle plots overload warnings (both plots load of a radar, and total plot load) at the level of the « radar data » local Datastore. Consequently, inability to update the « radar data » local Datastore with this information. • Inability for the TRK to be warned about a CPU overload or about a end of CPU overload.	C1: Risk of not applying the foreseen overload filtering levels. This could lead to a late delivery of track information to Users (degradation and even loss of C1) C9: No real impact since the RBR also sends overload filtering alarms to MMS. The Operator can thus be informed about this overload situation	C

Fig. 5 Example from the Tracker FMEA.

Dynamic approach

In addition to the FMEAs performed on the CSCs, a risk analysis was conducted on the ARTAS processes. The aim of such a study was to systematically check that every potential cause of failure at process level were efficiently covered by UBSS fault-tolerance mechanisms.
Application processes interact with each other through the following UBSS dedicated CSCIs:

- I/O calls used for managing the interface with external devices, files and directories;
- IPC, which is a library linked to every process

allowing the Inter Process Communications using FIFOs and secured FIFOs;
- BNS (*Basic Name Server*) used to provide a mapping between names and corresponding items;
- CDC (*Consistent Datastore Copies*) in charge of managing replicated information on several nodes, in a coherent way.

For each application process the following information was provided:

- indication of which processes are affected. Note that some of them may belong to the same ARTAS CSCI (*e.g. MMS is constituted of five different processes*) although most CSCI are composed of a unique process;
- identification of the failures effect on the other processes;
- existing detection means; e.g. « The FIFO layer of the writer process detects the overflow after 1 sec »;
- recovery actions, otherwise indication whether the risk of such failure mode is significant or not.

Figure 6 gives an example of results from such a risk analysis applied on the RECording processes.

Failure mode	Involved UBSS CSCI	Impacted processes	Detection means	Recovery actions / comments
Process P1: Rec Local Recording mng - process failed				
Disk I/O: Inability to locally store data between each recording time event	IOC	**Proc-8**: Don't receive local data from Proc-1 any longer	IOC forward an error code to the application process and inform SCM of the problem	switch-over on the slave
Inability to write **REC_LOCAL_MASTER_SLAVE (FIFO)**	IPC	**Slave** can't get Master duplicated messages; this rapidly will lead to inconsistency between them	Assuming that the writer process is still alive, the only case is due to an overflow situation: The FIFO layer of the writer process detects the overflow (1 sec. delay)	This error is a fatal error from the REC point of view. Like all fatal errors, the process is killed; then the node restarts automatically. Since the node is « master », this leads to a switch-over.
Process P8: Rec Central Recording mng - process failed				
Inability to send **REC_ACK_S_FIFO**	IPCS	**MMS**	The writer will not receive the Ack and will retry (3 times every 5 seconds)	A report is provided by MMS that no ack. has been received for the considered command

Fig.6 Example of FMEA dealing with ARTAS application process.

UBSS FMEA

The ARTAS risk analysis would have not been exhaustive without assessing the consequences of failures occurring on the UBSS CSCIs. Therefore, a dedicated risk analysis consisted of examining the most important problems that might occur with regard to UBSS. The following table (*see fig. 7*) gives an illustration of information that can be found in the resulting synthesis of that specific FMEA.

Failure mode	Effect	Recovery action
unsecured FIFO's overflow	Since this is the application responsibility to handle overflow situations, they might be incorrectly handled. (*This need to be thoroughly tested for each specific case.*)	The FIFO layer of the writer process detects the overflow. The writer is thus warned about such a situation (*after 1 sec.*). Overflow problems related to FIFO have normally been settled during the integration test phase. Overflow situations should ideally be reported to SCM UBSS process and then to MMS.
deadlock on CDC	The process aborts just after having locked a CDC. Consequently, the CDC is not deallocated since the semaphore has not been removed.	This problem is correctly handled by UNIX which will deallocate the resource by removing the semaphore. Furthermore, since NPM (UBSS CSCI) can detect the death of a process, this will lead to a switch-over. This failure situation has been tested during the integration phase.

Fig.7 Example of FMEA applied to UBSS

As a result of this FMEA, some recommendations were proposed with regard to the process management optimization. For example, since UBSS is able to restart a process that has unexpectedly halted, it could be opportune to redesign the current ARTAS process partioning in order to reduce the switch-over occurrences from the « master » nodes to « slaves ». Another example concerns the NPM UBSS « parent[9] » process. Given its role towards the other UBSS processes, this process is particularly « critical » with regard to failures. This is why it has been suggested that NPM could be supervised by a small and very simple (*thus highly reliable*) process acting as a « watchdog », in charge for controling that NPM is still alive.

Feared events assessment

An operational feared event can be defined as any unacceptable event from the user viewpoint, that shall not occur or that shall occur with a probability of occurrence less than or equal than the tolerated frequency. As an example, the french CAA has defined tolerated frequencies with regard to feared event related to its national ATC centers [5]. These occurences range from 1. 10^{-4}/h to 3.8 10^{-5}/h.

Feared events have been assessed using Fault Tree Analysis technique. Fault Trees are particularly convenient to describe and then compute certain types of basic event combinations for which simple combinatory rules hold; the simplest types of combinations are AND and OR.

Both a qualitative and quantitative analysis have been performed. The following assumptions have been made in order to restrict the study:

- feared events are expressed as the consequence of the loss or degradation of the identified capabilities; for example « *Impossibility for an ARTAS User to modify the current service parameter* » or « *The Operator is no longer informed about the system state* »; a feared event express thus a « degraded mode » of ARTAS functionning;
- the fault-trees have been build considering that failure detection by UBSS is ineffective. the reason of this choice is the following:

UBSS is assumed to operates on account of fault tolerance mechanisms, each time a failure occur on the software (*as well as on the hardware*). Therefore, the feared events assessment cannot take into account neither the total loss of a process nor hardware failures; in both cases, these situations would either lead to a switch-over (*totally transparent for the User when sucessfully achieved*) or to a fatal operation situation requiring a manual restart;

- only the software has been considered since reliability block diagrams of each HWCI were already provided by the designer;
- when a feared event occurs, the involved application processes are still running and are assumed to provide users with the other capabilities not involved by this feared event.

Feared events quantification involving software was restricted to sensitive analyses since confidence on modellng techniques to assess software reliability is still subject to controversial discussions in the Academic circles[10].

The assessment of the rate of occurrence of the feared events have been performed according to the following assumptions:

- it has been assumed that those parts of ARTAS

[9] refer to UNIX terminology.

[10] refer to e.g. « Software reliability measurement, prediction & application » Musa et al. AT&T Bell lab. McGraw-Hill Book C[ie] for a complete theoretical background in the field of software reliability.

SW developed in Ada are more reliable than those coded in C++ (*the risk of introducing design faults is assumed to be less important*);

- failure rate of each process has been set taking into account an average fault density and the size of each SW components expressed in number of line of source code (*without comments*);
- faults are randomly distributed;
- two cases have been considered: first, a pessimistic case where software is assumed to have low reliable (*the rate of fault activation* $\gamma = 10^{-3}/h$) and then, an optimistic case where software is assumed highly reliable ($\gamma = 10^{-5}/h$).

The main paths of the fault-tree indicate the contribution of each CSCI to the feared event. The leaves of the fault-tree correspond to the occurrence of each possible basic event (*failures impacting only a given capability, or failures affecting simulaneously several capabilities*) leading to the feared event as shown in fig. 8. Probabilities of basic events were estimated from the frenquencies of the occurrence, deduced from the FMEAs of each CSCI.

One of the interest of such quantification is to assess the average rate of occurrence of the feared event without the support of UBSS *(which is assumed to operate on account of fault tolerance mechanism*) in order to estimate how frequent UBSS should be sollicitated to recover a nominal mode. Depending on the results (*the sensitivity analysis enables to provide a range of values*), additional preventative measures can be undertaken to increase the reliability of the most vulnerable software components. Such measures are all the more relevant if the underlying middleware does not provide an acceptable (*or required*) level of dependability.

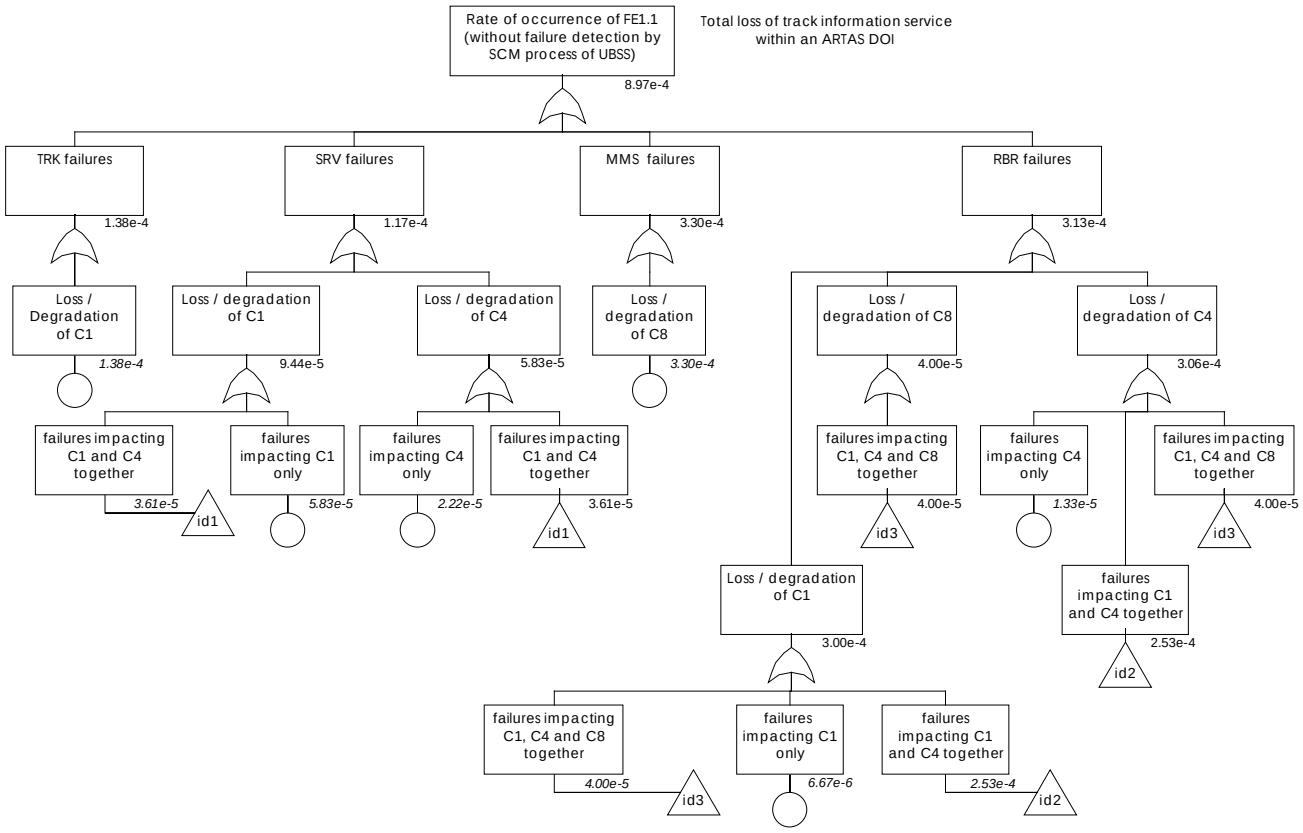

Fig. 8 Example of fault-tree

SDP systems dependability with regard to separation minima standards

The french CAA will to make sure that the french SDP systems currently in use allow the application of 5NM separation minima, even in a degraded mode of operation. In this perspective, a dependability study has recently been performed by the CENA [6], dealing with the current french radar processing system. The adopted methodology has lead to recommendations according to three main lines:

- control of the system parameterizing since some of the system configuration parameters stored in databases can have a great impact on the radar data accuracy;
- control of the system operation thanks to the identification of all failure modes that could lead to a degradation of accuracy and consequently compromise the application of a reduced separation minima;
- control of the system evolutions, especially by proposing adapted validation procedures to be applied following to software upgrades.

The dependability study achieved on ARTAS permits to have all the main elements in order to perform further risk analyses prior to the use of new generation of SDP systems, especially when the main objective is to focus on the impact of failures with regard to radar data accuracy.

Conclusions

This paper has described the methodology used to perform risk analyses on the ATM Surveillance data processing and distribution system. Partial results have been presented in this paper to illustrate the work done.

Such a dependability activity may form part of a wider approach known as « safety case » (*in practical the safety case is a set of documents including many of the elements of the safety assurance and demonstration process already employed in the aviation community for safety regulation purpose*). Nevertheless, one will agree on the benefits of such a dependability study when addressed to ground Air Navigation Systems as far as an adequate risk classification scheme is used. Outcomes issued from dependability studies are indeed tangible elements that should be part of a safety case. The notion of level of importance associated with capabilities for expressing the user point of view are key points of the methodology. Severity levels have directly been defined from them unlike what is generally proposed in most standards. As explained in the last part of this paper, a dependability process is generally be achieved to verify whether initial dependability objectives set to a system (*such as Radar Data Processing systems*), are met. However, advantage of such an approach can also be taken to make sure that such particular systems can be used in compliance with reduced separation minima (*e.g. 5NM*).

Future works

System dependability assessment is more efficient when carried out in the early phase of the system life-cycle. However the dependability study performed on ARTAS 1 was conducted when the system was in an advanced stage of development. Nevertheless, the results and lessons drawn from this activity have been useful. This is why EUROCONTROL envisage to continue using the dependability approach on future ARTAS developments.

For further information on ARTAS contact
Mr. J.M Duflot
Surveillance division (DED3)
EUROCONTROL
rue de la fusée, 96
B-1130 BRUSSELS
e-mail:jean-marc.duflot@eurocontrol.be

Bibliography

[1] ARTAS dependability study, Final Report - Eurocontrol / CENA document - 1998.

[2] DAAS, Dependable Approache to ATM Systems, Final Report - CEC DGXIII / CENA document - 1995.

[3] ARTAS: A Highly Performant, Integrated Surveillance Data Processing and Distribution System - Eurocontrol document - 199x.

[4] EATCHIP Air Navigation System Safety Assessment Methodology, Eurocontrol document - 1997.

[5] Analyses globales de la sûreté de fonctionnement de Phidias-ODS France,Note technique CENA - 1998.

[6] Etude SdF dans le cadre de l'abaissement de la norme de séparation en route 5NM, Note technique CENA - 1998.