Dynamic Reconfiguration of Safety-Critical Systems: Automation and Human Involvement

Giuseppe Montano

Ph.D. Thesis
This thesis is submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy.

The University of York
Department of Computer Science
September 2011

Abstract

This thesis describes the design and evaluation of a novel Decision Support System (DSS) for naturalistic, safety-critical decisions on-board modern aircraft. The system is intended to improve pilots' decision-making accuracy and performance, by supporting human cognitive strategies.

In recent years, the development of dynamically reconfigurable Safety-Critical Manned Systems (SCMS) has acquired increasing attention in several engineering domains including civil and military aerospace, marine and ground transportation. Dynamic reconfiguration of the on-board control systems enables adaptation to the changing conditions during operation. At the occurrence of a fault or damage, reconfiguration allows for the transition to a degraded operating mode by deactivating a number of services in order to preserve sufficient resources for the provision of essential functionality.

The current focus of mainstream research is on full autonomy and full authority solutions, which nonetheless make the transition to a degraded mode transparent to the operator, as much as possible. This thesis takes a different approach, developing a human-centred perspective: by drawing on well-established fields such as Cognitive Engineering, Human Factors and Naturalistic Decision Making, it identifies limitations of fully automated dynamic reconfiguration solutions, including some safety problems, and proposes novel technology to keep the operator much more effectively "in the control loop" during reconfiguration.

A review of the relevant literature leads to the identification of three main research problems: (a) determining the characteristics of effective decision support information for SCMS dynamic reconfiguration decisions; (b) developing DSS technology to autonomously generate the type of information required; (c) developing a methodology to evaluate and validate the performance of the DSS and assess its effectiveness in support of the decision making activity.

First, pilot behaviour during fault management decisions is investigated and a novel design for decision support information that parallels human cognitive strategies is devised. The hypothesis advanced is that decision support information that favours mental simulation by including (a) explanations that justify each reconfiguration alternative, (b) implications for each alternative and (c) an assessment of the uncertainty embedded in the sensor information would have a positive impact on both human decision accuracy and performance.

Second, a novel Constraint-based DSS is developed to generate the type of information suggested by the research hypothesis. A number of algorithms and software applications designed to handle the reconfiguration process and generate decision support information are developed and their performance is assessed. The tools developed are integrated into the Safe and Interactive Reconfiguration Architecture (SaIRA), a novel framework for automated decision support.

Third, seven experiments, which involved thirteen civilian aircraft pilots, were performed to (a) empirically verify the claims advanced throughout the thesis concerning the issues with automation and human involvement during SCMS dynamic reconfiguration, and (b) to assess the effectiveness of SaIRA. A validation methodology that merges a number of relevant objective and subjective metrics is proposed. The experiments reveal that SaIRA improves pilots' decision accuracy, decision performance, situation awareness and, more generally, their cognitive readiness whilst reducing cognitive workload and frustration under heavy time pressure. Whilst this work has been undertaken in the context of civil aviation systems, there is reason to believe such classes of decision support system would be of much wider applicability.

Contents

List of Tables

List of Figures

Acknowledgements

Starting from the most obvious, but truly sincere acknowledgement, I want to thank Prof. John McDermid who supervised this Ph.D. programme. I want to thank John for having picked up the phone in the late evening of Friday the 20th July 2007, for having listened to all the details of my Ph.D. research proposal for hours, and for finally believing in me and having found a way to fund this project in less than 24 hours. For driving and focusing my passion, for giving precious support at hard times, and for timely adjusting my research direction always with a warm and reassuring smile on the face, thank you John.

I am extremely grateful to Dr. Paul Cairns who, after having reviewed my initial experimental design, replied with a list of half a dozen books about statistics and psychology experiments design in preparation of our next meeting. Paul gave a crucial contribution to the design of the experiments presented in this thesis and was pivotal in my understanding of statistics.

My warm acknowledgment also goes to Dr. Iain Bate, my internal examiner, whose sharp comments at the beginning of this research programme revealed to be decisive in defining the final research direction.

The fourth acknowledgement is for Prof. Aldo Franco Dragoni, from the Università Politecnica delle Marche, for having helped me "forcing" the Italian academic bureaucracy to the benefit of my research career when needed and for motivating me to start a Ph.D. programme abroad.

I extend my acknowledgement to the whole High Integrity Systems Engineering group from the Department of Computer Science of The University of York for having funded this Ph.D. programme during the first two years of full-time enrollment.

I also want to thank Prof. Nerendra Jussien, Head of the Department of Computer Science of the Ecóle des Mines de Nantes, who hosted me in his Department when I was working on the Constraint Programming based algorithms presented in this thesis.

I thank Filomena Ottaway, Debbie Haverstock and Chrysle Hudson from the Department of Computer Science in York for the administrative and technical support during the whole Ph.D.

An exceptional thank goes to my father Silvano Montano, my mother Maria Rita Catalano, my special, *sui generis* sister Damaris, and the rest of my family. Their love and unconditioned support gave me the strongest motivation to complete this research programme.

Daiana Mattioli, Rafael Fidanza, Christofer Leone, Lusiana Mattioli, Daniela Rosania, Ellie Marchione, Rachel Freeman: these have been my friends (in no particular order) for a lifetime. Despite the geographical distance, each of them supported me in crucial circumstances of this long research journey, sometimes even unconsciously. Their support was invaluable at hard times and the fun we had together was momentous for the success of this enterprise.

I am in debt to my friend Juan Perna, who helped review the thesis. I am also grateful to all the other friends I made in York: Bernadette Martinez-Hernandez, Leo Freitas, Pierre Andrews, Silvia Quarteroni, Enda Ridge, Jan Tobias Muehlberg, Frank Zeyda and Tom Lampert. Each of them enriched me in a different way and gave a (non-necessarily scientific) contribution to this thesis. They made the stay in York a priceless experience.

I also want to thank my colleagues and friends from EADS Astrium Ltd for the moral support they provided throughout the second half of the Ph.D. programme, which I spent in London. In no particular order, they are Christian Corba, Sara Mugnaini, Alessio Brizzi, Davide Zilio, Alessandra Venturini, Giovanni Cavallo, Alessandro Comune, Lorenzo Serafini, Egidio Collavo, Silvia Raffaelli, Alexander Nucera and Angelo Povoleri. Their words of encouragement (besides the jokes about the Ph.D. deadline) were precious.

Last, but absolutely not least, a very special thank you goes to my fiancée, Melania Pulvirenti. For being so supportive in all circumstances throughout the four years of this Ph.D., for understanding the stress I was subject to, for having sacrificed time together, for having left me with the freedom to follow my scientific interests unconditionately, thank you Melania.

Author's Declaration

This thesis has not previously been accepted in substance for any degree and is not being concurrently submitted in candidature for any degree other than Doctor of Philosophy of The University of York. This thesis is the result of my own investigations, except where otherwise stated. Other sources are acknowledged by explicit references.

Some chapters of the thesis are based on articles that I published in the peer-reviewed scientific literature during the course of the research work, as follows:

1. Giuseppe Montano and John McDermid [2008], *Autonomous and/or Interactive Constraints-based Software Reconfiguration for Planetary Rover*, in Proceedings of the 10th European Space Agency Workshop on Advanced Space Technologies for Robotics and Automation (ASTRA 2008). ESA Communication Production Office.

2. Giuseppe Montano and John McDermid [2008], *Survivability Management for Integrated Modular Safety-Critical Space Systems*, in Proceedings of the Third European Space Agency IAASS Conference (International Association for the Advancement of Space Safety). ESA Communication Production Office. Vol.1, Part 20.

   Available at: http://www.esa.int/TEC/Robotics/SEMABJC4VUE_0.html

3. Giuseppe Montano and John McDermid [2008], *Human Involvement in the Dynamic Reconfiguration of Integrated Modular Avionics*, in Proceedings of the 27th IEEE/AIAA Digital Avionics Systems Conference, 2008 (DASC 2008). Vol. 4.A, pp. 1-13. Digital Object Identifier: 10.1109/DASC.2008.4702821

4. Giuseppe Montano, John McDermid and Paul Cairns [2011], *Effective Naturalistic Decision Support for Dynamic Reconfiguration Onboard Modern Aircraft*, in Proceedings of the 10th Naturalistic Decision Making Conference (NDM 2011), 31st May - 4th June 2011, Orlando, FL, USA.

I hereby give consent for my thesis, if accepted, to be made available for photocopying and for inter-library loan, and for the title and summary to be made available to outside organisations.

Signed ..... (candidate)

Date .....

Dedicated to the very few who know about Giuseppe and about Napo,
and have been obstinate enough to fathom out, without judgement,
starting with my parents.

Chapter 1

Introduction

This thesis describes the design and evaluation of a novel Decision Support System (DSS) for naturalistic, safety-critical decisions on-board modern aircraft. The system is intended to improve the pilot's decision-making accuracy and performance, by paralleling human cognitive strategies (e.g. supporting mental simulation).

Consider the cockpit cabin of a modern Boeing 737 aircraft. A severe fault suddenly affects the left engine. The left power generator is lost; sensors detect a high temperature in the surrounding area, the data handling system manager acknowledges the loss of communication with on-wing devices. Alarms sound in the cabin and the warning lights start flashing. Flashes, coloured lines and flickering lights reach the pilot's photoreceptors and nerve cells at the back of the eye. Light is converted into electrochemical signals, which climb up to the lateral geniculate nucleus in the thalamus. From there, signals are sent to visual area 1 (V1) which feeds areas V2, V3, V4 amongst others. Similarly, signals from the other senses reach the thalamus, a sort of signals switchboard[1], and are forwarded to specific brain areas. A huge amount of information reaches the dorsal pathway, flowing towards the parietal lobes, which allows the pilot to localise the flashing lights on the cockpit, locate the buttons, work out how to act on them and guide the necessary movements. Other signals are diverted to the ventral pathway and reach the temporal lobes, which allow the pilot to identify which type of alarm has just sounded. These bottom-up processes, driven by sensory information, are integrated with top-down processes, which are driven by the pilot's knowledge, expertise, goals and expectations. As time passes, the risks are increasing and, as a result, the electrochemical activity in the amygdala rapidly increases, signals from the brain reach the sweat glands on the pilot's skin. The attention is narrowed, the pilot starts becoming overloaded with information, the complexity is too intense and he or she starts to feel frustrated.

The pilot struggles to produce a mental representation of the current situation and the brain searches for similar events in the long term memory. Pre-motor, motor, parietal cortex and cerebellum are processing information rapidly to allow for the mental simulation of each potential decision alternative. This allows the consequences of each course of action to be calculated. Nev-

[1] An important observation must be made that it is becoming increasingly clear that the notion of a pure relay station is too simple here, as there are generally many more back-projections from the cortex to the thalamus compared to the forward projections between the thalamus and the cortex [Trappenberg 2010]. However, the simplification made here seems appropriate for the qualitative nature of this section.

ertheless, there is likely to be some information missing. The pilot struggles to organise the amount of data the cockpit is delivering. As time passes, the risks increase dramatically.

This extremely simplified description of the paths followed by a tiny portion of the information processed by the brain reveals how the *amount* and *type* of information generated by the cockpit is crucial, during safety-critical fault management decisions, to enable the pilot to work out a solution to such a complicated problem in a timely manner. The human brain is an astonishingly efficient problem solver but its decision-making effectiveness can be severely compromised by unfavourable framing of input information coming from the control system interface [Payne et al. 1998]. A solvable problem could become unsolvable with the wrong information framing. Too much data, too little data, display of the causes instead of the consequences of a fault (or vice versa), text or graphics, colour and flickering and frequency of alarms are just a few of the variables which have to be taken into consideration when designing cockpit instrumentation.

Making the right decisions during the reconfiguration of a safety-critical manned system during operation (e.g. whilst airborne if the system is an aircraft), under time pressure and uncertainty, is an extremely complicated problem for the system operator. The design of appropriate support for the decision-maker is the central problem of this thesis.

In response to the complexity of controlling modern aircraft, mainstream research is focusing on increasing the levels of autonomy and authority of the system. The vision is to make certain fault management processes completely transparent to the pilot, leaving her with just the task of 'flying' the aircraft. As discussed in detail in the following chapters, this approach has inherent drawbacks in terms of safety. For instance, a number of documented accidents (examined below) reveal that excessive automation is likely to lead to *cognitive mismatch*, a disparity between the operator's mental model of the system and the way the system is really working [Baxter et al. 2007]. If the automation fails to react to a fault or if the pilot detects that the system is not behaving correctly, she is not in a position to take control because the previous actions were masked from her. Additionally, as seen in the simplified scenario described above, fault management decisions usually need to be made in an extremely short time, hence the pilot doesn't have the time to retrace the previous actions of the system and achieve a reasonable degree of situation awareness.

This thesis takes into consideration a specific type of fault management process which is typical of next-generation aircraft and is currently subject of debate in both the academic and industrial communities: avionics dynamic reconfiguration. In brief, at the occurrence of a fault or damage, modern aircraft allow for the relocation of functions running on affected computing modules to other healthy modules. This is an extremely complex process because hundreds of functions run on dozens of computing modules on-board modern aircraft; the functions have different criticality levels and are connected by dependency relationships. At the occurrence of a fault or if the aircraft is damaged in-flight, deciding which functions should be deactivated and which should be kept active, having evaluated the consequences of each option, is a problem that goes beyond human cognitive capabilities. At the same time, changing the functionality of the aircraft whilst airborne, without properly involving or informing the pilot, is a dubious safety option; the complexity of the system in conjunction with the unstructured scenario of operation make the development of dependable, fully-automated, reconfiguration technology extremely difficult (these

topics are discussed in detail in Chapter 3).

This thesis proposes a human-centred alternative to full automation; the pilot is actively involved in the control of the fault management process by making critical decisions (e.g. deciding which configuration to apply between two options), but throughout the process she is assisted by bespoke DSS technology, integrated into the cockpit instrumentation.

An effective DSS for avionics dynamic reconfiguration must be able to (a) handle the reconfiguration process by generating configurations that can mitigate the effects of the fault or damage which triggered the reconfiguration, and (b) generate decision support information that allows the pilot to make an informed decision in a timely manner, reducing the opportunities for errors. The analysis presented in the following two chapters reveals a considerable number of technical challenges and obstacles that are hidden in the two main requirements. They can be divided into five main groups, as follows:

- **Decision-maker profiling.** A crucial element in the study of a decision-making problem is the characterisation of both the decision context and the factors influencing the decision maker. This type of analysis leads to the definition of a user profile. An effective way of shaping the content and form of the decision support information produced by the DSS is 'moulding' it on a user profile [Dale and Reiter 1995]. Material from the aviation psychology literature can be used to study pilot behaviour in the system dynamic reconfiguration context. However, avionics dynamic reconfiguration is a new technology and reconfiguration decisions have peculiarities that require appropriate investigation, e.g. a combinatorial problem underlies the decision-making problem, which is an uncommon type of decision for pilots.

- **Decision support information characterisation.** Once both the features of the decision-making problem and the decision-maker's profile are available, the type of interaction between the system and the human must be characterised, the content of the information must be defined and the way the information is presented to the decision maker must be delineated.

- **Dynamic reconfiguration technology.** The DSS must be integrated with the fault management technology in order to provide a practical alternative to a fully automated solution. Given the novelty of the human-centred approach proposed in this thesis, bespoke technology must be developed to demonstrate the practicability of the ideas promoted. The technology devised must be able to fuse sensor information with pre-defined fault management data in order to generate configurations that can mitigate the effects of the fault detected. Additionally, the developed process must allow for the automated extraction of decision support information for the pilot.

- **Validation methodology.** A method to validate the novel system proposed in this Ph.D. programme must be identified. The method must allow for the assessment of both the performance and effectiveness of the DSS.

- **Validation technology.** The technology to enable a robust validation of the system needs to be identified or developed, if unavailable.

The above challenges have been addressed as follows:

- **Decision-maker profiling.** Unlike mainstream profiling methods which focus on capturing the behavioural aspect of decision making, this thesis proposes a novel user profiling approach which draws on ideas from the Naturalistic Decision Making (NDM) domain and focuses on the cognitive aspect of decision support. A number of factors that are likely to affect pilot decision behaviour during avionics reconfiguration are studied; this information is then used to develop a user profile.

- **Decision support information characterisation.** This thesis develops a novel design for decision support information, intended to assist the decision maker by generating information that parallels human cognitive strategies. For instance, Chapter 2 shows that mental simulation plays a crucial role in decisions of the type analysed in this research programme. As as a consequence, the decision support information generated by the proposed system automatically produces the implications of each decision alternative and displays them in a concise way on the cockpit display, relieving the pilot of the task of calculating them in real-time (an activity that is likely to go beyond human capabilities in such a complex problem).

- **Dynamic reconfiguration technology.** Based on previous work in the Constraint Programming (CP) domain, novel CP-based models, algorithms and software tools have been developed to solve the avionics dynamic reconfiguration problem and manage the interaction with the pilot. All the technology developed in this Ph.D. programme is integrated into the Safe and Interactive Reconfiguration Architecture (SaIRA). The SaIRA concept is implemented in a real demonstrator and used during the human-computer interaction experiments presented in the final part of the thesis.

- **Validation methodology.** Two aspects of SaIRA require validation, its *performance* in generating system configurations and its *effectiveness* in supporting the pilot. Benchmarking experiments were designed and carried out in order to assess the performance of the core algorithm proposed for the generation of decision support information. Regarding effectiveness, a significant contribution of this thesis is the development of a validation methodology that goes beyond classic methods, which focus only on the decision outcome, and instead merges sophisticated subjective and objective techniques, such as eye movement analysis, mental workload and situation awareness estimation, to assess the effectiveness of the system.

  This methodology allows for robust conclusions, which take into consideration not only the decision accuracy obtained, but also the human behaviour observed during the decision making process. It is worth noticing that, unlike other decision-making contexts, the effectiveness of keeping the mental workload as low as possible is a parameter of quality for cockpit processes and instrumentation.

- **Validation technology.** In order to validate the performance and effectiveness of SaIRA in accordance with the pre-defined methodology, a number of tools have been designed,

developed and integrated with FAA[2]-approved flight simulation technology. Bespoke eye-tracking technology has been developed specifically for this study (Appendix A).

The groundwork for the research hypothesis of this Ph.D. programme emerges from the strategy set up to address the challenges of the central problem: *during the process of avionics dynamic reconfiguration, decision support information that parallels cognitive strategies should have a positive effect on pilots' decision-making performance and accuracy, thus it should improve the safety of the process*. The hypothesis is not finalised at this stage because propaedeutic ideas need to be presented in order to augment it and appreciate the rationale behind its final formulation.

The remainder of the thesis follows approximately the order of exposition of the challenges given above, as described in the next section.

1.1 Thesis Structure

Preliminaries (Chapter 2) provide propaedeutic information which is crucial to introduce both the central research problem and the rationale behind the research hypothesis. Background information is provided about (a) the topic of dynamic reconfiguration of safety-critical systems in general, (b) the problem of human involvement in the process in question, and (c) decision support systems engineering. The literature from all three domains is critically discussed, leading to details of the challenges which need to be addressed. Recent criticism of modern validation techniques from the Human Factors domain is also discussed, providing the basis for the development of a new validation methodology that goes beyond the criticism.

Chapter 3 questions the viability and safety of full autonomy and authority solutions to the avionics dynamic reconfiguration problem. A number of accidents, mainly from the aviation domain, are used to contribute to the argument. Two mental constructs, situation awareness and mental workload, are used to catalyse the discussion of the inherent drawbacks of both excessive autonomy and insufficient autonomy. The central hypothesis of research can then be formulated.

An extensive discussion of the factors that are likely to influence the pilot during avionics reconfiguration decisions is presented in Chapter 4. This material implicitly characterises the user profile for the proposed DSS. Subsequently, the profile is used to design the type of decision support provided by SaIRA; a number of claims about both pilot behaviour during reconfiguration decisions and the impact of the DSS are made and then organically connected to the research hypothesis.

Once all the required features of the decision support information are elaborated, Chapter 5 presents SaIRA in detail and reveals the logic implemented to generate the information required. Algorithms, software tools and techniques devised to solve the problems in the context of this Ph.D. programme are presented.

Chapter 6 provides a detailed description of the human-computer interaction experiments performed to validate SaIRA. Experimental methods, tools and results are discussed.

Finally, the conclusions are given in Chapter 7, along with the limitations and further work.

[2]Federal Aviation Administration (FAA) — http://www.faa.gov

Additionally, Appendix C illustrates how Evidential Reasoning algorithms have been assembled within SaIRA in order to represent uncertainty of sensor readings. Appendix A concludes the manuscript with a description of the tools developed to perform the human-computer interaction experiments, allowing for the reproducibility of this study.

This thesis has four core aims: a) identifying the limitations of fully automated approaches to the dynamic reconfiguration of SCMS, b) proposing an effective and practicable human-centred alternative in which the system operator is effectively involved in the system control loop, c) designing effective automated decision support technology to improve the operator's decision making accuracy and performance by paralleling human cognitive strategies, and finally d) empirically verifying the effectiveness of the framework and ideas proposed. The thesis develops and presents a DSS philosophy and system which respects the mental processes outlined at the beginning of this chapter, and shows the effectiveness of the approach through experimentation.

Chapter 2

Preliminaries

> Ποταμοισι τοισιν αύτοισιν έμβαίνουσιν, ηετερα καί ηετερα ηυδατα έπιρρει
> *Ever-newer waters flow on those who step into the same rivers.*
>
> Heraclitus of Ephesus, Greek philosopher
> c. 535 — c. 475 BCE

Heraclitus is famous for his doctrine of *change*. The environment of systems operating in the real world change; change happens, change is unavoidable, change is usually not predictable. In fact predictions are as reliable as the knowledge available to the agent who makes the prediction, and this knowledge is undoubtedly characterised by epistemic uncertainty, which increases with the complexity of the system under examination. In this light, change can be a threat to the safety of humans interacting with complex systems operating in complex, unstructured environments. Indeed, an argument can be made that a safety-critical system cannot be regarded as such unless adaptability to the changing conditions of its environment is accounted for in its design.

This thesis is about the process of adaptation of modern, manned, high-integrity systems to the changing conditions of the environment they operate in. The intention is to contribute to the improvement of the safety of next-generation high-integrity systems. **High-integrity systems** are complex, software controlled systems, which, in the event of failure, have a high impact on humans, the environment, organizations and society. They can be divided into two fields of application:

- **Safety critical systems (SCS)**. SCS have a direct influence on the life and health of humans and on the environment. Examples can be found in all areas of industry, such as aerospace, automotive, railway and marine systems, power generation and medical technology.
- **Mission critical systems (MCS)**. MCS possess a high criticality with respect to the functioning of an organization.

This research focuses specifically on **Safety-Critical Manned System (SCMS)**, a particular type of safety-critical system in which a human operator is involved in the control of the system and, to a certain extent, can be regarded to as part of the system. For ease of exposition, the

Introduction approached the research problem focusing on manned aircraft, which is a typical example of a SCMS; in the rest of the thesis, the discussion is extended to other SCMS in order to generalise the conclusions as far as possible.

Figure 2.1 provides a map of the following chapters of the thesis and their aims. This chapter contains a critical review of the literature from heterogeneous domains which are propaedeutic to the study of effective solutions for the management of the dynamic reconfiguration of SCMS. Firstly, the state-of-the-art technology for the dynamic reconfiguration of SCMS is presented. Then, ideas from the Naturalistic Decision Making field are put into the context of the decisional problem being addressed. Finally, relevant ideas and state-of-the-art technology from the Decision Support Systems engineering domain are discussed.

With reference to Figure 2.1, the material included in this chapter is propaedeutic to Chapters 3, 4 and 5. Furthermore, the background material related to the problem of human involvement (Section 2.2) is important for the definition of the empirical evaluation approach adopted in Chapter 6.

2.1 Dynamic Reconfiguration of Safety-Critical Systems

In this section, the state-of-the-art technology for the dynamic reconfiguration of SCMS is reviewed.

Section 2.1.1 should be considered as a preamble to the remainder of the thesis and is deliberately written in slightly controversial and informal terms. The tone of the exposition, however, is not meant to diminish its relevance, on the contrary, its content represents the starting point of the research journey followed by the author and provides a key motivation for the research presented in the following chapters.

2.1.1 Integrity and Safety

Integrity.

It is "one of the most important and oft-cited of virtues. It is also perhaps the most puzzling" [Stanford University 2010]. "Integrity" stems from the Latin adjective *integer* (whole, complete). The Babylon Dictionary [2010] defines it as follows:

1. the state or quality of being entire or complete; wholeness; entireness; *unbroken state*;
2. moral soundness; honesty; freedom from corrupting influence or motive;
3. *unimpaired, unadulterated, or genuine state*; entire correspondence with an original condition; purity.

This definition is projected into the industrial engineering of developing systems with substantial safety, availability, reliability and robustness requirements [Leveson 1995]. In force of this praxis, this research journey started with the aim of providing a small contribution to the development of *high-integrity* solutions to improve the safety of next generation, dynamically reconfigurable aircraft. The initial expectations, however, changed substantially in the light of the ideas discussed below.

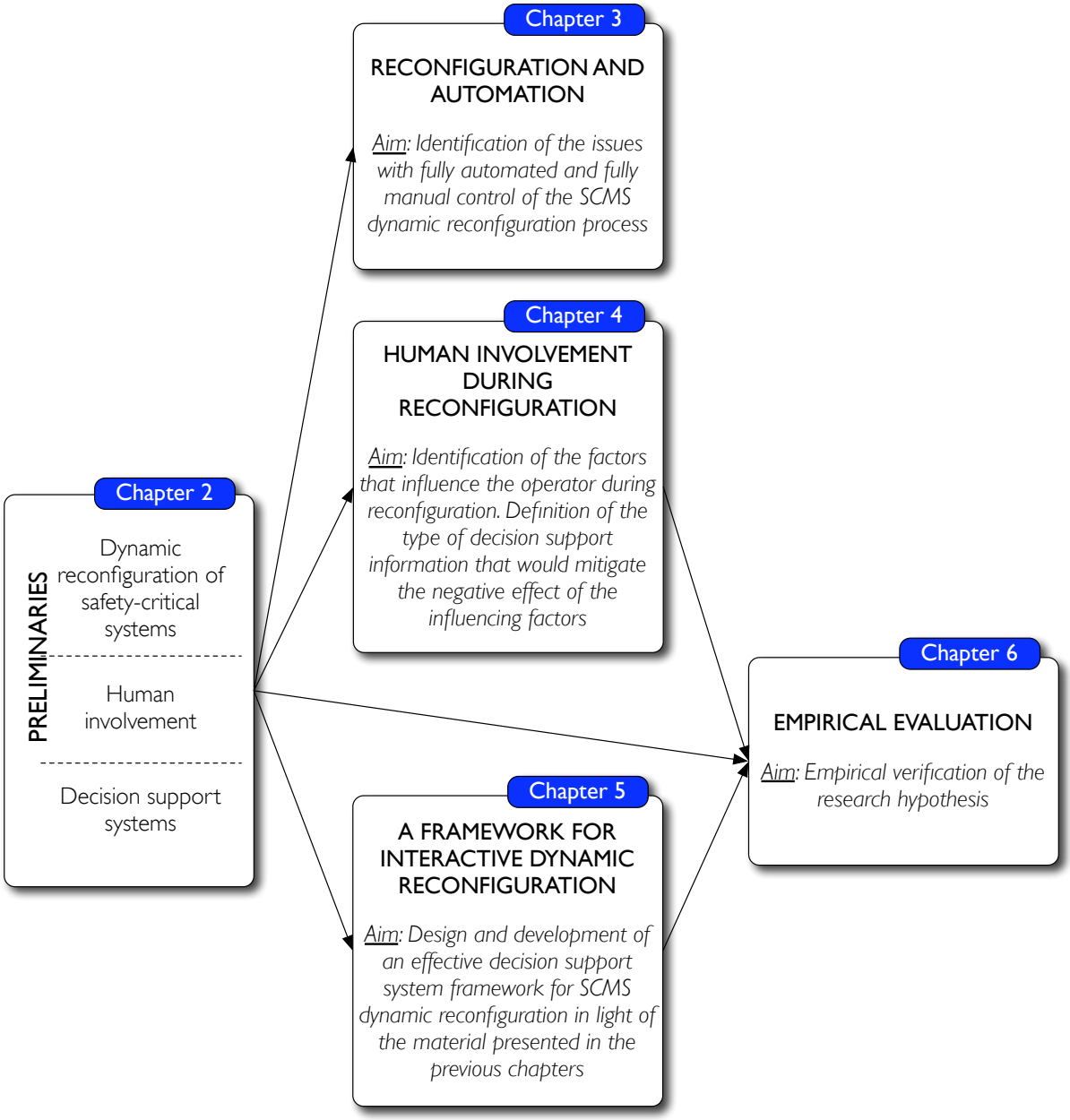


Figure 2.1: Map of the following chapters of the thesis and their aims.

The Greek philosopher Socrates (469 ~ 399 BC) was an exemplary model of integrity, he placed his life at risk by refusing to carry out orders that were immoral. His lack of repentance infuriated the court, which then condemned him to death. Whilst in prison, before being given poisonous hemlock to drink, some friends visited him and planned his escape. But Socrates refused to flee, maintaining that although the charges against him were unjust, they were made by a legitimate court and therefore must be obeyed.

Integrity is not meant to preserve the safety of human beings. Socratic integrity, as it is referred to in the literature, is about avoiding alterations to (pre-defined) moral behaviour. Making a parallel with modern systems, integrity is about avoiding *improper* alterations to a defined system

[Avizienis et al. 2001]. However, is it possible to define, *a priori*, what is a proper and what is an improper alteration of a complex, modern system like a aircraft, operating in an unstructured environment?

At this point, it is useful to consider the following accident scenario:

Accident 2.1 *On the* 14^{th} *September 1993 the Airbus A320-200 (Lufthansa Flight 2904) was cleared to land at Okęcie International Airport (Warsaw, Poland). Pilots were informed of wind-shear conditions on the runaway, so they tried to compensate by touching down with the aircraft banked slightly to the right.*

Shortly before touchdown the wind conditions changed from a cross-wind to a tail-wind. The right gear touched first, the left gear touched after 9 seconds and only at that point did the ground spoilers and engine thrust reversers deploy because these automatic systems depend on oleo strut (shock absorber) compression.

The pilots managed to land but the aircraft was running too fast, it overran the runaway and hit the embankment and an LLZ aerial with the left wing. Two people out of seventy died, one of whom was the co-pilot.

[Aviation Safety 1993]

Airbus high-integrity technology for flight management is the successful result of decades of study, analysis and tests. It would not be possible for pilots to control such complex machines without the support of automation. Airbus engineers developed the logic that controls the spoilers in a way that they are armed after having touched the ground. The reason for this behaviour is that landing gears have shock absorbers which communicate to the aircraft when it is on the ground. When the gear compresses, the logic tells the aircraft that it has landed [Dekker 2001].

The mechanism that controls the spoilers and thrust reverser is high-integrity and even though it was not subject to any improper alteration during Accident 2.1, people's safety was impaired. In this regard it is interesting to consider the definition of **complexity** given by Weaver [1948]:

> *The complexity of a particular system is seen as the degree of difficulty in predicting the properties of the system, if the properties of the system's parts are given.*

Weaver's idea of complexity, which has strongly influenced contemporary thinking, clearly unveils the difficulty (to use an euphemism) of defining proper and improper system alterations *a priori*. In real, unstructured environments the high-integrity of a system can guarantee the safety of people only in the context of circumstances that can be foreseen by system designers. If the system does not allow overriding of its pre-defined logic, high-integrity can have the unintended effect of undermining safety.

This concept is captured by Avizienis et al. [2001], who interpret system integrity as *one of the dimensions* of **dependability**. Avizienis et al. define dependability as "the trustworthiness of a computing system which allows reliance to be justifiably placed on the service it delivers" and encompasses the following attributes[1]:

[1]Confidentiality, the absence of unauthorized disclosure of information, is also included in the list of attributes of dependability, however, it is only applied when addressing system security, which is outside the scope of this thesis.

- *Availability* - readiness for correct service;
- *Reliability* - continuity of correct service;
- *Safety* - absence of catastrophic consequences on the user(s) and the environment;
- *Integrity* - absence of improper system alteration;
- *Maintainability* - ability of a process to undergo modifications and repairs.

As discussed below, a significant body of research tackles the problem of improving the safety of reconfigurable systems from the point of view of high-integrity engineering. Outstanding results in this domain have been achieved by employing formal methods (e.g. Knight et al. [2003]; Strunk and Knight [2004]) and by integrating safety requirements early in the development process (e.g. Blackwell et al. [1999]; Rushby [2002]; Bate [2003]; Bate et al. [2003]).

This Ph.D. programme extends current research by approaching the problem orthogonally. In the light of (a) the complexity of modern systems, (b) the impossibility of forecasting *a priori* all possible operating conditions and (c) the consequent limitations of classic high-integrity engineering methods, this thesis investigates novel solutions for the dynamic reconfiguration of SCMS *which involve the system operator at run-time*, allowing the operator to override part of the logic of the system in the interest of safety.

2.1.2 Safety-Critical Integrated Modular Systems

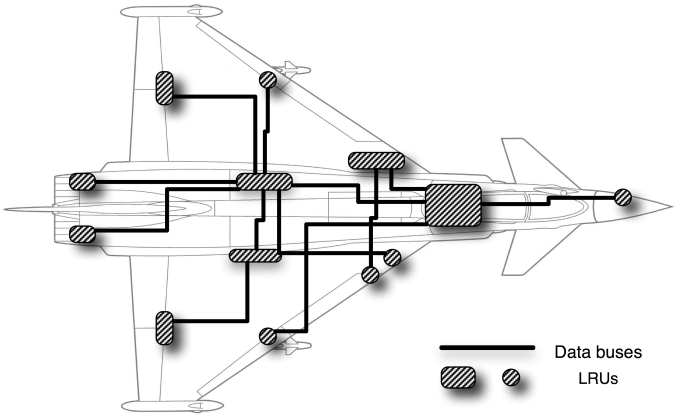
In recent years, aviation systems have been developed as *federated systems*, with each major function, or application, in a separate hardware unit [Conmy and McDermid 2001]. As a consequence, aviation applications are physically separated from one another, and are often developed and maintained by different companies.

The aviation industry is moving towards a new approach to the development of avionic systems: Integrated Modular Avionics (IMA). The IMA concept, in brief, is an airborne real-time computer network consisting of sensors, actuators and a number of computing modules capable of supporting numerous applications of differing criticality levels[2] (see Figure 2.2).

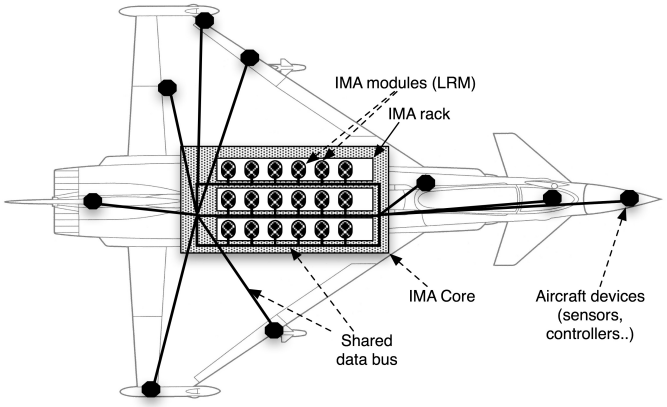
The IMA concept is currently applied, with different degrees of adherence to the standards, by the largest aircraft manufacturers in their latest products (e.g. Common Core System (CCS) by Boeing, OpenIMA by Airbus and Modular Data Processing Unit (MDPU) by Thales). At the time of writing, the IMA of the Airbus A380 represent the state-of-the-art implementation in the civil domain [Itier 2007]. Dramatic improvements have recently been announced in the military domain for the upcoming F-35 Joint Strike Fighter aircraft by Lockheed Martin [Sutterfield et al. 2008; Simmonds and Nesterov 2010].

In a full IMA implementation (i.e. totally adherent to the standards) each computing module, called a 'Line Replaceable Module' (LRM), is able to safely run all of the avionic software applications. The reliability of the execution of on-board flight management functions is guaranteed by real-time operating systems equipped with a battery of fault-tolerant facilities like space and

[2] See RTCA Inc. [1992] for more information on airborne software criticality levels.



(a) Legacy federated avionics



(b) Integrated Modular Avionics

Figure 2.2: Federated avionics (a), and Integrated Modular Avionics (b) architecures

time partitioning of the execution space, predictable multi-level scheduling policies, functional redundancy, and others (see Krodel and Romanski [2008] for a review).

Filyner [2003] identifies two main benefits of the IMA architecture. *Common processing subsystems* enable multiple applications to share the same computing resources and *software abstraction* removes the dependency of the application layer from the underlying physical systems.

A number of standards in both the civil and military domains provide the basic guidelines for the development of IMA architectures. In Europe, EUROCAE ED-79 [EUROCAE 1996] is the reference for the civil market; the ARINC 653 [Aeronautical Radio Incorporated (ARINC-653) 2006] provides guidelines for the IMA software[3]. The main European IMA standard for the military arena is defined by the Allied Standards Avionics Architecture Council (ASAAC) and is known as 'Def Stan 00-74' [UK Ministry of Defence 1966].

The aviation field represents only one dimension of a general industry trend towards Integrated Modular Systems (IMS) as a response to the exponential increase in the functions and sub-systems of modern SCMS. In the marine domain, Hughes et al. [2006] describe the Integrated Reconfigur-

[3][Coutinho 2008] puts forward interesting extensions to the standard, especially concerning dynamic reconfiguration.

able Intelligent System (IRIS) which is an IMS for military ships. Goodchild and Whiston [1998] present SHIMA, a Small Helicopter IMA. Wills et al. [2000] put forward an IMA implementation for an Uninhabited Aerial Vehicle (UAV). In the space arena, the author (as an EADS Astrium Ltd employee) is heavily involved in the design and development of two next-generation IMS architectures for spacecraft: (a) the 'Dynamically Reconfigurable Processing Module for Future Space Applications' (DRPM) [Montano et al. 2010] and the 'Modular Architecture for Robust Computing' (MARC) [Gasti et al. 2007].

This thesis extends current research in the SCMS engineering field by taking the IMA architecture as a model and investigating ways of managing the process of dynamic reconfiguration when the human is 'in the loop' (the problem is better characterised later in the chapter). As IMA is a specific case of IMS, most of the conclusions reached can easily be generalised to domains other than aviation.

2.1.3 IMA Dynamic Reconfiguration

The modularity and flexibility of the IMA architecture enables advantage to be taken of the possibility of reconfiguring the avionics of the aircraft, in flight, to adapt the functionality to changing conditions. The changing conditions can be *planned* (e.g. mode-change, conditions change, alteration of mission objectives in the case of military aircraft) or *unplanned* (e.g. faults). In the domain of adaptive embedded systems, Trapp and Schürmann [2003] refer to the two types of reconfiguration as *function-based adaptation* and *fault-based adaptation*. The authors use the concept of 'graceful degradation' of service in relation to the latter form of adaptation.

Dynamic reconfiguration brings important benefits to safety-critical systems engineering, including improved dependability, improved mission performance, reduced operational and maintenance costs (see Rushby [2002]; Parkinson et al. [2003]; Jolliffe and Nicholson [2005] for a comprehensive analysis).

Figure 2.3 illustrates the dynamic reconfiguration process. Each Line Replaceable Module runs a real-time, partitioning operating system (Module OS). Each partition runs one or more applications, which, optionally, reside on another 'local', partition-level operating system (Partition OS). This level of segregation avoids faults in one partition jeopardising the execution of the software running within other partitions (see Aeronautical Radio Incorporated (ARINC-653) [2006]). By pooling the computing resources and allowing them to be shared by different subsystems, when a fault occurrs the affected software can be relocated to another healthy LRM.

Approaches and Techniques for Dynamic Reconfiguration

A review of the literature shows that the problem of IMS dynamic reconfiguration has been examined from different perspectives already.

The ASAAC standards define the **reconfiguration** of integrated avionics as *the transient activity between two ultimate states of the system* [NATO 2005*a*;*b*].

A similar definition is proposed by Strunk et al. [2004]: *reconfiguration is the process through which a system halts operation under its current source specification S_i and begins operation*

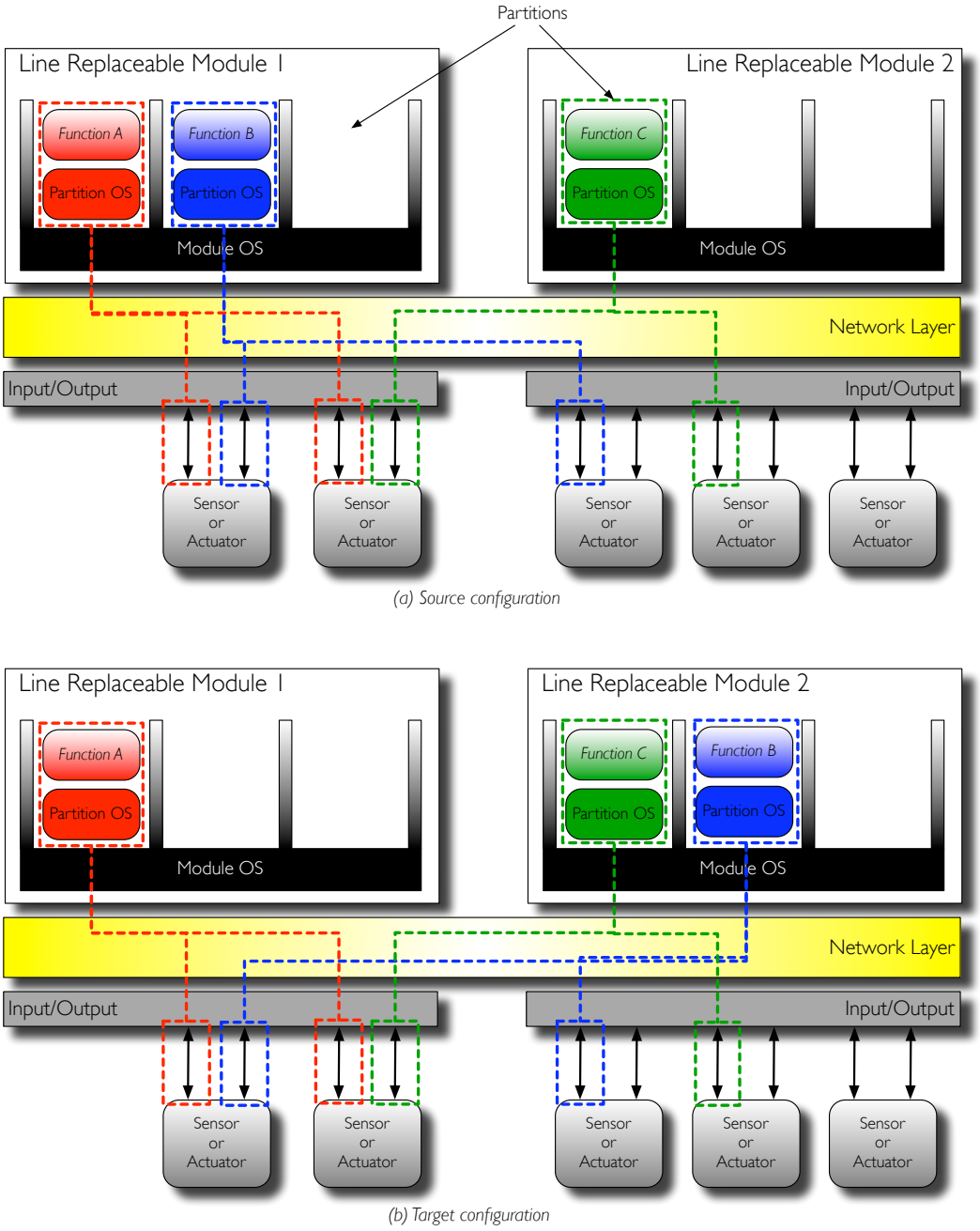


Figure 2.3: IMA reconfiguration process: function B is relocated from LRM 1 to LRM 2.

under a different target specification S_j (see Figure 2.3).

Strunk and colleagues propose a generic architecture for an IMS reconfiguration controller called SCRAM (Subsystem Control Reconfiguration Analysis and Management). SCRAM is used as a vehicle to introduce a formal framework for **assured reconfiguration** of embedded systems [Strunk and Knight 2004]. The authors express functional and state properties in set theory, whilst timing properties are expressed in Real-Time Logic (RTL) [Jahanian and Mok 1986]. Particular attention is given to the reconfiguration sequencing mechanism and the timing. An important assumption is that applications, and computing modules in general, have a *fail-stop* behaviour: the

application or computing module is required to operate correctly or stop and signal an error. The rationale is that such behaviour is much simpler to verify than complete functionality and thus, in the authors' view, presents an advantage in both development and certification.

Knight et al. [2003] relate assured reconfiguration to the concept of **survivability**. Having reviewed several definitions of survivability from different engineering domains (such as information systems, combat machines and telecommunications services), they provide a first formal definition of survivability: *a system is survivable if it complies with its survivability specification*. The apparent simplicity of the definition stems from the underlying complexity of the specification structure which they formally present using a Z-like notation [Woodcock and Davies 1996]. The link between reconfiguration and survivability is interesting for our research work as it cross-refers to our ideas concerning integrity and safety introduced in Section 2.1.1.

Stephenson et al. [2005; 2006] give a contribution to the IMS reconfiguration problem from the point of view of **product lines**. They propose a framework in which "the impact of the different possible configuration and reconfiguration schemes is assessed by instantiating them as staged product-line configuration processes".

Schrage and Vachtsevanos [1999]; Wills et al. [2000] introduce the Open Control Platform (OCP), a hierarchical software infrastructure for real-time, reconfigurable complex control systems. OCP has been successfully demonstrated on an Uninhabited Aerial Vehicle (UAV). An interesting achievement of this project, undertaken in collaboration with Boeing, is that *heterogeneous information* is merged by the reconfiguration mechanism, including aerodynamics and motor control dynamics and processed using innovative algorithms. Data fusion is an important feature of the framework proposed later in this thesis, although a different approach from OCP is adopted.

Arshad [2003] sheds light on the *planning phase*, a phase of the IMS reconfiguration which went unnoticed in the literature. Arshad highlights that the dynamic reconfiguration process can be divided into three phases, (a) sensing the need for reconfiguration, (b) planning it and (c) carrying it out. The author introduces a novel technique for carrying out the planning process using 'temporal planners', which compute the plan for dynamic reconfiguration under tight time and resource constraints. Arshad's work identifies a further dimension of the IMS dynamic reconfiguration problem that is critical to guarantee the integrity of the process.

Outside the context of this Ph.D. programme, Montano et al. [2010] proposed the Dynamically Reconfigurable Processing Module (DRPM), a framework for the dynamic reconfiguration of IMS for next-generation spacecraft which encompasses the dynamic reconfiguration of FPGA[4] modules which are part of the on-board network. Besides taking care of the allocation of functions on each LRM, the Reconfiguration Manager reconfigures the functionality of each processor at runtime. The technology, developed in collaboration with the European Space Agency, has been successfully demonstrated on real space flight hardware.

The problem of software execution *timing analysis* for IMS dynamic reconfiguration has been investigated by several authors. Starting with the two-level scheduling hierarchy architecture proposed by the ARINC standard for IMA, Lee et al. [2000]; Younis et al. [2000] first model IMA as a composition of multiple partition servers and channel servers; they then introduce a method to

[4]FPGA – Field-Programmable Gate Array.

provide schedules for both tasks and messages that provide for robust temporal partitioning.

Bate and Burns [2003] express their concerns on the adequacy of standard timing analysis for real IMA implementations. They propose extensions to standard timing techniques that account for the safety requirements that are characteristic of avionics.

Watkins et al. [2006] focus on the *resource allocation* aspect of the IMA reconfiguration problem. They propose an analytical method based on compositional reasoning which they call a "contract-based approach to the integration of modular systems". The method is implemented through the use of a set of tools that analyses the system, allocates the logical avionics architectures to the physical platform architecture, and subsequently generates the platform configuration parameters. This technology was implemented on the Boeing 787 aircraft as the Common Core System (CCS).

The general feeling resulting from the material presented so far is that the IMA dynamic reconfiguration process is assumed to be autonomous. In this regard, it is interesting to consider the following excerpt from the EUROCAE ED-79 standard:

> *Some failure conditions can be mitigated through* ***human interaction****. Recognizing that incorrect human interactions could exacerbate, rather than mitigate, the situation, it is essential to examine the* ***type and independence of the support provided*** *to ensure the correctness of such interaction. Support should be provided for both human recognition of the system or item failure condition and human action to mitigate the failure effects.*
>
> [EUROCAE 1996]

This passage refers specifically to real-time fault management processes. As previously seen, dynamic reconfiguration is a form of adaptation to unexpected events like faults, hence dynamic reconfiguration *is* a sophisticated form of fault management for next generation aircraft. As a consequence, the passage from EUROCAE ED-79 applies to the IMA dynamic reconfiguration process.

The extract from the EUROCAE ED-79 pivots around two main issues, (a) characterisation of the type of support that the human can provide to the system and (b) characterisation of the type of decision support that the system can provide the pilot during the real-time, fault management process.

To the best of our knowledge, to date no study has considered human involvement in the process. This thesis proposes a novel framework for IMA dynamic reconfiguration based on the Constraint Programming paradigm that is designed to actively involve the pilot in the process (Chapter 5). Moreover, the experimental data gathered provides a significant insight into pilot experience during IMA dynamic reconfiguration. The overall research makes a contribution in both the directions highlighted by EUROCAE ED-79:

(a) A series of human-computer interaction experiments is performed in order to specifically characterise pilot behaviour during IMA dynamic reconfiguration.

(b) A novel framework for a DSS for IMA dynamic reconfiguration is presented; the system is used to investigate the impact and effectiveness of different, purpose-made types of decision support information.

2.1.4 Conclusions

The limits of employing high-integrity technology alone to achieve system safety become progressively more apparent with the increasing complexity of modern SCMS. Investigating ways to involve the human in certain safety-critical processes, including SCMS dynamic reconfiguration, seems to be a promising solution.

The industrial trend of moving from legacy federated systems towards the Integrated Modular Systems (IMS) concept has a number of benefits but, at the same time, it generates new issues, especially in terms of safety. In this context, dynamic reconfiguration is one of the most challenging processes and represents an open problem for both the academic and industrial communities.

Modern standards for IMS, such as EUROCAE ED-79, acknowledge the importance of human interaction during fault management processes. They also prescribe an examination of the type of intervention and the type of support provided by the system to the operators. This thesis investigates the problem of human involvement during IMA dynamic reconfiguration.

2.2 Human Involvement

Human involvement during the dynamic reconfiguration process of SCMS is a key contribution of this thesis. This section starts by exploring how the interpretation of human error has recently changed within the human factors community. Subsequently, a number of theories of Decision Making (DM) are discussed. These theories are propaedeutic to the analysis of the behaviour of operators during SCMS dynamic reconfiguration decisions, which is the content of Chapter 4.

2.2.1 Old and New Views on Human Error

In his lectures on rhetoric, Nietzsche [1922][5] defines **metonymy** as "the substitution of cause and effect"[6]. The philosopher looks upon the fact that we say 'the rock is hard' as if it were something other than a judgement on our part. What we refer to as 'hard' is actually an effect, namely hardness, which is projected back onto the object as a cause of that effect [Klein 1997*b*].

It seems that in recent years the human factors community has witnessed a metonymy, in Nietzsche's terms, concerning the interpretation of *human error* which, as observed by Woods et al. [1994], was first regarded to as a *cause* of failure whilst now it is considered more as a *symptom* of failure.

Dekker made an extensive analysis of this subject, exploring its repercussions in different areas of the human factors domain, such as safety assessment, accident reconstruction, automation engineering, decision making in disaster relief and healthcare [Dekker 2003; 2002; 2000; Singer

[5] Translation into English: [Blair 1983].

[6] TheFreeDictionary [2010] defines metonymy in more general terms: "A figure of speech in which one word or phrase is substituted for another with which it is closely associated."

and Dekker 2000; Dekker 2001; Dekker et al. 2005]. The author acknowledges the existence of an *old* and a *new* human error view. In the old view, systems are considered safe, and it is human error that leads to most of the accidents. The new view is summarised in three main points:

- systems are basically not safe;
- people are central to creating safety;
- their "errors" are indications of irreconcilable goals and pressures farther up-stream.

In the new view, it is assumed that operators of safety-critical systems, like pilots, behave in a 'reasonably' rational way, depending on the characteristics of their operating environment. In this view, the categorisation of human error becomes less interesting than investigating the *rationale* for the pilot's course of action. Dekker [2001] maintains that "understanding how people make the systems they operate so successful, and probing the patterns by which their successes are defeated" is the only hope for real safety improvement.

This thesis provides a contribution to the field by investigating how pilots behave during IMA dynamic reconfiguration under different operating conditions and with different types of support from the system. Pilots are exposed to situations in which the system behaves in an unsafe way on purpose, explicitly breaking the assumptions of the old human error view. This allows the collection of empirical data that, in line with the latest ideas about human error, accounts for real situations and can effectively help practitioners in the development of SCMS.

Given that understanding the behaviour of pilots is crucial to improving safety in the new human error view, basic ideas from cognitive psychology, that apply to pilot decision making activity during IMA dynamic reconfiguration, are elaborated and implemented in our experiments (Chapters 4 and 6). Propaedeutic information concerning decision making theory is presented below.

2.2.2 Theories of Decision Making

By investigating the characteristics of SCMS dynamic reconfigurations and how humans reason in such circumstances, we aim at shaping an effective framework for decision support during the process of dynamic reconfiguration of SCMS. This section introduces concepts of Decision Making (DM) theories in preparation for the remainder of the thesis.

The field of DM is vast, including topics from cognitive psychology, computer science, human-computer interaction, automation engineering and cognitive engineering. Only the topics that are relevant for our analysis are introduced here. For a complete dissertation on DM, the reader should refer to Schraagen et al. [2008]; Smith and Kosslyn [2007].

Hammond [2000*a*] (page 53) states that the literature on judgement and DM is populated by theories that can be gathered under two general, rival meta-theories. The **coherence meta-theory** supports the achievement of logical or statistical *rationality* whilst the **correspondence meta-theory** strives for *empirical accuracy*. Falzer [2004] traces the roots of the former back to Peterson and Beach's Bayesian-inspired claim that humans function as intuitive statisticians [Peterson and Beach 1967] and studies by Hoffman et al. on regression models as representations

of clinical judgement [Hoffman 1960; Dawes 1986]. Instead correspondence, is traced back to Brunswik's probabilistic functionalism [Brunswik 1947; Gigerenzer et al. 1991] and Tversky and Kohler's work on *support theory* [Tversky and Koehler 1994].

The literature also contains another subdivision of DM theories which is structured in three groups as follows:

- **Normative** (or **prescriptive**) theories, which investigate analytically how decisions *should* be made.
- **Descriptive (analytical)** theories, which describe how people *should* and *can* make decisions (assuming they're actually not perfect statisticians).
- **Descriptive (naturalistic)** theories, which focus on how humans *actually* make decisions in real circumstances.

Before proceeding, it must be noted that a general, unifying theory of DM applicable to all situations has not yet been developed. The common logic underlying each theory is that decision-makers are 'rational'; however, each theory starts with different assumptions and is crafted to suit a particular set of circumstances [Zsambok et al. 2002].

The most influential theories available in the literature are briefly discussed below, highlighting ideas that are applicable to our work. At this point it is important to note that it is not the intention of this thesis to gain an insight into the brain mechanisms that characterise pilot decision behaviour during IMA dynamic reconfiguration. Hence, the focus is not on examining *all* the potential non-rational behaviours that can possibly emerge (an objective of dubious practicability anyway). Rather, this thesis aims at verifying empirically that the complexity of SCMS dynamic reconfiguration decisions is such that, *without effective decision support information*, operators are likely to incur the risk of making wrong safety-critical decisions. The information gathered is then used to shape an effective framework for decision support.

In this context, pilot brain mechanisms are considered as a "black box". Several ideas from cognitive psychology are used to make hypotheses which are then empirically investigated, but the focus is always on pilot behaviour rather than on the discussion of which mechanism actually determines it.

2.2.2.1 Normative theories

Taking the temporary risk of oversimplification, a **decision** can be brought down to a choice amongst two or more possibilities. In an ideal world, the decision maker uses a goal function to evaluate the *utility* associated with each possibility and chooses the best one. In such theoretical conditions the decision outcome is pretty much obvious since no room is left for uncertainty.

The behaviour of the decision maker in this type of ideal scenario was first studied in the 1950s (the phrase "decision theory" was first used by Lehmann [1950]), aiming to provide a framework for making the best possible decision in a given set of circumstances [Neumann et al. 1953; Edwards 1954].

These studies gave rise to a number of theories on decision making that are now referred to as **normative** (or **prescriptive**) since they investigate how decisions 'should' be made. In normative theories, the decision maker's evaluations of consequences are signified in terms of utilities, which are numbers that, at the time the human is faced with the decision, express the strength of like or dislike of the outcomes that might occur. The basic normative model is provided by **Expected Utility Theory** (EUT), which can be summarised as follows:

Definition 2.2.1 *The (subjective) expected utility E of an action is equal to the sum of the probabilities p of each possible outcome x_i multiplied by the utility u of that outcome:*

$$E = \sum p(x_i)u(x_i)$$

EUT and normative theories in general provide a good approximation of human behaviour, but only if the situation is simple and provides all the relevant information. People's judgements under conditions of uncertainty, risk and time pressure are not captured by the rules of mathematical probability theory; similarly, any formal theory that prescribes an ideal relation between values and utilities does not provide a realistic model. Real world human decisions, like those that an aircraft pilot faces during a real-time fault management procedure, happen in scenarios characterised by time pressure, risks and uncertainties. All these factors drastically influence the behaviour of the decision maker which drifts away from any prediction made through a normative model.

The literature contains evidence of several decision biases (not necessarily independent) that invalidate normative theories in real contexts including: *time pressure* [Janis and Mann 1977; Stokes et al. 1987; Orasanu and Fischer 1997; Orasanu 1997; Zsambok and Klein 1997; Klein 1997*a*; Klapproth 2008], decision *relational complexity* [Quesada et al. 2005; Halford and Wilson 1998; Cowan 2001], loss of *situation awareness* [Endsley and Strauch 1997], *mental misrepresentation* of the situation [Baxter et al. 2007], *emotions* [Luce et al. 1999; Mano 1999; Isen 2001; Fiedler 2001; Rahman 2009; Mosier and Fischer 2009], *stress* [Poulton 1976; Gillis 1993; Kontogiannis and Kossiavelou 1999; Kowalski-Trakofler et al. 2003], decision information *framing* [Payne et al. 1998], *complacency* towards the information provided [Billings et al. 1976; Parasuraman et al. 1993; Smith and Geddes 2003; Bustamante et al. 2009] and *negative attitude against automated decision support aids* [Lee and Moray 1992; Muir 1994; Riley 1996; Endsley and Kiris 1995; Endsley 1996; Parasuraman and Riley 1997; Dzindolet et al. 2003; Lee and See 2004].

It can be concluded that, because of the likelihood of SCMS reconfiguration decisions being biased by the factors mentioned above, *normative DM theories are not suitable to investigate human decision behaviour in our context*. Therefore, they are not referenced further in this thesis.

2.2.2.2 Descriptive (analytical) theories

A number of new DM theories that take into consideration the potential emergence of decision biases arose in the 1950s, after empirical studies started to reveal the inability of normative theories to capture human decision behaviour in real scenarios.

Simon [1955] introduces the concept of **bounded rationality**. The author suggests that our brain develops adaptive strategies that trade-off between, on the one hand, the cognitive effort of

searching for and processing information and, on the other hand, the choice of the best alternative. Such strategies are defined as **satisficing**, methods that aim at finding not necessarily the best of all alternatives, but one that is good enough to meet the desires of the decision maker.

Most important dimension theory [Slovic 1975] advocates that when humans are faced with an equal choice they choose the alternative that rates highest on the "most important dimension".

The fact that the perception of risk influences decision behaviour is well captured by **prospect theory** [Tversky and Kahneman 1992], which is one of the most influential descriptive analytic models dealing with risk and uncertainty [Smith and Kosslyn 2007]. In support of the robustness of the theory, more recently Trepel et al. [2005] outlined the possible neural basis of the components of prospect theory, surveying evidence from human imaging, lesion, and neuropharmacology studies, as well as animal neurophysiology studies.

According to the theory, human actions are determined by the mental representations of a situation, not directly by the situation itself. The first stage of the decision making process is comprehending the prospects ahead by framing the terms of the decision, which involves using heuristics to simplify the scenario and evaluate prospective gains and losses in relation to a **reference point** (or **anchor**). The anchor is the present situation, before the decision is made. Once the terms of decision are represented in memory and associated to values and weights, which are the subjective estimates of probabilities for the prospects under consideration, an expected-value is calculated for each prospect from all the relative values and weights.

Several aspects of these theories are relevant for our analysis, hence they are discussed in more detail later in the thesis. A more comprehensive review can be found in [Elliot 2005].

Payne et al. [1993] gathers all the descriptive theories under the meta-theory of **adaptive decision maker**: he envisions a cognitive system with a "toolbox" which includes several algorithms and heuristics to perform *complex* decisions in an efficient way. The decision maker chooses algorithms or heuristics in an adaptive manner, relying on strategies that fit the needs at the time and that capitalise on the specific structures of the current environment.

Elliot [2005] highlights that a number of studies on decision biases have been criticised for artful production of biases [Lopes 1992] and that these biases are reduced if the study includes contextual factors [Klein 1998]; furthermore the biases are less likely to emerge in experienced decision makers [Christensen-Szalanski and Beach 1984; Fraser et al. 1992; Gigerenzer 1987; Shanteau 1992; Smith and Kida 1991], a result which is taken into particular consideration in the conclusions of this research.

The theories mentioned in this section still incorporate some mechanistic, analytical processes (e.g. probability based assumptions), which are not representative of how the human brain actually works during real decisions [Tversky and Kahneman 1973]. However, they provide explanations to some observable phenomena which are relevant to this research. As a result, some of these theories are referenced later in the thesis.

2.2.2.3 Descriptive (naturalistic) theories

Descriptive (naturalistic) theories, which focus on how humans *actually* make real decisions, not on how humans *should* make them or how they would make them in protected environments, are

most suitable to this research. Originating in the 1970s, these theories reveal how human behaviour departs from the prescriptions of entirely rational choices specifically in risky and uncertain scenarios. At the present time, the Naturalistic Decision Making (NDM) framework is probably the most referenced set of descriptive theories in the literature. It emerged in the early 1990s, to study decisions taken in natural settings that take forms that are not easily replicated in the laboratory [Salthouse 1992; Klein et al. 1993; Klein 1993*b*].

NDM theories are suitable for investigating human decision behaviour during SCMS dynamic reconfiguration because these theories are specifically targeted to understanding how people use their experience to make decisions in complex, dynamic, real-time environments [Meso et al. 2002]. They explore methods used by *experts*, either working as individuals or in groups, to assess the characteristics of the current situation and make decisions.

The basis of NDM lies in the fact that human cognitive resources are limited in memory and computational power [Norman and Bobrow 1975]. It was argued that, in order to compensate for its limitations, the brain engages in the least amount of cognitive work it can get away with in face of tasks that exceed the available resources [Payne et al. 1993; Fiske 1993]. This phenomenon is also referred to as *cognitive misery* by some authors (e.g. Skitka et al. [1999]).

The remainder of this section provides a brief overview of the five most influential NDM models available in the literature. A more comprehensive review is given by Zsambok et al. [2002]. The argumentation is kept at a macroscopic level, as the aim is to provide background information for later chapters and at the same time, rule out those models that are evidently not applicable to the problem being investigated.

It is anticipated that, throughout the description of the DM models, the reader should pay attention to the role of *mental simulation* (the definition is given hereinafter). In fact, this process plays a crucial role in the characterisation of the decision support system proposed in Chapters 4 and 5.

Recognition-Primed Decision

The **Recognition-Primed Decision (RPD)** model [Klein 1989; 1993*a*] asserts that decision makers draw upon their experience to identify a situation as representative of, or analogous to, a particular class of problems. Recognition leads to the identification of a course of actions, which is then evaluated through a process of mental simulation.

Klein and Calderwood [1986] focus on dynamic situations, characterised by unfolding events and rapid changes over time. They observe that decision-makers do not usually react to the changing situation by first diagnosing an initial part of the event as Situation-A, the second as Situation-B and so on. Rather, they usually identify the initial situation and, as time goes by, they generate expectations on how it can possibly evolve. *Mental simulation* is critical for this operation.

The concept of **mental simulation** has been defined in several ways in the literature, both in terms of a process or the end-product of a process. Here mental simulation is described in terms of the *simulation theory* [Goldman 2002], which posits that humans gain insights into the plans, beliefs, and desires that motivate the actions of others by covertly simulating those same actions in themselves, without actually performing them. This theory is robust enough to have a

physiological counterpart [Hesslow 2002].

Noble's cognitive model

Noble [1993] focuses on the situation assessment portion of the decision making process and proposes a cognitive model, somewhat similar to RPD, in which each type of previously experienced problem (or decision) is treated as if it is stored in memory as a separate *reference problem*. Each reference problem embeds context, goals, solution methods and other forms of information relative to a specific type of problem.

A reference problem gets activated if its features match those of the new problem under examination. Several different reference problems can be activated with different strengths at the same time, depending on how much they 'match' with the new problem. Once the situation has been assessed, as with RPD, *mental simulation* and correspondences with the reference problem allow the generation of expectations about how the situation could possibly evolve.

Image Theory

Image Theory [Beach 1998] is vast and accounts for a large number of aspects of decision behaviour. Here only the portion of the theory that is required for this argumentation is briefly introduced.

In Image Theory, the features of the current problem (*stimulus situation*) are evaluated and if they match with other features stored in memory, then the situation is said to be *recognised*. If the features are not perfectly matching, but only resemble those stored in memory, then the situation is said to be *identified*. Recognition and identification allow the current situation to be *framed*; a frame is information stored in memory that allows meaning to be given to the current situation. Frames are used as a point of reference for each new situation and are updated as events unfold.

Once a frame is activated, associated *policies* are activated, too. Policies guide courses of actions (or plans). Plans are evaluated through a *compatibility test* which allows plans whose features are not suitable for the current situation to be dropped. Additionally, the *profitability test* is performed through a set of strategies that allow selection of the best option.

A critical operation for the decision maker is figuring out whether the plan under evaluation allows the pre-defined goals to be reached. This is done by *mental simulation*; if the simulation is not satisfactory, plans are changed.

Rasmussen's model

Rasmussen [1983] proposes a model for representing human performance at *skill-*, *rule-* and *knowledge-based* levels. His analysis focuses on the decision making activity of expert operators of complex automated systems like nuclear power plans.

The knowledge and experience that the operator has about the process/system determines the level of cognitive control that is exercised (skill-, rule- or knowledge-based) and as a consequence, the features and structure of the information flow between the human and the system.

Skill-based control is exercised by people with a higher level of expertise. Actions are generated in the subconscious by means of dynamic mental models of a familiar situation.

In rule-based control, actions are conscious and follow *rules*, which are procedures or subroutines stored in memory that define the behaviour for each situation. The boundary between skill and rule-based is not sharp; the decision maker could switch from one type of control to the other, depending on the familiarity with the task.

Finally, knowledge-based control is conscious decision making activity that is performed in situations in which the goals change dynamically or the knowledge is insufficient. In these situations humans construct plans and test their applicability through "thought experiments". If the test is not successful, the plans can be modified dynamically. Again, this process is akin to the *mental simulation* activity found in later stages of the decision process, as well as in the other models previously described.

Explanation-based model

Unlike previous models, the explanation-based model developed by Pennington and Hastie [1988] is not targeted at the control of automated processes, rather it is designed on studies of jury decisions. However, such decisions are complex and characterised by high uncertainty, hence the model can be included in this analysis with good reason.

In the explanation-based model, the decision maker builds a causal *story* that explains information previously received. The story, along with the information, determines the decisions. The rationale given by Pennington and Hastie is that constructing stories and asking 'why-type' questions helps establish relationships between facts; the more complete and consistent the story, the lower the uncertainty.

Once again, *mental simulation* is a critical process to construct stories, to discard inconsistent ones and to establish casual chains of relationships.

2.2.3 Conclusions

In recent years the human factors community has witnessed a metonymy as to what concerns the interpretation of human error, which was first regarded as a cause of failure (*old human error view*), whilst now it is considered more as a symptom of failure (*new human error view*). In this light, a good way forward to improve the safety of next-generation SCMS is getting an insight into the rationale for human decisions during system operation.

Five well established NDM cognitive models that are applicable to the decision making activity of pilots during IMA dynamic reconfiguration have been described. According to each model, *mental simulation* plays a crucial role, allowing the establishment of a casual link between the features of the decision scenario (input) and the consequences of an action/decision (output) by the human. Indeed, this concept was clear to the German philosopher and classical philologist Nietzsche, who pointed out that few things make us as anxious as not having a cause for things that go wrong [Nietzsche 1888]. *Explanations* of the causes of unexpected events and *justifications* for the courses of action to be taken are critically important for the human decision maker;

they allow causes to be linked to effects, situation awareness to be built and informed choices to be made. However, generating justifications for the evaluation of each decision alternative is a difficult problem for a human being; fragmented or incorrect information, cognitive limitations, stress, risks and other decision biases make the problem unfeasible in demanding situations.

During safety-critical decisions for the control of modern SCMS, would the availability of automatically generated decision support information, justifying each decision alternative, allow a more informed choice and improve human decision making performance and accuracy? Should this be the case, such a decision support approach would undoubtedly make the overall SCMS safer.

Our hypothesis, which is extended and refined in the next chapter, is that in the context of the problem examined here, the human decision making performance and accuracy could be improved by a *decision support aid* specifically designed to favour mental simulation.

2.3 Decision Support Systems

As previously seen in this chapter, unplanned dynamic reconfiguration of IMA is tangled up in the real-time fault management process. The position of this thesis is that at the occurrence of a fault that requires reconfiguration, a timely and effective exchange of information should be established between the aircraft and the pilot, with the system starting the "communication".

Warnings that follow an unexpected event have three main purposes, (a) to *alert* the pilot that something is wrong, (b) to *report* what is wrong, (c) to *guide* the pilot in what to do [Martensson and Singer 1999]. Studies of aviation psychology document that pilots lament that current warning systems in commercial aircraft are not effective in accomplishing these three objectives. For instance, pilots report that fault management technology delivers too much data, particularly all kinds of secondary and tertiary failures, with no logical order; furthermore, primary faults, which are the root causes, are rarely highlighted [Martensson and Singer 1999; Singer and Dekker 2000].

The approach taken in this thesis, to structure the exchange of information with pilots during IMA dynamic reconfiguration, exploits and extends current research in the domain of **Decision Support Systems (DSS)** engineering. The interpretation of the term DSS adopted here is 'a class of computer-based information systems that support human decision-making activities'. Nowadays, DSS are effectively used to support decision makers in diverse domains, such as railway [Dadashi et al. 2011], medicine [Shortliffe et al. 1979], retail industry [Häubl and Trifts 2000], nuclear emergencies [Ehrhardt et al. 1993; Vamanu et al. 2004], national security [NASA and University Of California San Diego 2005] and military tactics [Hutchins, Morrison and Kelly 1996].

DSS have also appeared in the aviation field, e.g. [Sarter and Schroeder 2001; Painter et al. 1997]. A notable example from the fault management field is the Hazard Monitor [Bass et al. 1997], which tracks user interactions with the system and tries to match them against information contained in a knowledge-base as a way of doing plan recognition. The objective of Hazard Monitor is to make suggestions about what the user should consider doing next.

Recommender systems (RS) are a specific type of DSS that "aim to relieve information and

interaction overload over users by applying intelligent filtering techniques that ultimately present to the user the most relevant and attractive information, people, or communities" [ACM TIST 2010]. At present, RS are extensively used on the web by retailers of music, videos, books, news, film, images, and other items [Jannach 2010]. In a typical scenario, the RS uses a *user profile* to predict the 'rating' that a decision maker would give to an item he has not yet considered. Intuitively, this problem has certain similarities with the problem faced by pilots during an IMA dynamic reconfiguration: (a) the search space is too big to be explored without the support of computer-based intelligent filtering; (b) the decision time is limited; and, (c) a solution (i.e. an avionics configuration) is generated by the system on the basis of the present circumstances.

This thesis makes a contribution to the field by presenting a novel recommendation system framework for IMS dynamic reconfiguration, based on the Constraint Programming paradigm and a number of domain-dependent heuristics. Empirical data about its effect on pilot decision making performance is collected and discussed.

2.3.1 The Generic Decision Support Problem

The first papers to mark the emergence of RS as an independent area of research date back to the 1990s [Resnick et al. 1994; Hill et al. 1995; Shardanand and Maes 1995]. More recently, Adomavicius and Tuzhilin [2005] defined the recommendation problem as follows:

> Let C be the set of all users and let S be the set of all possible items that can be recommended, such as books, movies, or restaurants. The space S of possible items can be very large, ranging in hundreds of thousands or even millions of items in some applications, such as recommending books or CDs. Similarly, the user space can also be very large-millions in some cases. Let u be a utility function that measures the usefulness of item s to user c, i.e., $u : C \times S \rightarrow R$, where R is a totally ordered set (e.g., non-negative integers or real numbers within a certain range). Then, for each user $c \in C$, we want to choose such item $s' \in S$ that maximizes the user's utility. More formally:

$$\forall c \in C, s'_c = \arg\max_{s \in S} u(c, s) \tag{2.1}$$

The major difficulty in recommendation problems is that u is usually ill-defined and, when a clear definition exists, it is usually available only for a subset of the items in $C \times S$, hence it must be extrapolated to all of them. In most cases, u is extrapolated through heuristics or statistical functions, such as means.

As discussed in Chapter 5, the utility of each IMS configuration is not easy to define and summarise in a single number. Furthermore, IMS configurations are dynamically generated on the basis of the current operating conditions, hence static utilities extrapolated on the set of applicable configurations are scarcely effective.

Decision support measures are used to determine how well an RS predicts high-relevance items. Examples of measures are classical IR measures of precision (the percentage of correctly

predicted high ratings among those that were predicted to be high by the RS), F-measures (a harmonic mean of precision and recall), and Receiver Operating Characteristic (ROC) measures, demonstrating the trade-off between true positive and false positive rates [Herlocker et al. 1999].

Classic decision support measures do not seem sufficient to characterise the effectiveness of a SCMS reconfiguration recommendation, e.g. precision is only one dimension of relevance; decision time, the pilot's situation awareness and cognitive demand are some of the other dimensions of decision support measurement that are significant for the SCMS dynamic reconfiguration problem which are not easily captured by classical measures.

Dale and Reiter [1995] refer to the classic measures of evaluation as *glass box* methods, as opposed to *black box* evaluations which look at the performance of the system as a whole. Black box methods have been used to evaluate well-known interactive systems like 'KNIGHT' [Lester and Porter 1997] and 'AlethGen' [Coch 1996*b*;*a*].

This thesis makes a contribution to the field of research by: (a) investigating pilot reaction to different types of recommendations and (b) investigating the use of black box measures to define the utility of recommendations in SCMS reconfiguration problems. To this purpose, a number of metrics of user experience that are uncommon in the RS domain but are typical of aviation psychology studies are used to evaluate the performance of the system as a whole, e.g. situation awareness, decision time, perceived cognitive demand and eye movement analysis.

2.3.2 User Profiling

The definition of the *user profile*, which is part of the formulation of the recommendation problem, is a subject that receives a lot of attention in the RS engineering field, because the success of the system depends to a large extent on the ability of designers to capture user interests and of the system to satisfy them.

With reference to Equation 2.1, Adomavicius and Tuzhilin [2005] maintain that "each element of the user space C can be defined with a profile that includes various user characteristics, such as age, gender, income, marital status, etc. In the simplest case, the user profile contains a single element, which is the User ID".

Ramezani et al. [2008] distinguish between (a) **persistent user models**, which contain user interests and preferences deduced from user inputs accumulated over time, and (b) **ephemeral user models**, which contain current user interests (or intentions) based solely on inputs from the current user session.

A review of the literature shows that the two approaches are implemented in different ways by means of a plethora of algorithms. Syskill & Webert [Pazzani and Muramatsu 1996], an RS that suggests what web pages might interest a user, constructs a user profile from the user's ratings of pages and uses this profile to suggest other pages accessible from the index page. In VITA [Felfernig et al. 2007], an RS for financial decisions, each customer registered with the system has a profile which consists of personal information, previous recommender session data, purchased products and other financial service providers. In FAB [Balabanović and Shoham 1997], a hybrid RS for the Web, *relevance feedback* is used to generate user profiles: weights are associated with the words of web pages previously selected by the user; the weights of the words contained in all

the pages selected by the user are used to build a profile. This approach is informally referred to as the "bag of words" [Mooney and Roy 2000; Bilgic and Mooney 2005].

The majority of current approaches to generating user profiles aims at capturing the behavioural aspect of the decision making and the analytical capabilities of decision makers. However, in the 1980s, Young [1983] envisioned DSS technology to be "executive mind-support systems" that seek to establish a symbiosis of human mind and computer. Chen and Lee [2003] notice that, up to then, this grand vision had yet to become a reality. To the best of our knowledge, this objective has not yet been accomplished; the cognitive aspect of decision support has received relatively little attention, despite its evident criticality in the design of any form of decision support.

This thesis makes a step in the direction indicated by Young. Instead of deducing user's interest and preferences from user inputs accumulated over time, the approach adopted is to characterise an user (pilot) profile starting from basic assumptions from aviation psychology and testing them empirically using a novel RS, purpose-made for this PhD programme. In many interactive systems the user model is not explictly defined but it is effectively hard wired into the system [Dale and Reiter 1995]; this is the approach followed here. A typical pilot's profile, relevant to the problem of decision making during SCMS dynamic reconfiguration, is elaborated in Chapter 4; this profile is used as a reference point for the design of the RS algorithms presented in Chapter 5 and its effectiveness is assessed in Chapter 6, where a series of human-computer interaction experiments is presented.

2.3.3 Recommendation Approaches

User interaction

The *type of interaction* that is established with the user allows the RS to be classified as follows:

- **Conversational recommenders**: an interactive dialogue is established with the user (e.g. FindMe [Burke et al. 1997]). The system asks for feedback or answers to questions. **Question/answer** systems are a form of conversational RS in which the system questions the user and uses the responses to formulate a recommendation (e.g. INCA [Langley 1999]). Another variation of conversational RS are **candidate/critique** systems, which display a basic set of recommendations to a user and solicit feedback (e.g. ATA [Linden et al. 1997]).

- **Single-shot recommenders**: the user is not provided with any feedback; each interaction produces an independent recommendation (e.g. FAB [Balabanović and Shoham 1997]).

Conversational recommenders have the advantage that the user can "drive" the construction of the final recommendation by interacting with the system in a sort of refinement process. On the other hand, the interaction monopolises the attention of the decision maker for a prolonged time.

With reference to the IMA dynamic reconfiguration problem, the pilot time budget to complete the reconfiguration process and the cognitive resources availability vary depending on the phase of the flight and the operating scenario in general. A prolonged interaction would be unfeasible in the majority of situations, such as after a fault. For this reason, this thesis focuses on single-shot

recommendation only; however, the RS framework proposed in Chapter 5 is general enough to be extended to allow more interaction with the pilot during the search for an applicable configuration. This allows the application of the technology introduced in this thesis to problems with a similar structure to the IMA dynamic reconfiguration problem but with more relaxed decision time budgets.

Selection versus configuration

RS are designed to help the decision maker to solve one of the following two combinatorial, complex problems: selection or configuration. In a **selection problem** (e.g. Granston and Holler [2001]), the decision maker chooses from many options in a problem whose search space size is too large for them to readily scan and to compare all the options. In a **configuration problem** (e.g. Inakoshi et al. [2001]), the user determines the "best" ways to combine sets of components or attributes of an object.

A configuration problem requires the decision maker to have knowledge about the interaction of the components or attributes that are combined in the final product. Making a parallel with the IMA dynamic reconfiguration problem, structuring the interaction in the form of a configuration problem requires pilots to "construct" an avionics configuration in real-time, through the combination of its components or attributes. On the contrary, in a selection problem the pilot would need to select a single configuration from a reduced set of configurations suggested by the system.

The narrow time budget available to complete a reconfiguration and the deep, pervasive knowledge of the avionics that pilots would need to construct safe and effective configurations in real-time make this type of interaction hardly applicable to our problem.

The approach taken in this thesis is to structure the interaction between pilot and system during IMA dynamic reconfiguration in the form of a selection problem.

Knowledge base

The type of knowledge base used by RS to produce recommendations divides them into four main groups: (a) collaborative (b) content-based (c) knowledge-based, and (d) hybrid systems.

In **collaborative recommendation** systems the knowledge base is build up incrementally, merging collaborative opinion profiles, demographic profiles and user opinions. The most well-known approach is *nearest neighbour*. Another variant is the *popular item* method, in which the collaborative information is processed as features associated with items, rather than with users. Furthermore, some researchers have proposed model-based methods to compress the collaborative opinion data, including clustering, singular value decomposition and others [Resnick et al. 1994; Sarwar et al. 2001]. An example of collaborative RS is Ringo [Shardanand and Maes 1995], a music recommender in which users express their musical preferences by rating various artists and albums and get suggestions of groups and recordings that others with similar preferences also like.

Felfernig and Burke [2008] classify **content-based recommendation** as "a pure classification task in the machine learning sense". The RS produces a classification rule for each user on the basis of the user's ratings and the attributes of each item. This mechanism allows the classification of

each item as likely to be interesting or not. Contrary to collaborative systems, no social knowledge is used. A well known example of this type of RS is NewsDude [Billsus and Pazzani 1999], a system that suggests news stories the user might like to read.

The literature identifies all recommendations that rely on knowledge sources other than those of collaborative and content-based approaches as **knowledge-based recommendations**. Bilgic and Mooney [2005] describe them as "something of an accident of history". An example of these systems is the restaurant recommender Entree [Burke and Hammond 1996], which recommends restaurants in a new city similar to restaurants the user knows and likes. Whilst the system allows the user to navigate the search space by stating the preferences with respect to a given restaurant (i.e. redefinition of search criteria), the user profile is not updated at run-time.

There are two well-known variants of knowledge-based RS, (a) case-based and (b) constraint-based RS. **Case-based recommendation** systems treat recommendation primarily as a similarity-assessment problem. A database of well-known cases is used to make comparisons with each item in the search space; a matching function calculates the likelihood of user interest for each item. Ginty and Smyth [2002] illustrate this process neatly on a real RS for the retail industry.

Constraint-based recommendations are constructed by explicitly defining constraints over the search space that drive the quest for the right item to recommend. Examples of this technology are provided by Boutilier et al. [1997], Felfernig et al. [2007] and Felfernig and Burke [2008].

Finally, **hybrid recommendation** systems use components or logic from a mixture of the approaches mentioned so far. For instance, Inakoshi et al. [2001] propose a RS that combines constraint-based and case-based reasoning; Zanker [2008] investigates the combination of constraint-based reasoning and collaborative filtering.

This thesis extends current research in constraint-based recommendations by proposing a novel framework for recommendation construction that employs Explanation-based Constraint Programming and a number of domain-dependent heuristics (e.g. ranking avionics configuration by means of a bespoke Weighted Sum Model). The technology proposed is discussed in detail in Chapter 5.

2.3.4 Information Framing

The decision support information generated by the framework proposed in this thesis contains descriptive data about the unexpected event that triggered the reconfiguration (for example, the fault description) and other information about one or more configuration recommendations (such as the reason why the configuration in question is being proposed, the consequences of applying the configuration suggested by the system).

This type of information is complex, structured and contains causal relationships. Intuitively, alarms and simple warning messages are not sufficient to convey it effectively, hence the system must provide the pilot with textual or graphical information without overloading him or her.

Natural Language Generation (NLG) theory structures the process of construction of textual information in the following five phases:

1. *Content Determination*: what information should be communicated;

2. *Information Structuring*: how chunks of information are grouped;

3. *Lexicalisation*: deciding the specific words that should be used;
4. *Aggregation*: deciding how the structures created before should be mapped onto linguistic structures (sentences, paragraphs);
5. *Linguistic Realisation*: converting abstract representations of sentences into real text.

Through the five phases just mentioned, state-of-the-art, sophisticated NLG systems are able to generate 'readable' and 'appropriate' textual information for heterogeneous contexts. For example, 'FOG' [Goldberg et al. 1994] generates textual weather forecasts from numerical weather simulations produced by a supercomputer and annotated by a human. Amongst the other achievements of FOG, the system is able to decide how detailed the information it provides should be, depending on the weather information being processed.

In the software engineering domain, 'ModelExplainer' [Lavoie et al. 1997] describes models of object-oriented software in textual format. One complexity in this system is the capability of aggregating information in order to produce sentences which contain several clauses. Reiter et al. [1995] propose 'IDAS', a NLG system that produces hypertext help information for operators of complex machines. It uses data stored in a knowledge base that describes the machine. 'AlethGen' [Coch 1996*b*] focuses on the high quality of the multi-paragraph text generated and the data-driven planning approach, which allows production of an extensive set of different text structures.

As previously mentioned, alarms and warning messages are not sufficient to provide pilots with enough information during IMA dynamic reconfiguration. On the other hand, the text generated should have a pre-defined structure that makes the information conveyed readily understood; in consequence, high-level, human-like text could become counterproductive for the IMA reconfiguration problem. As a result, the approach of this thesis is not to focus on sophisticated techniques for lexicalisation, aggregation and linguistic realisation, but to focus the attention mainly on content determination and information structuring. An approach to structured decision support information for avionics reconfiguration decisions based on the *schema* paradigm is introduced in Chapter 5.

2.3.5 Explanations

Several researchers have recognised the importance of providing users with *explanations* of recommendations, in order to improve both their trust in the system and the effectiveness of recommendations. Explanations help users to either detect or make an estimate of the likelihood of errors in the recommendation. Some RS provide explanations for their suggestions in the form of similar items the user has rated highly in the past (e.g. Amazon) or keywords describing the item that caused it to be recommended.

Previous work in the DSS field has proved the effectiveness of explanations in improving human's decision performance. MYCIN [Buchanan and Shortliffe 1984] is a fitting example of a DSS of this type from the medical domain. The system is designed to identify bacteria causing severe infections and to recommend antibiotics, with the dosage adjusted for the patient's body weight. Explanations are used to make recommendations transparent to the user; in fact, the user can ask both *why* the system arrived at a conclusion and *how much it knows* about the subject.

Matsumoto and Sakaguchi [1992] propose a knowledge-based RS that uses explanations to support operators of power plants during critical activities. Regarding this thesis, the relevance of this work is in the effective exploitation of explanatory facilities to progress the safety of decision making problems.

Bresina and Morris [2006] propose a constraint-based approach to automatically provide explanations and recommendations for temporal inconsistencies within the context of planning. Even though the recommendations provided by this framework are specifically designed for temporal inconsistencies, the way the information is codified into constraints and processed is interesting for this thesis.

Lester and Porter [1997] present a seven-year project in the field of explanation generation that led to the development of 'KNIGHT', a robust explanation system that constructs multi-sentential and multi-paragraph explanations from the large-scale knowledge base in the domain of botanical anatomy, physiology and development. KNIGHT uses a semantically-rich, large-scale knowledge base to generate explanations; the small size of the knowledge base required by the SCMS dynamic reconfiguration problem, allows the structuring of the data in the form of a constraint network and the generation of explanations by means of bespoke constraint programming algorithms and heuristics (Chapter 5).

Bélanger and Martel [2005] introduce an advisor tool to assist military staff in managing events and related courses of actions (COAs), as well as prioritising these COAs according to different evaluation criteria. The system is equipped with an automated generator of explanations for the ranking proposed to the operator which is based on a Multi Criterion Aggregation Procedure (MCAP) technique. Multiple criteria are also used to evaluate decision options by the system proposed in this thesis but a novel algorithm based on Weighted-Sum Model has been implemented (Chapter 5).

Other examples and theories on the effectiveness of explanations in RS are provided by Gregor and Yu [2002], Miller and Larson [1992] and Horvitz et al. [1988].

Despite the successful experience with explanatory facilities in the expert systems domain in general, Bilgic and Mooney [2005] comment that, at least in the RS domain, very few studies provide a systematic analysis of explanation generation methods for recommenders which takes into account the *cognitive aspect* of the user.

A positive example in this direction is provided by Herlocker et al. [2000]. The authors present a study in which a cognitive model of the typical user is constructed; then the behaviour of 210 participants who interact with the 'MovieLens' web-based video recommender [Dahlen et al. 1998] is investigated. The participants are supported by computer-generated explanations of recommendations. The result is that the user experience was improved for 86% of participants. The authors also show that certain *styles* of explanation increase the likelihood that the user will follow the recommendation. This thesis takes into account the problem of framing the explanations provided to the pilots from a human-computer interaction point of view in Chapter 4 and the conclusions reached are empirically assessed in Chapter 6.

Bilgic and Mooney [2005] take into account the cognitive aspect of generating explanations of recommendations. In their study, three methods for explaining recommendations of content-based

and/or collaborative systems are compared. The authors show experimentally what terms each method uses to improve the user's estimated quality of an item.

The same benefits provided by the explanations in the DSS mentioned to this point could be obtained in the context of IMA dynamic reconfiguration. There are considerable differences regarding the user profile and the operating context. During avionics reconfiguration, the pilot has a short time budget available to make a safety-critical decision; stress, frustration and cognitive workload are all likely to emerge and potential errors can lead to catastrophic consequences. These are only a few, perhaps only the most obvious, conditions of interaction between a self-explanatory DSS and the human that differentiate the studies previously mentioned and this work on IMA dynamic reconfiguration.

This thesis extends current research in the domain of DSS and RS by exploring the effect of explanations of recommendations during IMA dynamic reconfiguration, a safety-critical scenario which is quite unusual in the recommender systems literature. This work aims at providing a contribution by giving an insight into the cognitive aspects of the interaction between a self-explanatory DSS and an aircraft pilot. Previous results from Herlocker et al. on the effectiveness of explanations are used as a starting point for the investigations.

2.3.6 From explanations to persuasion

Real-time fault management processes on-board modern aircraft are built around intricate statistical models. Deciding what to explain and how to do so in order to provide *effective* support to the pilot during dynamic reconfiguration decisions, is a complex task, which follows the preliminary problem of extracting the relevant information from the fault management logic of the system. In this regard, Bilgic and Mooney [2005] make an interesting observation from the generic point of view of recommender systems engineering:

> "The effectiveness of an explanation system can be measured using two fundamentally different approaches: the *promotion approach* and the *satisfaction approach*. For the promotion approach, the best explanation is the one that is most successful at convincing the user to adopt an item. For the satisfaction approach, the best explanation is the one that lets the users assess the quality of the item the best. Unfortunately, there is little existing research on explaining recommender systems."

This is an extremely important observation for this thesis. Unlike on-line shopping assistants, restaurant recommenders or other similar RS, a DSS for IMA dynamic reconfiguration should put the pilot in the position of being able to make an *informed* and *unbiased* choice, i.e. choose the best configuration with respect to the current operating conditions from those that are applicable. However, *persuasive* explanations, as suggested by Bilgic and Mooney, could impair the impartiality of pilot decisions. This could be an attribute of quality for certain on-line shopping assistants, but would be dramatically counterproductive for the avionics reconfiguration problem, as it could undermine the safety of the process itself.

The ability of self-explanatory RS to persuade the user towards erroneous choices should not be underestimated. In a study with eighty participants, on a full mission simulation in the NASA

Ames Advanced Concepts Flight Simulator, Skitka et al. [1999] demonstrate that an RS for critical flight decisions can persuade pilots towards wrong choices and lead to both commission and omission errors. Evidence of the same issue in a slightly different context was previously provided by Mosier and Skitka [1996].

The effect of computer-generated persuasive information on human decision behaviour has been also investigated in the domain of Natural Language Processing [Guerini et al. 2003; Mazzotta et al. 2007; Andrews 2008].

As the potential emergence of persuasion is most likely to hinder the reliability of IMA reconfiguration decisions, we investigate whether the explanations provided by the decision support framework presented in Chapter 5 have the by-product effect of persuading pilots to select the wrong avionics configurations or if they actually allow a more informed and impartial choice in most cases. The subjects of persuasion and complacency are discussed in more detail in Chapter 4.

2.3.7 Conclusions

Decision Support System (DSS) technology is effectively employed in several industrial domains, encompassing safety-critical applications. Recommender systems (RS), a specific class of DSS, are employed in decision making problems that have similar characteristics to the SCMS dynamic reconfiguration problem. The same benefits obtained in other fields seem to be achievable for SCMS dynamic reconfiguration, provided that the specific characteristics of this problem are taken into account (e.g. safety-critical context, limited decision time budget).

The definition of a user profile is critical for the effectiveness of a DSS. The majority of studies in the RS domain aim at capturing the behavioural aspect of the decision making process. Given the safety-critical context, which distinguishes SCMS dynamic reconfiguration from typical problems considered by the mainstream literature in the RS domain, further research seems necessary to characterise the pilot profile, taking into account the cognitive aspect of the problem.

The type of interaction between the system and the operator allows classification of RS as either ‘conversational’ or ‘single-shot’. Furthermore, the decision making process can be structured as a ‘selection’ or as a ‘configuration’ problem. The short decision time budget and the complexity of modern SCMS like those on aircraft narrow the possibilities available. It seems reasonable to structure the problem as a ‘selection’ problem and, consequently, the interaction as ‘single-shot’.

Alarms and warning messages are not sufficient to convey the information required by pilots during IMA dynamic reconfiguration. State-of-the-art Natural Language Generation (NLG) technology allows the generation of high-level, structured dialogues. However, the text generated by a DSS for avionics reconfiguration should have a pre-defined structure, in order to facilitate a ready understanding in critical situations, as the text generated by sophisticated natural language algorithms could become counterproductive in those circumstances. Moreover, further research is required to understand how to determine and structure the content of decision support information in a way that effectively helps pilots during reconfiguration.

Finally, explanations of recommendations have proven to improve user accuracy during decisions made with the support of an automated system. The possibility of achieving similar bene-

fits during IMA dynamic reconfiguration will be investigated. However, given the safety-critical context, the potential by-product effect of persuasion towards the wrong recommendation, resulting from the exceptional trust of the user in the DSS, clearly requires investigation.

2.4 On Mental Constructs and Measurement Methods

This thesis makes use of mental constructs and principles that result from the past three decades of research in Cognitive Psychology and Human Factors/Ergonomics (HF/E). Examples of these constructs are mental workload (ML), situation awareness (SA) and trust in the automation (TR), which are all defined and discussed below.

These constructs are used to put forward claims about pilot behaviour during avionics dynamic reconfiguration decisions. A number of experiments are then designed in order to assess the claims, and pilot responses are measured using well-established HF/E measurement methods, e.g. eye movement analysis, decision accuracy, decision time.

Recently, Dekker and Hollnagel [2004] have criticised the use and relevance of some of these constructs (i.e. SA, MW and TR) describing them as "folk models", lacking strong empirical foundations and scientific status. The authors raise three different points of criticism:

- *They explain complex mechanisms by means of substitution instead of decomposition*: the explanation of complex behaviours is made by referring to another phenomenon or construct that itself is in equal need of explanation. For instance, the authors point out that the literature equates complacency with boredom [Wiener 1988]; overconfidence [Stokes and Kite 1994]; contentment [Campbell and Bagshaw 2002]; unwarranted faith [O'Hare et al. 1992]; over-reliance [Kern and Kern 1998]; a low index of suspicion [Wiener 1988] and self-satisfaction [Parasuraman et al. 1993].

- *They cannot be falsified*: linking back to Popper [1972], Dekker and Hollnagel argue that "folk models" are under-specified, hence difficult to criticise and falsify;

- *They tend to rely on over-generalisation*: both ill-definition and immunity to falsification contribute to the over-generalisation of the constructs in question.

Parasuraman et al. [2008] contrast the position of Dekker and Hollnagel on each of the three points, gathering literature and examples of the application of the cognitive constructs and principles in question from the past three decades. Four major conclusions from Parasuraman et al. concerning SA, MW and TR are: (a) the three constructs have all been linked to information-processing or other psychological processes; (b) a number of studies have identified the brain mechanisms underlying some of them; (c) they have been modelled computationally; (d) SA, MW and TR have proven to be highly valuable for understanding and predicting human-system performance.

Dekker and Hollnagel [2004] also express concerns about measurement methods that have been typically employed in HCI studies for a number of years. As shown in Figure 2.4, in the authors' view some methods require little effort but their theoretical basis is poor (e.g. keyboard

interaction); other methods have strong theoretical foundations but are difficult to apply (e.g. eye movement analysis).

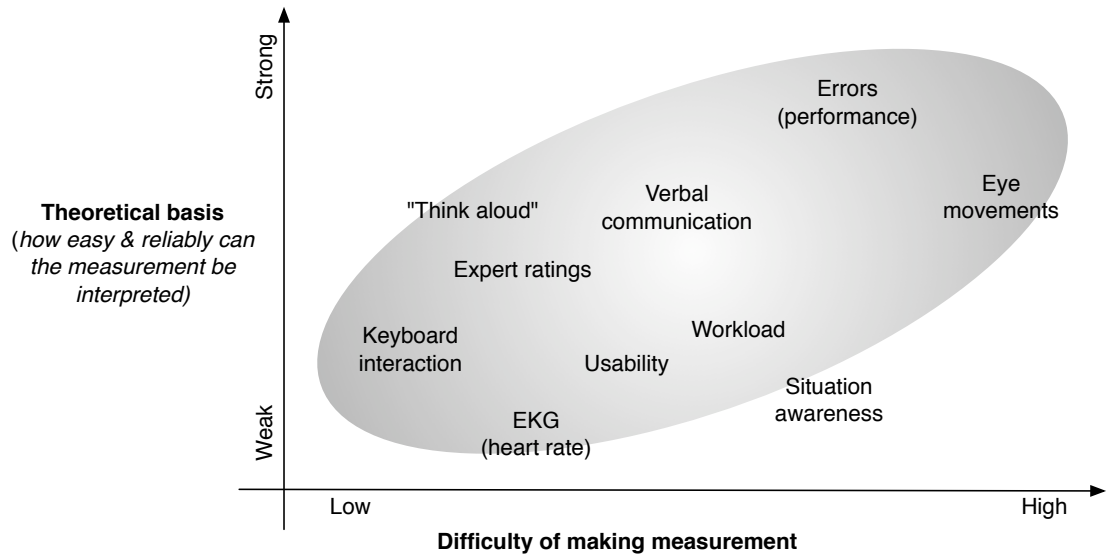


Figure 2.4: Meaning of measurements (adapted from Dekker and Hollnagel [2004]).

Following these arguments, although this thesis uses "modern" mental constructs like SA, MW and TR to make predictions about pilot behaviour during SCMS reconfiguration decisions, particular attention is paid to empirically verifying the predictions through measurement methods that have a strong theoretical foundation, such as eye movement and errors (performance), as suggested by Dekker and Hollnagel [2004]. The effectiveness of the multiple-metrics validation method we propose (Chapter 6) is advocated by the coherence amongst the heterogeneous results obtained.

As already mentioned in Section 2.2.2, the aim of this thesis is not to draw conclusions about the details of the brain mechanisms underlying certain pilot behaviour during SCMS dynamic reconfiguration. Instead, a number of cognitive constructs and principles are used to make *claims* about pilot behaviour. The conclusions drawn from our experimental results are only intended to devise effective decision support technology whose positive effect on pilot behaviour is observable (i.e. measurable).

2.5 Chapter Summary

This chapter covered the following topics:

- *State-of-the-art technology for the dynamic reconfiguration* of Safety-Critical Manned Systems from different engineering domains was reviewed, with a focus on aircraft avionics reconfiguration. The material presented casts light on the complexity of the process and the heterogeneous problems it encompasses.

- Particular attention was paid to the problem of *human involvement* in the dynamic reconfiguration process; several theories of naturalistic decision making were discussed, highlighting the critical role played by mental simulation during decisions of the type studied in this thesis. The arguments expounded suggest that providing decision support designed to favour mental simulation could be an effective approach to improve decision accuracy and performance of avionics dynamic reconfiguration decisions.

- *State-of-the-art decision support system (DSS) technology* was reviewed and basic conclusions concerning the design of a DSS for avionics dynamic reconfiguration were drawn; this allows the research performed in this PhD programme to be positioned in the DSS domain. The general inferences made in this chapter strongly influence the design of the technology proposed in the remainder of this document.

- *Recent controversies* in the domain of Human Factors/Ergonomics regarding the application of mental constructs and measurement methods to empirical studies have been discussed; although this thesis uses "modern" mental constructs (e.g. situation awareness and mental workload) to make predictions about pilot behaviour during avionics reconfiguration decisions, particular attention is paid to empirically verifying the predictions through measurement methods that have a strong theoretical foundation (e.g. eye movement analysis).

Chapter 3

Reconfiguration and Automation

"As machines become more and more efficient and perfect,
so it will become clear that imperfection is the greatness of man."
— Ernst Fischer

The previous chapter reported that the majority of current research focuses on techniques for autonomous reconfiguration of IMA (Section 2.1.3). This chapter collects material to question this assumption, setting up the context for and justifying a novel framework for IMA dynamic reconfiguration which accounts for pilot involvement.

The analysis put forward in this chapter is enriched by the description of a number of accidents from the aviation domain; their discussion is used to infer a set of drawbacks of both highly autonomous and insufficiently autonomous solutions for the avionics dynamic reconfiguration process. Two mental constructs, situation awareness and decision complexity, are brought into the discussion in order to highlight the issues with high and low autonomy from the pilots' perspective.

The material presented in this chapter allows for an appreciation of the central research hypothesis (originally introduced in Chapter 1), which is finalised at the end of the chapter.

3.1 More Autonomy, More Authority, More... Silence

Accident 3.1 *On the 9^{th} January 1997, an Embraer EMB-120RT aircraft operated by COMAIR Airlines, Inc., as flight 3272 was flying from Cincinnati/Northern Kentucky International Airport towards the Detroit Metropolitan Wayne County Airport. Despite the icy conditions, the pilots decided to keep the autopilot on.*

*At approximately 15:54 GMT the aircraft crashed during a rapid descent after an automated (**uncommanded**) roll excursion near Monroe, Michigan. All twenty-six passengers and three crew members died as a result of the accident.*

National Transportation Safety Board [1998] reports:

- *"Had the pilots been flying the airplane manually (without the autopilot engaged) they likely would have noted the increased right-wing-down control wheel force needed to maintain*

the desired left bank, become aware of the airplane's altered performance characteristics, and increased their airspeed or otherwise altered their flight situation to avoid the loss of control." (page 178)

- *"If the pilots of Comair flight 3272 had received a ground proximity warning system, autopilot, or other system-generated cockpit warning when the airplane first exceeded the autopilot's maximum bank command limits with the autopilot activated, they might have been able to avoid the unusual attitude condition that resulted from the autopilot's sudden disengagement". (page 179)*

In the late 1980s, it was expected that increased autonomy of on-board systems would reduce human error and workload, whilst increasing operations performance and safety. However, during the last couple of decades, empirical research has provided evidence that automation is not linearly correlated to these parameters, instead it seems to provide new opportunities for different kinds of errors [Woods 1994; Sarter et al. 1997]. It was also expected that the introduction of automation would reduce the amount of training required by humans, but this has not been the case either [Orlady and Orlady 2002].

Recent research reveals that reduced human physical activity in a highly automated environment does not necessarily relate to reduced perceived cognitive workload. Interestingly, in the railway automation domain, Sharples et al. [2009] observe that in a high automation scenario, signallers may need to work harder in order to maintain situation awareness, as they not only need to keep track of the movements of trains on the screen but also maintain an understanding of the way in which the automation is controlling the movements.

There is a history of accidents similar to Accident 3.1 that can be traced back to the erroneous design of highly automated safety-critical processes and systems: Ministry of Civil Aviation - Government of India [1990]; Investigation Commission of Ministry of Transport - France [1989]; Aeronautica Civil Of The Republic Of Colombia [1996]; National Transportation Safety Board [1997; 1986]; Main Commission Aircraft Accident Investigation - Poland [1994]; Aviation Safety Network [1993]; Fiorino [2009]; Aviation Safety Network [2009], amongst many.

The high degree of *autonomy* seems to play a central role in scenarios similar to Accident 3.1. The terms 'automation', 'dependability' and 'autonomy' are extensively used in the literature; however, the interpretations vary depending on the application domain. In this thesis, **automation** is interpreted as the execution by a machine of a function previously carried out by humans [Parasuraman and Riley 1997]; **autonomy** is regarded as the independence of the system from human control [Visentin 2007]; **authority** is a property of automation that allows it to take over control of a monitored process from the humans, if it decides that intervention is warranted, based on its perception of the situation and its internal criteria [Sarter and Woods 1994].

Autonomy and authority are attributes of automation, i.e. automated processes can differ in degree of autonomy and degree of authority. Accident 3.1 is representative of a case history of mishaps in which a mixture of high autonomy and authority in a safety-critical context provides the recipe for catastrophic consequences; Jones et al. [2010] present a detailed review of several aviation mishaps due to onboard automation.

Norman [1990] suggests that most of the time the issue with high autonomy systems is not in the automation itself but in the *feedback* provided to the humans. In fact, feedback allows the operator to evaluate the system state in relation to actions, goals, and expectations. Lack of feedback forces the human into an open-loop processing situation in which performance is generally poor [Hollands and Wickens 1999].

Similarly, Sarter and Woods [1995] express significant concerns about certain "strong and silent" automated processes on-board modern aircraft. The risk is that the pilot loses awareness of the way the automation changes the state of the system, an eventuality that could have catastrophic consequences.

Given the concerns from the human factors community about highly autonomous processes on-board modern aircraft on the one side, and the significant body of research targeting fully automated avionics reconfiguration solutions on the other side, the remainder of this chapter aims at casting light on the question of whether a high degree of autonomy during dynamic reconfiguration represents a step in the direction of improving the safety of next-generation aircraft or not.

3.2 Autonomy and Authority on-board Modern Aircraft

The two biggest aviation manufacturers, Boeing and Airbus, have developed two different philosophies about the degree of autonomy and authority of on-board systems, which are usually referred to as **soft automation** and **hard automation** respectively.

Boeing (soft approach) gives pilots of their latest aircraft models (e.g. 777 series) high authority over the automation, allowing them to override automated actions when needed. Pilots can access the full performance envelope without being constrained by the automation. For instance, if the pilot wishes to exceed set limits such as exceeding 3.5 degrees of bank, or pulling the yoke back as the aircraft decelerates below the minimum manoeuvre speed, she can apply more force than normal on the yoke and the aircraft will react accordingly [Hughes and Dornheim 1995].

On the other hand, the hard automation philosophy used by Airbus preaches that automation technology exists to prevent the pilot from inadvertently exceeding safety limits. Airbus systems have hard speed envelope protection features that prevent pilots from stalling the aircraft; for instance, the automation prevents pilots from pulling more than 2.5g, even in an emergency. The automation has ultimate authority over pilot actions, hence it can override their actions; for example, should the pilot inadvertently take the aircraft beyond its performance envelope, the automation would prevent damage to the airframe and maintain the hard-coded flight dynamics. Basically, automation is employed as a technique for error prevention; it creates a middle-layer between the pilot and the avionics that *filters* the inputs and *masks* eventual non-nominal conditions by compensating through automated actions.

Accident 3.2 *On a non-stop China Airlines Flight 006 between Taipei and Los Angeles (Boeing 747SP-09 aircraft), flying on the* 19^{th} *February 1985, the aircraft suffered significant structural damage while losing almost 30,000 feet in an uncontrollable dive [National Transportation Safety Board 1986]. Because the control inputs that led to the recovery exceeded the performance envelope, the aircraft frame was severely damaged. However, as the avionics actually allowed the pilot*

to exceed the performance envelope, a crash was avoided and only a few injuries were reported [Young et al. 2007].

From a different perspective, consider another case:

Accident 3.3 *On the 24th April 2004, an Airbus A300 aircraft started a landing manoeuvre towards the Nagoya Airport (Japan), under manual control. The First Officer inadvertently activated the GO lever, which changed the Flight Director to GO AROUND mode and caused a thrust increase. This made the aircraft deviate above its normal glide path. The crew started the manoeuvre,* unaware of the abnormal situation. *There was no warning and recognition function to alert the crew directly and actively to the onset of the abnormal out-of-trim condition. During the landing, the co-pilot had to 'fight' against the automation, but the aircraft eventually stalled and crashed. The Japanese Ministry of Transport reports that "the Captain's and First Officer's awareness of the flight conditions, after the PIC*[1] *took over the controls and during their recovery operation, was inadequate respectively" [Ministry Of Transport 1994].*

The fact that the co-pilot had to literally 'fight' against the automation raises severe concerns about a high degree of authority for the automation. However, the two accidents reveal quite evidently that both soft and hard automation approaches seem to have *prō* and *contrā*.

The discussion will now be brought closer to the problem of IMA dynamic reconfiguration.

3.2.1 Searching for the right degree

Sheridan et al. [1978] first proposed ten possible levels of autonomy to define the interaction between humans and computers. More recently, Parasuraman et al. [2000] revised the original levels in the light of a four-stage model of independent information processing functions (information acquisition, analysis, decision making and action implementation, shown in Figure 3.1). The autonomy levels of Parasuraman et al. are shown in Table 3.1, which is the scale used as a reference in this thesis.

#	*Autonomy Levels*
10	The computer decides everything and acts autonomously, ignoring the human.
9	The computer informs the human only if the computer decides to.
8	The computer informs the human only if asked.
7	The computer executes automatically, then necessarily informs the human.
6	The computer allows the human a restricted time before automatic execution.
5	The computer executes the suggestion if the human approves.
4	The computer suggests an alternative.
3	The computer narrows the selection down to a few.
2	The computer offers a complete set of decision alternatives.
1	The computer offers no assistance. The human must make all the decisions and actions.

Table 3.1: Levels of autonomy introduced by Parasuraman, Sheridan & Wickens (2000)

[1] Pilot In Command

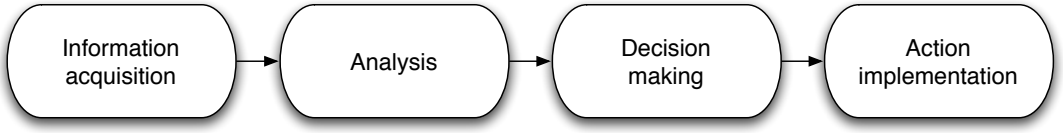


Figure 3.1: Simple four-stage model of human information processing.

Depending on which of the four stages of the human information processing scheme shown in Figures 3.1 is automated, the relevant system autonomy level is chosen from Table 3.1. Full autonomy/authority leads to the automation of all four information processing stages, including 'action implementation', and corresponds to levels 9 or 10 of the scale. The degree of autonomy of the function that caused the crash of the A300 aircraft involved in Accident 3.3 can be ranked at this level.

With minimal autonomy (levels 1 and 2) the pilot would receive minimal or no support from the system in the *selection* of a new configuration. After an unexpected event, the system would prompt the pilot with a list of all the possible applicable configurations. In order to make an informed decision, she would need to investigate the causes of the triggering unexpected event, try to understand why the system was suggesting a specific configuration and what its consequences would be on aircraft functionality; finally the pilot should choose the option that best fitted the operating conditions at that time.

With this level of system autonomy the pilot would have *full authority*, which is theoretically a desirable quality, but, at the same time, she would have to put *in relation* a vast number of heterogeneous information items in a short time, e.g. type of fault, type of alarm, operating conditions, reliability of the instrumentation, implications of different actions, tactical objectives (in case of a military aircraft), and so on. This thesis rises concerns about the ability of a pilot to solve such a complicated problem, in real-time, without any automated decision support.

In this regard, Quesada et al. [2005] point out that Complex Problem Solving (CPS) tasks are:

1. *Dynamic*: early actions determine the environment in which subsequent decisions must be made and features of the task environment may change independently of the actions of the "solver".

2. *Time-dependent*: decisions must be completed within the deadlines defined by the environmental demands.

3. *Relationally complex*: most variables do not have a one-to-one relationship with each other.

Working memory plays a central role in complex problem solving tasks; more specifically, the lateral prefrontal and the parietal cortices are heavily involved in tasks that require keeping a certain number of items easily and promptly accessible at the same time. Braver et al. [1997] show that this capability is very limited.

Halford and Wilson [1998] define **relational complexity** as the number of unique entities that one must process in parallel to arrive at a solution; the authors also define working memory limits in terms of *relational complexity*. Through neural network models and empirical research, they

prove the existence of a soft limit of *four* parallel relations. More recently Cowan [2001] reviewed several studies of cognitive capacity and set the limit at 3 ± 1.

Halford and Wilson assert that complex problems are characterised by numerous elements which are related to each other and cannot be considered meaningfully in isolation. Beyond the 3 ± 1 limit, **segmentation** comes into action; tasks are broken into components that do not exceed processing capacity and can be processed serially.

Segmentation is directly correlated to *cognitive demand*, *decision time* and *probability of error* [Halford and Wilson 1998], which are all parameters that need to be kept as low as possible during avionics dynamic reconfiguration (ADR) decisions. Hogarth [1975] adds that because of the limits of information processing capacity, optimal decision time is a concave function of task complexity, for both simple and complex tasks, decision time is relatively small. However, with very complex problems, the probability of error drastically increases.

The aim here is not to quantify the complexity of ADR decisions. However, Quesada et al. show quite evidently that ADR decisions have all the characteristics of a complex problem whose solution requires a cognitive demand which is likely to exceed human capabilities in the majority of cases.

Just to provide a picture of the relational complexity of the ADR problem, without going too deeply into the details, the avionics of the Airbus A380 aircraft amount to 80 computing modules, each of them can run up to 21 avionics functions which can be activated and deactivated during a reconfiguration and are linked by inter-dependency relationships [Itier 2007]. This represents only one dimension of the information that needs to be processed by the pilot during ADR if the system is designed at autonomy level 2.

All the material discussed in this section supports the claim that some form of decision support would be extremely beneficial for the pilot during safety-critical ADR decisions. The next section elaborates this claim further, introducing an important mental construct which has acquired a lot of attention in the aviation psychology domain in the last couple of decades: situation awareness.

3.2.2 Situation awareness

Situation awareness (SA) plays a central role during ADR (this claim is empirically verified in Chapter 6), hence particular attention is given to the introduction and discussion of this topic. Despite having attracted a lot of interest amongst psychology, human factors and ergonomics researchers in the last couple of decades, a universally accepted definition of SA is still not available. Three definitions seem to dominate the literature [Stanton 2001]:

1. Smith and Hancock [1995] introduce the **perceptual cycle model**: SA resides in the interaction of the human with the world, it is described both in terms of the cognitive processes used to engineer it and the continuously updating product of SA.

2. Bedny and Meister [1999] suggest the **activity theory model**: SA is defined as the human's conscious dynamic reflection on the situation.

3. Endsley [1999] proposes a **three-stage model**: SA is defined as "the perception of the elements in the environment within a volume of time and space, the comprehension of their meaning and the projection of their status in the near future".

The main difference amongst the three models lies in whether SA is considered a *process*, employed in achieving and maintaining a mental state, or the end *product* of this process.

The three-stage model by Endsley [1999] is by far the most cited in the literature and the majority of SA measurement techniques rely on this model. The conclusions reached by this thesis are empirically evaluated, SA is directly and indirectly assessed through qualitative and quantitative methods in Chapter 6, therefore, Endsley's model is used here. The three stages of the Endsley model are as follows:

- *Level 1 SA: Perception of the elements in the environment*: the first step in achieving SA is to perceive the status, attributes and dynamics of relevant elements in the environment.

- *Level 2 SA: Comprehension of the current situation*: This phase is based on a synthesis of disjoint level 1 elements. It goes beyond simply being aware of the elements that are present, to include an understanding of the significance of those elements in the light of pertinent goals.

- *Level 3 SA: Projection of future status*: This stage requires the projection of future actions of the elements in the environment, at least in the very near term.

Endsley's definition of SA is compatible with the RPD model (Section 2.2.2.3). In fact, the first step of the decision problem is collecting information and recognising a situation from past experience. This process is referred to as **situation assessment**. SA is the product of continual situation assessment [Elliot 2005].

Describing the crash of the Embraer 120 RT Brasilia aircraft that happened on the 9^{th} January 1997 (Accident 3.1 in this thesis), the National Transportation Safety Board [1998] reports the following:

- "Had the pilots been flying the airplane manually (without the autopilot engaged) they likely would have noted the increased right-wing-down control wheel force needed to maintain the desired left bank, become aware of the airplane's altered performance characteristics, and increased their airspeed or otherwise altered their flight situation to avoid the loss of control. Disengagement of the autopilot during all operations in icing conditions is necessary to enable pilots to sense the aerodynamic effects of icing and enhance their ability to retain control of the airplane". (page 178)

- "Because the pilots of Comair flight 3272 were operating the airplane with the autopilot engaged during a series of descents, right and left turns, power adjustments, and airspeed reductions, they might not have perceived the airplane's gradually deteriorating performance." (page 145)

The Comair Flight 3272 is a clear example of loss of SA. The history of aircraft accidents counts several mishaps due to this specific phenomenon. Endsley [1995*a*] performed a study amongst the major air-carriers over a period of 4 years and found that 88% of accidents involving human error could be attributed to problems with SA.

Another accident partially due to loss of SA concerns the American Airlines Flight 965 (Boeing 757-223 aircraft) flying between Miami (Florida, USA) and Cali (Colombia) on the 20^{th} December 1995. Aeronautica Civil Of The Republic Of Colombia [1996] reports the following:

- "The evidence suggests several explanations for this deficiency in the flight crew's situational awareness: [...] Terrain information was not shown on the electronic horizontal situation indicator (EHSI) or graphically portrayed on the approach chart" (page 35)

- "Aeronautica Civil determines that the probable causes of this accident were: [...] 3. The lack of situational awareness of the flight crew regarding vertical navigation, proximity to terrain, and the relative location of critical radio aids" (page 57)

A detailed analysis of this accident is provided by Endsley and Strauch [1997].

Endsley [1999] posits that SA is built up in three stages, which can be associated with different cognitive processes. We argue that the capacity of the pilot to build up SA during ADR changes, depending on the level of autonomy of the process, as shown hereinafter.

3.2.2.1 Level 1 SA

By definition, level 1 SA implies a *perception* of the elements in the environment. In terms of ADR, this means perceiving (e.g. visually, through the cockpit, and aurally, through an alarm) the characteristics of the unexpected event that issues the reconfiguration (e.g. a fault), the characteristics of the reconfiguration options generated by the system and other relevant details of the current operating conditions (e.g. maximum time available to make a decision and complete the reconfiguration process).

In a low-autonomy/authority design, pilots would have to process a substantial amount of relationally complex information in real-time which, as discussed earlier in this section, is realistically impossible in the majority of the circumstances. However, in a full-autonomy design, no feedback information is generated for the pilot, the system proceeds autonomously, hence the pilot doesn't have the possibility of perceiving the elements in the environment. As a result, the process of constructing SA is affected at its roots in both the extreme cases of very low and full autonomy and authority.

3.2.2.2 Level 2 SA

Level 2 SA implies the processing of perceived information and the generation of a *mental representation* of objects, events and concepts that are part of the reconfiguration scenario. Knowledge representation is a complicated and debated area of cognitive psychology; at present no definition has been fully accepted and most of those proposed are very technical and able to capture only

some aspects of the general concept. In this thesis we refer to two complementary definitions. The first provided by Smith and Kosslyn [2007]:

Definition 3.2.1 *A [knowledge]* ***representation*** *is a physical state (such as marks on a page, magnetic fields in a computer, or neural connections in a brain) that stands for an object, event, or concept. Representations also carry information about what they stand for. [...] But representation involves more that this.*

Mental representation is a complicated process that still requires scientific exploration. However, it is widely accepted that the process must meet two criteria [Smith and Kosslyn 2007]:

- **Intentionality criterion**: A representation must be constructed intentionally to stand for an object, event or concept. This can happen unconsciously, because the brain at an unconscious level has features designed to store in memory information about experiences of the world, to stand for those experiences.
- **Information-carrying criterion**: A representation must carry information about what it stands for.

The second definition is provided by Besnard [2004]:

Definition 3.2.2 *A* ***mental model*** *is a scarce, goal-driven image of the world that is built to understand the current and future states of a situation.*

For the sake of clarity, in the remainder of the document 'mental representation', 'mental model' and 'schema' are used interchangeably.

Once the brain intentionally establishes representations that carry information about something, all sorts of sophisticated cognitive abilities become possible. However, serious issues arise in safety-critical contexts when these models are either wrong or poor. It is worth spending a few more words on mental models, because they are of critical importance for the conclusions that will be drawn later.

In their day-to-day job, pilots construct **homomorphic mental models** [Moray 1987] of the system, simplified models made of all the mental representations related to the system constructed up to the present time. In contrast, an **isomorphic mental model** is one that would require the pilot to have a comprehensive and correct knowledge of the system, including all its possible states. This is not possible given the complexity of modern aircraft. For more information on mental models, Moray [1996] provides a detailed taxonomy.

A homomorphic model is a simplified state of the system which is easily broken when new situations and states are encountered. This situation is described by Baxter et al. [2007] as **cognitive mismatch**, which is a disparity between the operator's mental model of the system and the way the system is really working. A cognitive mismatch can be considered a precursor of loss of SA. The higher the accuracy of mental models, the lower the risk of cognitive mismatches, even when there is significant inconsistency between the problem-solving processes of the human and the system [Lehner and Zirk 1987; Kaber et al. 2001].

Because of the complexity of modern aircraft, no form of training can be realistically devised to build an isomorphic model of the system in the pilots' brains. However, pilots' homomorphic models of the system can be enriched by increasing their involvement in the reconfiguration process, for example, by questioning them before applying a new configuration, by providing them with decision support information like fault description, implications of the current reconfiguration, reasons to switch off Application A instead of Application B etc.

Fully autonomous design (autonomy levels 9 and 10 on the Parasuraman scale) do not favour the construction of rich and correct mental models. With such a design, pilots do not receive any feedback from the system before and/or after an ADR. The result is that the state of the system changes during operation, but the mental model of the pilot is not updated accordingly. This is a perfect recipe for a cognitive mismatch.

With a low autonomy level (levels 1 and 2) the pilot is provided with a vast amount of information concerning all the applicable configurations. However, the well documented limitations of memory and processing capabilities will eventually lead to the construction of poor homomorphic mental models (the issues with problem complexity and cognitive limitations have already been discussed earlier in this section). The phenomenon of poor mental models construction is exacerbated in situations of time pressure and stress [Zakay and Wooler 1984; Benson III and Beach 1998; Weenig and Maarleveld 2002*a*; Klapproth 2008].

With a *medium autonomy* design (between levels 3 and 6), the pilot would be actively involved in the ADR process and, at the same time, would not be overloaded with information. The effectiveness of such a design in relation to the construction of correct mental models of the situation is a function of how *reasonable* the exchange of information is. The number of configuration options, with annexed attributes, that can be effectively processed by the pilot in real-time is one of the topics investigated in the experiments described in Chapter 6.

3.2.2.3 Level 3 SA

As previously mentioned, it is not possible for the pilot to successfully achieve this SA level without having gone through the previous steps. For instance, it is not possible to forecast the implications of a reconfiguration if the pilot does not have the chance to construct a mental representation of the current state of the avionics. Mental simulation is a critical cognitive function for this process and representations are an input for it.

For the moment, it is concluded that both high and low autonomy levels seem unsuitable for ADR in the light of the material presented in this section; since they do not facilitate the construction and real-time update of correct mental representations of the system, they are likely to jeopardise the achievement of level 3 SA.

3.2.2.4 Automation surprises

It is a well documented fact that in highly automated processes under human control, lack of SA and cognitive mismatches can easily lead to **automation surprises**, failures of the human operator to track, monitor, or anticipate the actions of automated systems, leading to unintended system

behaviour [Sarter et al. 1997].

Mumaw et al. [2001] relate automation surprises to mode errors and loss of mode awareness, defining **mode awareness** as "the knowledge and understanding of the current and future state and behaviour of the system". They also describe **mode error** as an error that "occurs when the pilot performs an action appropriate for the assumed system state but not for the actual state". Mode errors lead to automation surprises when the pilot notices that the automation is engaged in activities that were not commanded. Unfortunately this phenomenon is common in cockpit automation and despite the amount of research performed in this field so far, the problem is still open and represents a contentious issue [Mosier 2010].

To give an idea of the incidence of the phenomenon, Olson and Sarter [2000] conducted a study of automation management strategies on 206 airline pilots flying for two major U.S. air carriers. Their sample included 59 B-757, 84 A-320, and 63 MD-11 pilots. To the question: "Have you experienced cases in which automation did too much or too little?", 151 pilots out of 193 (78.2%) responded they had been *surprised* by the automation. More specifically, the automation did more than expected for 39 of them, less than expected for 55 of them, both less and more for 57 of them.

Extensive research, both in the fields of aviation and medicine, reveals that the combination of strong autonomy and poor feedback from an automated system is a precursor of automation surprises [Van Charante et al. 1992; Woods 1996; Sarter et al. 1997; Billings 1997; Woods et al. 2002]. There seem to be no reasons why a high autonomy ADR design should be an exception.

In fact, reconfigurable avionics are designed to continuously adapt to changing operating conditions. In full autonomy and authority systems, the mental representations could be correct in the early stages of the operation but over time, after one or more reconfigurations, if the pilot is not made aware of the characteristics and implications of each configuration in a correct and timely manner, the mental representations eventually become obsolete and any following action could hide a risk. In this regard, Baxter and Ritter [1999] report that during interactions with dynamic systems, as time passes and the system is subject to degradation (e.g. due to a fault), the human mental representations get simplified and become more based on correlation between system elements.

In conclusion, autonomy levels that do not foster the continuous update of pilots' mental representations (1, 2, 7–10 on the Parasuraman scale) should not be considered for ADR, as they would most probably lead to automation surprises during operation.

3.2.2.5 Conclusions

Mark and Kobsa [2005] define a system as 'transparent' when the underlying reason and information behind the behaviours of the automation are understood by the human. A reasonable degree of transparency is vital for the efficiency of the interaction and to avoid automation surprises during reconfiguration.

All the material in this section leads to the conclusion that both high and low autonomy levels do not provide a reasonable degree of transparency, hence they are not suitable for ADR; given the specific characteristics of the ADR problem, both high and low autonomy levels are likely to jeopardise the pilot's ability to achieve SA at each level of Endsley's model.

This section provides further support to the claim made in Section 3.2.1 concerning the automation of the ADR process, that there are serious concerns about the ability of pilots to perform ADR decisions safely, without any kind of decision support, especially in harsh operating conditions.

3.2.3 A note about minor and major reconfigurations

The Lockheed Martin F-22 Raptor aircraft [Lockheed Martin 2010], one of the most advanced military aircraft on the market, mounts a state-of-the-art Integrated Modular Avionics with reconfiguration capabilities, called 'Block 3.0 avionics' [Caires 2001; Caires and Stout 2002].

The designers of Block 3.0 have distinguished between *minor* and *major reconfiguration* [Spitzer 2000]:

- **Minor reconfiguration**: occurs due to the loss of one or more modules. A module is reprogrammed to perform functions previously executed by other modules. When all the spare computing modules are used, the lowest-priority function is dropped on behalf of higher-priority functions.
- **Major reconfiguration**: occurs due to the loss of an entire rack of computing modules. It can be caused by battle damage, loss of an engine and/or power generator, or overheating due to loss of cooling. It allows reprogramming and reconfiguring an entire rack in order to guarantee basic functionality like communication systems, basic navigation systems and landing instruments.

Military pilots have extreme conditions of time pressure, stress and mental demand. Even though this thesis is not focused on military aircraft, the highly disadvantageous operating conditions of combat pilots can be exploited to draw out further conclusions that can be applied to the dynamic reconfiguration of SCMS in general.

Major reconfigurations fall exactly into the type of reconfiguration discussed so far. The importance of human involvement for the improvement of system dependability during such an invasive process has been already elaborated. One could argue that human involvement could be avoided during a minor reconfiguration. This seems to be true only in part. In fact, the priority of a sub-system is defined by system engineers off-line but the knowledge they rely on is permeated by epistemic uncertainty: only pilots know what is really high priority in relation to the current operating conditions and their plans. During a minor reconfiguration, the system could shut down a low-priority function which is actually high-priority for the pilot in that specific circumstance. Human intentions are unpredictable. This argument brings the discussion back to the topic of integrity and safety (Chapter 2). In complex, manned systems like modern aircraft, integrity alone is not enough to achieve safety.

Human involvement could be unnecessary when the functionality of the aircraft does not change, i.e. when there are enough spare computing modules and no low-priority functionality is dropped as a result of an ADR. However, if any 'low-priority' function is dropped, even during a minor reconfiguration, then human involvement seems to be critical, in order to avoid cognitive mismatches and automation surprises.

3.3 Hypothesis Statement

It is now possible to identify the central hypothesis of this research:

> During the process of avionics dynamic reconfiguration, decision support information that parallels cognitive strategies by including *explanations*, *implications* and an *assessment of the uncertainty* associated with the reconfiguration advice provided by the system should have a positive effect on pilot *situation awareness*, *workload*, *decision accuracy* and *performance*, thus it should improve the overall decision making effectiveness of the pilot and, as a result, the safety of the process.

The hypothesis defines the components of the decision support information design proposed (i.e. explanations, implications and uncertainty assessment) and the aspects of pilot behaviour that should benefit from it (i.e. situation awareness, workload, decision accuracy and performance). Chapter 4 provides the links between all these elements, after discussing them in detail.

In the remainder of this thesis a number of claims are raised about the behaviour of pilots during avionics reconfiguration decisions and about how effective decision support information should be designed. The claims are used to develop an architecture for the automated generation of decision support information for ADR decisions, in accordance with the hypothesis. A set of subjective and objective metrics for the assessment of the effectiveness of the decision support will be identified and if the hypothesis is correct, a measurable improvement will be detected throughout the series of experiments described in Chapter 6.

3.4 Chapter Summary

The superimposition of the basic four-stage human information processing scheme on the scale proposed by Parasuraman et al. [2000] partitions the autonomy levels into four sets, as shown in Figure 3.2.

The *prō* and *contrā* of the degrees, grouped by low, medium and high autonomy, have been evaluated in the light of two main arguments, problem complexity and situation awareness. The material presented in this chapter brings a compelling amount of evidence against the applicability of both high and low autonomy designs to the ADR process.

"Strong and silent" automation would *apparently* (this is discussed again, later in the thesis) reduce pilot problem complexity. However, the costs in terms of safety exceed the benefits. Medium autonomy (levels 3 to 6) seems to better fit the ADR process. As a consequence, it is reasonable to 'break' full autonomy at decision making level (Figure 3.2).

The material presented so far leads to the following conclusions:

- the complexity of the IMA dynamic reconfiguration problem makes *automation unavoidable*;
- however, a *high autonomy/authority* design constitutes a risk for the process of dynamic reconfiguration of IMA.

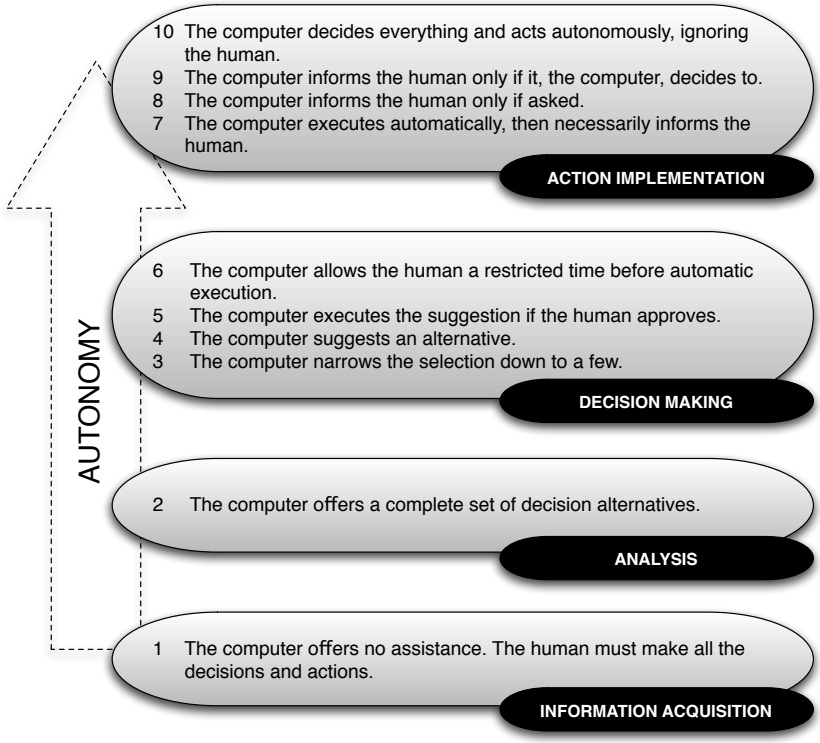


Figure 3.2: Autonomy levels defined by Parasuraman et al. [2000] divided by information processing stages.

- given the safety-critical context, and in line with the EUROCAE ED-79 standard for IMA (see Section 2.1.3), pilots should be provided with appropriate *decision support*.
- humans should be involved at *decision making* and *action implementation* levels (in Parasuraman's model). This means that the final decision, before performing a reconfiguration, is up to the human (some exceptions to this conclusions are discussed further, e.g., a very stringent decision time budget).

Having acknowledged the necessity of making the human an active part of SCMS dynamic reconfiguration, ways to effectively involve him or her in such a complicated, safety-critical process require investigation. This is the topic of the next chapter.

Chapter 4

Human Involvement During Reconfiguration

Human decision making activity in real, complex contexts is influenced by a huge number of factors, including emotions, stress, time pressure and the framing of the information provided to the decision maker, to mention but a few. The list could be extended *ad infinitum*.

In order to provide a grounding for the design of an effective decision support system, the first part of the chapter (Section 4.1) takes a pragmatic approach and discusses a number of factors which are likely to have a significant impact on the type of decisions investigated in this research. Assuming that taking into consideration all possible factors that could influence ADR decisions is an option of dubious practicality, the analysis undertaken is set to understand the main characteristics of ADR decisions, advance claims about how to improve safety *which can be empirically tested* and draw conclusions that can also be generalised to dynamically reconfigurable SCMS.

For each factor of influence, one or more claims are made, which are used to drive the experiments. When the literature relevant to a claim is in agreement, the claim is intended to motivate empirical confirmation of known ideas in the particular context of ADR decisions. When the literature contains contrasting positions, the claims are more speculative and aim at motivating empirical tests that target the ADR problem specifically, to assess which, if any, of the positions apply. The factors are not presented in any particular order.

It must be noted that, throughout the process of empirical verification presented below, the set of brain mechanisms employed by the pilots are considered to be "black box"; however, the *effects* on the overall pilot decisional behaviour are measured and discussed. In other words, this thesis does not question the results from the cognitive psychology studies referenced; instead, it uses them to make predictions that are empirically verified.

The second part of the chapter (Section 4.2) builds on the discussion of the factors presented in the first part; a new set of claims about how to mitigate the impact of each factor on pilot decisions by means of tailored decision support information is put forward. The new material is used to define the content and structure of *effective* decision support information for ADR.

Finally, all the claims made in Sections 4.1 and 4.2 are re-considered against the research hypothesis originally presented in Chapter 3 (Section 3.3). This step enables a precise agenda

for the empirical investigation of the hypothesis to be defined; each claim made throughout this chapter is associated with one of the seven experiments that will be presented in Chapter 6. An estimate of the work done so far and of the remaining work is also provided.

4.1 Decision Biases

4.1.1 Risk and uncertainty

Slovic [1999] argues that **perception of risk** is a complicated matter because there is no clear and universally accepted definition of it. Risk is a social construct invented to cope with the dangers and uncertainties of life. A review of the literature shows a huge number of definitions; however, two of them are particularly appropriate to this study:

1. Slovic [1987] associates the lay concept of risk with "hazards that fill one with dread and/or are *poorly understood*".

2. Fischhoff and Lichtenstein [1984] define risk as "a multidimensional construct with dimensions labelled as dread, lack of familiarity, and *lack of controllability*".

Slovic's definition brings the roots of risk back to a poor understanding of the functioning of a system, hence it links to the discussion about the issues with a pilot's homomorphic representations of the system (Section 3.2.2). If pilots realise their understanding of a process is poor, their perception of risk grows.

Fischhoff and Lichtenstein identify *lack of controllability* as a dimension of risk. An interpretation relevant to the ADR problem is that pilot perception of risk would grow when and if they realise they are not in control of the process to the degree they feel they should be. Interestingly, it has been demonstrated that perceived uncontrollability of adverse events is among the greatest nonspecific human stressors, leading to a host of undesirable effects, including depression and anxiety [Johnson and Sarason 1977] and even illness [Stern et al. 1982]. Studies from heterogeneous fields, including military tactics, medical, aviation, nuclear and manufacturing [Cook et al. 2007] reveal that the perception of risk influences the way humans make decisions.

In a study about strategies for risky decisions, Russo and Dosher [1983] found evidence of the emergence of the **'elimination by aspects'**: the attributes of each choice are evaluated and the choices whose attributes don't match the decision maker's criteria are eliminated.

Mellers et al. [1998] report that there is widespread agreement that risk and uncertainty lead the decision maker to adopt a strategy of *ranking* the available options. They also maintain that specific issues about the utility function and the weighting function, including shape, form (cumulative vs noncumulative) and factors that influence it are still debatable.

Elimination by aspect and ranking are just two of several strategies that emerge when the decision maker faces risks. There is no gain in expanding the list and investigating which specific strategy is more likely to be adopted by pilots during ADR decisions, because it is probable that each of them will behave slightly differently. In the context of this study, what is really interesting to understand, regardless of the specific strategy of choice, is *if the uncertainties that pilots face*

during ADR can possibly lead to a significant perception of risk that is observable as a clear bias to their decisions.

It could be argued that, by gathering evidence of this effect, the research mentioned so far make this conclusion obvious. However, other studies note that such behaviours are less likely to occur with highly experienced decision makers such as aircraft pilots [Christensen-Szalanski and Beach 1984; Fraser et al. 1992; Gigerenzer 1987; Shanteau 1992; Smith and Kida 1991].

Additionally, some research collects evidence of how analytical decision models (for example, elimination by aspect) used in some studies fail to provide coherent answers in specific situations, characterised by time pressure, uncertainty, complexity and high stakes (see Salas and Cannon-Bowers [2003]; Cannon-Bowers and Salas [1998]; Salas et al. [1995] for a review). As a result, experimental investigation is required to assess the influence of the perception of risk on ADR decisions.

In the gambling domain, a well documented phenomenon related to risky decisions which is called the **reflection effect** seems to be more or less evident in the vast majority of people [Kahneman and Tversky 1979]: decisions are typically risk averse in the gain domain, but they are frequently risk seeking in the loss domain. In gambling, the risk is due to potential losses; during SCMS reconfiguration, the risk is given by the safety-critical context. It is interesting to investigate whether the reflection effect also applies to the ADR problem or not. For instance, consider the following two typical ADR scenarios:

1. *No risk scenario*: the aircraft is in 'enroute cruise'. Previously, a fault to a power generator led to a reconfiguration. A sub-optimal configuration was applied. Whilst flying, the ADR system continues to work in the background and finally generates a new and optimised configuration; it then asks the pilot whether he/she wants to apply it or not.

2. *Risky scenario*: the aircraft is in the 'descent' phase (higher risks than in 'enroute cruise'). A sudden fault to a power generator, three minutes before landing, results in no redundant buses or critical functions, i.e. any critical function represents a single-point-of-failure for the whole avionics and risks jeopardising the landing manoeuvre and the safety of the crew. The ADR system provides one or more reconfiguration alternatives to choose from, which entails deactivating one or more critical functions.

If the reflection effect applies to ADR decisions, then it is expected that the pilot would not accept the ADR suggestion in the former scenario (if it works, why change it?), but they would do so in the latter scenario. On the other hand, given the safety-critical context, changing the state of the system in an unstable situation (e.g. before landing) could become a catalyst for potential catastrophic consequences and pilots would probably avoid this as much as possible. This issue is a determining factor for the definition of the autonomy of the ADR process. We now formulate the following claim:

Claim 1 *If prompted by the system, pilots would choose to reconfigure the avionics in situations which are not particularly risky (e.g. whilst cruising). They would refrain from doing so in situations of pressing risks (e.g. before landing, when the system is more unstable and a change to the current state could provide a catalyst for catastrophic consequences).*

This claim basically contrasts the applicability of the reflection effect to safety-critical scenarios.

As the perception of risk is by definition due to uncertainties and poor understanding of the functioning of the system, the decision support information should be shaped to reduce pilot uncertainties and improve pilot trust. In this regard, it is interesting to consider Lipshitz and Strauss [1997], who found that decision makers distinguish between three types of uncertainty, in the decision support information they obtain: (a) inadequate understanding, (b) incomplete information and (c) undifferentiated alternatives. Inadequate understanding is primarily managed by strategies aimed at reducing the complexity (such as searching for more information and delaying the decision, if applicable); incomplete information is managed by assumption-based reasoning; conflict amongst alternatives is managed mainly by weighing the pros and cons (i.e. choosing between alternatives in terms of potential gains and losses).

Hence, if Claim 1 is verified, it seems reasonable to argue that an effective DSS for ADR should:

- provide a reasonable (in cognitive terms) amount of information to be processed in real-time *but*, on request, it should allow further details to be obtained in order to reduce complexity and uncertainty (e.g. the DSS would allow the system to be queried for more information concerning each configuration);

- generate uniform and complete information for every configuration;

- generate information such that the differences amongst any configurations are explicit to the 'eyes' of the pilot.

4.1.2 Time pressure

As mentioned in Section 2.1.3, avionics reconfigurations can be either planned (for example, issued to perform a mode change or, in the case of a military aircraft, a mission change) or unplanned (for example, issued to mitigate the effect of a fault). Planned reconfigurations are *usually* performed when operating conditions allow the pilots to concentrate on the operation, when there is enough time and cognitive resources available. Unplanned reconfiguration, however, is usually a response to an emergency situation. It typically happens in the worst operating conditions, when pilots are challenged by critical tasks that require attention. In brief, unplanned ADR are a 'bomb' in terms of cognitive demand.

Under conditions of no time pressure, humans accumulate sensory evidence for one decision option over another, until a fixed threshold is reached [Ratcliff and Rouder 1998; Bogacz et al. 2006]. This rarely applies to time pressure scenarios because the time available to make a decision is not enough to reach the fixed threshold.

Several researchers have studied the effects of time pressure on complex decisions, agreeing that this condition can lead to perceptual narrowing and thus, to a reduced utilization of available cues, decreased vigilance and reduction in working memory capacity [Janis and Mann 1977;

Stokes et al. 1987; Orasanu and Fischer 1997; Orasanu 1997; Zsambok and Klein 1997; Klein 1997*a*; Klapproth 2008]. Less clear is the effect on experienced decision makers, like pilots.

It makes sense for this study to investigate whether time pressure influences the decision making behaviour of pilots during ADR decisions and, if this were the case, it would be interesting to understand how. The following sections address this topic in more detail.

4.1.2.1 Time pressure related strategies

Under extreme time pressure conditions, **fast-and-frugal-strategies** can eventually emerge. Decisions are made without any stimulus processing [Yellott 1971]; they are much faster than usual decisions, but they have only a reduced probability of being correct. Gigerenzer and Todd [1999] show that these strategies sometimes work better and faster than more complicated algorithms. For instance, the research demonstrates how humans make accurate judgements relying on a single piece of information, if they know that piece of information is correct.

We argue that this is not the case for ADR decisions. In fact fast-and-frugal-strategies rely on stimulus-response habit and/or on skill learning mechanisms. **Stimulus-response habits** emerge through the slow accumulation of knowledge about the *predictive* relation between a stimulus and a response [Smith and Kosslyn 2007]. For ADR decisions, the stimulus is the occurrence of an unexpected event, along with a request to apply the configuration suggested by the system, possibly including alarms and warning messages. The characteristics of each configuration depend on the current operating conditions; as previously mentioned, there are a huge number of events that could trigger a reconfiguration at any time and, for each event, the system would generate one or more bespoke applicable configurations. Given the numbers in question, it is impossible for the pilot to establish stimulus-response habits over time.

Similar conclusions can be extended to the more general topic of **skill learning**, which consists of three consecutive stages [Fitts and Posner 1967]. During the *cognitive stage* the knowledge is declaratively represented in memory; attention demands are high at this point in the process. With practice, the human moves towards the *associative stage* in which the behaviour begins to become tuned and error rates decrease. Eventually, the decision maker could reach the *autonomous stage*; the behaviour is highly accurate, the execution is rapid and automatic, requiring little attention. Fast-and-frugal-strategies require reaching the autonomous stage of the skill learning process which, for the same reasons given for the stimulus-response habit, cannot be achieved in the context of ADR decisions.

Fast-and-frugal-strategies are extreme methods to save both decision time and cognitive resources. Whilst they are not characteristic of ADR decisions for the reasons given above, other 'less extreme' decision behaviours could be more relevant for the decisional context examined in this thesis.

A good portion of the literature supports that people cope with time pressure using three main strategies, depending on the specific decision scenario, a) acceleration, b) selection of information, and c) alteration of the information search pattern [Ben Zur and Breznitz 1981; Edland and Svenson 1993*a*; Johnson and Payne 1995]. The order of enumeration of the three strategies reflects their hierarchy of application: acceleration is the first strategy used under time pressure; when it is

not sufficient to meet the task demands, the decision maker switches to selection, and eventually to alteration of the search pattern.

Acceleration means that the decision maker works faster as a result of time pressure; for example, less time is spent on each decision option. **Selection** entails focusing only on a specific portion of the available information; as a result of time pressure, people seem to concentrate only on the most important 'chunks' of information. **Alteration of the search pattern** represents a change in the way the available information is explored by the decision maker. This usually involves a switch from an alternative-based to an attribute-based search strategy, which is less cognitively demanding.

Payne and Braunstein [1978] focus on decision problems characterised by the *increasing* number of alternatives, which is a dimension of complexity. This type of scenario is applicable to the ADR problem. If the autonomy level for the ADR process is set at 3 on the Parasuraman scale (Section 3.2.1), then the pilot could be prompted with a *varying* number of configuration from which the choice should be made. The participants to Payne and Braunstein's study were found to switch from an alternative-wise information search pattern to the less cognitive demanding attribute-wise search pattern with an increase in the number of alternatives. Similar findings are reported by Lohse and Johnson [1996], Cook and Swain [1993], and Weenig and Maarleveld [2002*b*].

An empirical study by Weenig and Maarleveld [2002*a*] reveals partially contrasting results: the screening is reduced to fewer attributes and alternatives in relationally complex decisions under time pressure, but no sign of acceleration is found.

Beach [1993] offers yet another slightly different option: in the author's view, people seem to cope with complex choice tasks under time pressure by screening the information about alternatives for violations of the minimal level of acceptance in one or more attributes ('cut-off points'). A decision option is rejected if the number of violations on various attributes exceeds an individual rejection threshold. The complexity of a decision is reduced through limiting the number of options to choose from.

Empirical results of a study by Benson III and Beach [1998] support another variation on the classical three-strategies theory. They focus on the differences in reactions to time pressure, but distinguish between complex and relatively simple choices. In contrast to findings on simple choice tasks, they found no sign of change in information selection or a switch in search strategy. However, both Ben Zur and Breznitz [1981] and Benson III and Beach [1998] acknowledge the possibility that it might be that the time constraint which Ben Zur et al. imposed on their participants was not high enough to evoke one of the other two reactions to time constraint.

In the light of the material presented so far, the following three sections advance claims concerning three aspects of pilot behaviour during ADR decisions: decision strategy, decision accuracy and decision time. The claims advanced are then used to define the characteristics of the decision support system proposed later in the thesis.

4.1.2.2 Effects on decision strategy

The literature reviewed above shows that the three-stage model of decision making under time pressure provides a good approximation of what can be expected of a human facing this type of

task. However, there is also disagreement on the hierarchy of the stages of the model and its applicability to different decision contexts. As a result, it is not possible to make predictions on how pilots would react under time pressure during a typical ADR, without experimental tests. The following claim is advanced to motivate empirical investigation:

Claim 2 *Pilots would react to time pressure by means of: a) acceleration, b) selection of information, and c) alteration of information search pattern.*

There is more to be said concerning the alteration of search patterns; the fact that humans tend to concentrate on the most important attributes when they increase the selectivity of their information search [Edland and Svenson 1993*b*] suggests a linear relationship between the importance of a 'chunk' of information and the degree of attention to it, i.e. attention to the least important attribute suffers first and most from time pressure and attention to the most important attributes last and least.

However, Weenig and Maarleveld [2002*a*] argue that in a complex and strongly demanding task such as ADR this relationship may be curvilinear rather than linear, with the strongest impact of time constraint on the attributes of moderate importance. They infer that during complex tasks, it is probable that many people ignore some information anyway, regardless of time pressure and this will most likely concern the least important attributes. On the other hand, even under severe time pressure, decision makers are not likely to ignore information about the most important attributes; this is especially true in safety-critical decisions. In complex decision tasks, time pressure may therefore have relatively little impact on attention to the most important and the least important attributes, but would have the strongest negative impact on attributes of moderate importance.

These arguments are now brought back to the ADR problem. For reasons that will be clarified below, we maintain that in the information generated by the proposed DSS, the *implications* of applying a configuration (what if...) would have highest importance; therefore, they would attract the attention of the pilot under any circumstances. Instead, the *reasons why* the system reaches certain conclusions (e.g. why does it suggests `Configuration-A`?) would attract the pilot's attention when there is no time pressure (pilots would use this information to construct situation awareness) but would be discarded when a decision has to be made in, for example, fifteen seconds. In our hypothesis, this type of information is of medium importance.

These arguments enrich the expectations raised by Claim 2, allowing speculation on *how* the information search pattern would be altered under severe time pressure. This discussion continues in Section 4.2, when further arguments are introduced and the formulation of another, more specific claim is made.

4.1.2.3 Effects on decision accuracy

Regardless of the strategies adopted by pilots to cope with time pressure, it is important to understand whether time pressure affects decision accuracy during ADR or not. Zakay and Wooler [1984] and Zakay [1993] collect evidence that shows how time pressure might keep decision makers from choosing the best option, since they divide their attention between estimating the elapsed time and selecting an alternative. The following claim needs to be empirically verified:

Claim 3 *Time pressure would decrease pilot decision accuracy.*

This claim is somewhat controversial; although there is evidence that faster processing of information sometimes correspond to less careful consideration of alternatives [Benson III and Beach 1998], it has also been demonstrated that the application of simplified and even more effective strategies could improve decision accuracy [Staw et al. 1981].

4.1.2.4 Effects on decision time

Hogarth [1975] proposes that due to the limits of information processing capacity, the optimal decision time is a concave function of task complexity; for both simple and extremely complex tasks, decision time is relatively small.

Later studies on variation of decision strategies under time pressure that have been mentioned so far are in agreement with Hogarth's hypothesis. For instance, the decrease in decision time during extremely complex tasks can be brought back to a variation in information selection strategy: the number of options evaluated is drastically reduced to a few or even a single one (fast-and-frugal-strategies) under severe time pressure. From the series of experiments performed in this research, it is expected that information about the limit of configuration options that pilots can process in real-time before the problem becomes intractable will be collected. This limit is expected to slide under time pressure.

During avionics reconfiguration, time pressure is provoked by showing a timer which reminds the pilot of the approaching deadline for the completion of the reconfiguration. In this regard, Trujillo et al. [2008] report that predictive information (e.g., a timer that informs about the imminent occurrence of an event with severe consequences) improves human decision performance during real-time safety-critical processes. However, in that study, no severe time pressure conditions were simulated and, in those circumstances, the difference between having and not having predictive information was approximately 30 seconds; in this research it is more interesting to collect information about the behaviour of pilots in extreme situations in which, for example, 30 seconds is the total amount of time available for a decision. The rationale is that this is a realistic circumstance for ADR decisions.

This thesis aims at extending the results from past research by Hogarth, Trujillo et al. starting with their results and looking at what happens during ADR decisions when a predictive timer is shown, but the time available is very limited. To the best of our knowledge, this case has not yet been investigated in the literature.

The following claim is therefore formulated and requires verification:

Claim 4 *Time pressure would reduce the number of configuration options considered and the way their information is explored.*

4.1.3 Stress and Frustration

Niedenthal and Kitayama [1994] describe **emotion** as "a set of adaptive functions of acting or responding to stimuli that are prewired or 'prepared' by biological evolution and yet at the same time, shaped, elaborated, and finely configured by social and cultural learning".

Emotions in general have been found to determine people's cognitive strategies [Mosier and Fischer 2009]. For instance, a number of studies concerning the effect of fear and anxiety reveal that both feelings are associated with risk-averse choices and with the perception that a situation is not under one's control [Isen et al. 1988; Lerner and Keltner 2000; Lerner and Tiedens 2006]. Interestingly, Loewenstein and Lerner [2003] argue that fear and anxiety lead to more systematic and comprehensive information processing.

Section 4.1.1 mentions a number of studies that support the hypothesis that decision biases are less likely to appear in expert decision makers. In principle, this should apply also to emotional states. However, Estrada et al. [1997] warn that this position overlooks the important distinction between incidental (or task-irrelevant) affect and task-integral affect.

Integral affect concerns emotional responses that are elicited by the decision situation itself or its potential consequences. Usually the pleasantness or unpleasantness of the task, or the effort required, are considered for empirical analysis.

Incidental affect instead is not related to the decision in course, it is brought into context by the decision maker. Estrada et al. maintain that experts are susceptible to incidental affect and, furthermore, they may not always succeed in recognizing it. However, this type of affect is not relevant to this study, therefore, it is not taken into consideration here.

Contrary to the position of Estrada et al., Reber [1989] elicits conclusions from empirical tests on fire–fighters and pilots (expert decision makers in their respective contexts) arguing that being cognisant of an immediate threat to life (which is an integral affect) may result in the optimal arousal of the core affective circuit. Reber argue that such a psychological state may have improved the ability of decision maker to solve the problem using knowledge acquired during past experience.

Controversial positions concerning the actual impact of integral affect on expert decision makers are also documented in neuroanatomical studies. LeDoux [1998; 2002] argues that emotion and cognition operate in two distinct regions of the brain, the lower and upper cortical centres respectively. However, more recent studies reveal that emotions are a form of cognition and since the brain makes no distinction between them, they should be seen as complementary processes [Duncan and Barrett 2007; Storbeck and Clore 2007].

The subject is particularly interesting for the ADR problem, especially given the contrasting positions in the literature. However, only a single, small, aspect of such a complex topic can (start to) be addressed. The focus here is on *frustration*, resulting from situations of heightened stress.

4.1.3.1 Stress and frustration during ADR

The literature contains several definitions of stress. As a matter of choice, in this research the one proposed by Salas et al. [1996] is adopted. They define **stress** as "a process by which certain work demands evoke an appraisal process in which perceived demands exceed resources and result in undesirable physiological, emotional, cognitive and social changes". This definition seems to be particularly suitable to this work because 'demand exceeding resource' is a key factor during ADR decisions.

Closely related to stress is **frustration**, the *emotion* that accompanies an experience of being

thwarted in attaining certain goals [Salmon et al. 2006]. Situations of stress can possibly lead to the emergence of frustration. We speculate that frustration is likely to emerge in extreme ADR scenarios (e.g. after a safety-critical fault, with a very limited time budget to complete the process before the occurrence of more severe consequences).

Despite research so far, the general feeling in the academic community is that the relationship between frustration, stress and decision behaviour has not yet been adequately explored [Gillis 1993; Hammond 2000*b*; Kowalski-Trakofler et al. 2003]. Some empirical studies have proved that negative effects like frustration can produce a narrowing of attention and a failure to search for new alternatives [Fiedler 2001]. Luce et al. [1999] found that people in negative moods make more attribute-based comparisons than alternative-based comparisons (the former are less demanding in cognitive terms). In addition, Luce et al. highlight that decision makers in negative moods make faster and less discriminate use of information that can increase accuracy in easier tasks and decrease it in harder tasks.

As a result of an investigation into the influence of arousal states on human decision making behaviour, Mano [1992; 1999] obtains results similar to those of Luce et al. and in line with the 'prospects theory' in general. Mano asserts that decision makers in positive moods deliberate longer, use more information, and examine the same information more times than others, whereas those who are aroused and in unpleasant moods employ simpler decision strategies and form more polarised judgements.

4.1.3.2 Controversy

Whilst the studies mentioned so far document the negative effects of stress and frustration, other work focusing specifically on stress (which is a potential precursor of frustration) shows that its effects on decision performance are not negative in all cases. It is generally accepted that both improved and degraded performance can be associated with increased stress [Poulton 1976; Gillis 1993; Kowalski-Trakofler et al. 2003]. This relationship is described by an 'inverted-U arousal-performance' model [Evans 1984]: for some individuals, heightened stress elevates decision performance; other individuals are vulnerable to the negative impacts of stress, which results in diminished performance (Figure 4.1).

Gillis [1993] clarifies that stressful circumstances do not necessarily lead to decreased decision making performance in all cases, because the ability to cope with stress is dependent on the human's perception and/or the interpretation of an event. In other words, the emergence of frustration depends on both the specific characteristics of the stressful situation and the pilot's mental representation of it.

Flanagan [1954] and Kowalski-Trakofler et al. [2003] focus on situations in which time pressure is combined with risk. Their studies show that the decision maker becomes more cautious and adopts risk-avoiding behaviour, paying attention to avoid losses. In these situations, strategies like loss avoidance, elimination by aspect and fast-and-frugal search are used. In other words, the human focuses only on what is believed to be the most relevant portion of the information. Whilst this can lead to gross errors, in certain situations it helps by eliminating non-essential information.

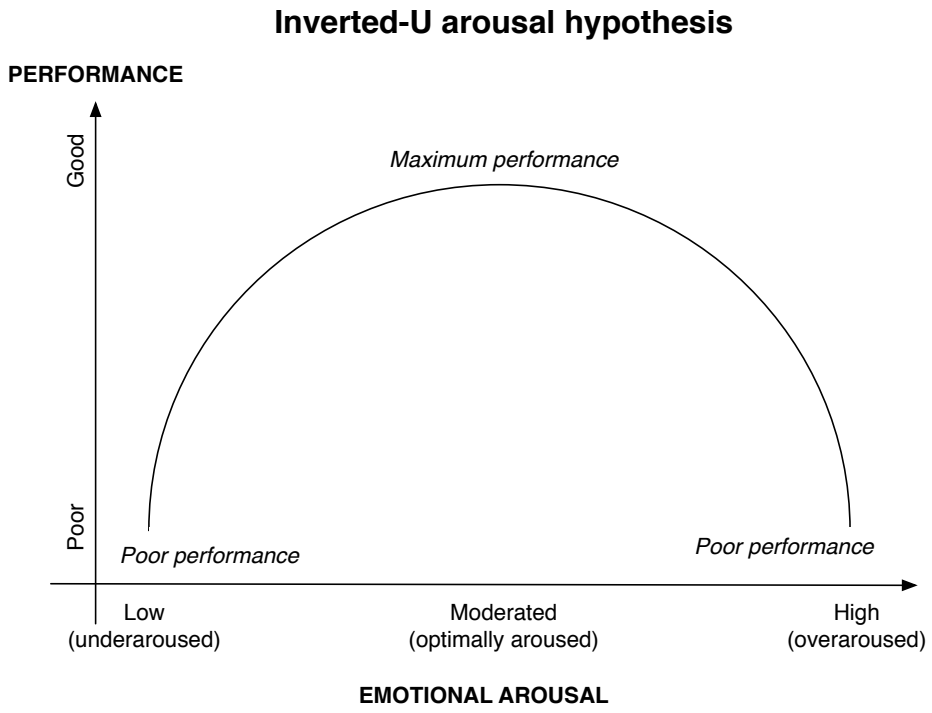


Figure 4.1: Inverted-U arousal-performance model [Yerkes and Dodson 1908].

Kontogiannis and Kossiavelou [1999] present slightly more negative results and demonstrate how stress restricts cue sampling, decreases vigilance, reduces the capacity of working memory, causes premature closure in evaluating alternative options, and results in task shedding. Similar results, relative to both stress and frustration, are found by other researchers [Zakay and Wooler 1984; Hutchins, Kelly and Morrison 1996; Gaillard 2008].

As well as the controversy about the effects of stress and frustration on human decision behaviour, Dekker and Hollnagel [2004] give an even stronger criticism of the overall applicability of the Inverted-U arousal-performance model, pointing out that it lacks precision and falsifiability.

In summary, severely stressful ADR situations could provoke frustration, but not necessarily. Whilst stress has an effect on decision behaviour which is not necessarily negative, frustration seems to have a negative influence in all cases. In the context of this research it is worth investigating the possibility that frustration effects pilots during ADR decisions characterised by severely stressful conditions. If this were the case, it would be interesting to address the issue of mitigating such a negative effect.

Mitigating the effects of frustration

Rahman [2007] proposes a set of prescriptive recommendations, "the laws of High Velocity Human Factors", which focus on emotional modulation of cognition of mission critical personnel in nonequilibrium conditions. The seven HVHF laws proposed by Rahman are briefly described here:

1. *Law of Relevance*: only provide information relevant to the event that can be used to dia-

gnose and resolve the situation.

2. *Law of Acceptance*: provide information in a format that can be processed by the human agent given the diminished cognitive capacities due to emotional arousal.

3. *Law of Transparence*: technology should not become a barrier to information that can be directly perceived in the immediate environment.

4. *Law of Clairvoyance*: technology, where possible, should assist the human agent to predict the immediate future course of an event.

5. *Law of Absoluteness*: critical functions, such as emergency call placement, should have their own dedicated control elements that are accessible and operable in an instant.

6. *Law of Intelligence*: technology should be smart enough to take over operations when the agent is overloaded with other more important tasks.

7. *Law of Reliance*: technology should be fail-safe & fool-proof and should accommodate human-interaction errors caused due to high stress.

The laws have a very wide scope, covering many aspects of cognitive system engineering, most of which are beyond the interests of this PhD programme. This thesis focuses on only a few aspects of the laws, which are elaborated further in Section 4.2, when enough material will be presented to shape the decision support information for ADR in more details.

For the moment, the following claim is advanced:

Claim 5 *Heightened states of stress during ADR decisions can possibly lead to frustration. The negative effect of frustration would be mitigated by providing pilots with* effective *decision support information.*

We claim that effective decision support information should increase pilot perception of control during ADR. The meaning of 'effective decision support information', as used here, is discussed in detail in Section 4.2.

4.1.4 Framing effect and ambiguity aversion

The first step in the decision making process is perception. According to normative decision theories, the way information is presented and *perceived* is irrelevant. This principle is known as **descriptive invariance**.

In real situations descriptive invariance does not hold. Depending on the way the information is *presented* to the human and consequently perceived, decision biases emerge and modify the decision maker behaviour. This phenomenon is known as the **framing effect** [Payne et al. 1998]. The theory in support of the framing effect is robust insomuch as neurological evidence of its emergence has been recently documented [Gonzalez et al. 2005; Weller et al. 2007].

Given the relevance of the framing effect, it makes sense to investigate whether pilots are susceptible to the way ADR decision support information is presented and, if this were the case, to what extent.

The framing effect is not only related to the graphical representation of the information, which is beyond the scope of this thesis, but also refers to the way decision options are *semantically* described. In order to better describe the concept, we first introduce the **concreteness principle**. Slovic [1972] asserts that "a decision maker tends to use only the information that is explicitly displayed in the stimulus object, and will use it only in the form in which it is displayed". When this principle holds, preferences for identical options with different reference points can reverse [Fischhoff et al. 1974].

For instance, in the context of ADR, the system could equally tell the pilot that a certain sensor reading is *reliable* at 70% or that the sensor reading is *uncertain* at 30%. According to the framing effect, the decisions of the pilot would be influenced by giving them sensor readings in terms of reliability or uncertainty, but how exactly?

The **ambiguity aversion** (also know as **uncertainty aversion**) is a decision making attitude which is closely related to the framing effect. According to the theory, when a human has to choose between two options characterised by different degrees of uncertainty, the contrast makes the more uncertain option less attractive (or the less uncertain option more attractive) [Heath and Tversky 1991].

Fox and Tversky [1995] cite an exception to the ambiguity aversion principle: ambiguity aversion applies only when options are *compared*; if options are evaluated singly, humans value them impartially. In fact, when evaluating an uncertain event *in isolation*, humans try to assess its likelihood without paying too much attention to second-order characteristics such as vagueness or weight of evidence. This principle is referred to as the principle of **comparative ignorance**. Tversky et al. [1988] notes that the comparative ignorance effect violates the **principle of procedure invariance**, which posits that strategically equivalent elicitation procedures should produce the same preference order.

In typical ADR decisions pilots always have at least two options to compare and choose from, at least one new configuration proposed by the system and the current one. The pilot can choose to decline the execution of ADR and leave the state of the system unchanged. As a result, the exception of comparative ignorance principle will not to be considered in this context.

In the gambling domain, Schie and Pligt [1995] add that the emphasis on positive features promotes greater risk-seeking decisions in both the gain and loss domains and emphasis on negative features promotes greater risk aversion in both domains. If this also applies to the context of ADR decisions, reliability (a positive feature) would make pilots more comfortable accepting reconfiguration advice than uncertainty (a negative feature).

We argue that the effect of ambiguity aversion should be particularly evident in ADR decisions because they are safety-critical and risky. Risky decisions are known to be subject to the effect of **loss aversion**, which manifests itself as a tendency to favour the *status quo* over change [Tversky and Kahneman 1984; 1991]. Loss aversion is a well-documented decision bias, supported by recent investigation at the neurological level [Tom et al. 2007].

Levin et al. [1998] distinguish three forms of framing effect:

- *Attribute framing effect*. This occurs when evaluations of an object or event are more favourable if a key attribute is framed in positive rather than negative terms.

- *Goal framing effect*. This occurs when a persuasive message has different appeal depending on whether it stresses the positive consequences of performing an act to achieve a particular goal or the negative consequences of not performing the act.

- *Risky choice framing effect*. This occurs when willingness to take a risk depends on whether the potential outcomes are positively framed (for example in terms of success rate) or negatively framed (for example in terms of failure rate).

The messages generated by the DSS being proposed in this research are not designed to persuade the pilot towards a specific decision alternative but to allow an informed, impartial choice. As a consequence, the goal framing effect is not addressed here; however, both attribute and the risky choice framing effect require attention.

Each reconfiguration alternative generated by the system is associated with a degree of 'reliability *vs* uncertainty' which is calculated at run-time on the basis of the quality of the information coming from the on-board sensors. The uncertainty value is an attribute of each decision alternative (attribute framing effect) and it also contains information about the potential risks hidden behind each option (risky choice framing effect). If the framing effect holds for ADR decisions, it can be expected that the pilot is more comfortable accepting a configuration when it is based on sensor readings that are, for example, '70% reliable' than '30% uncertain'. In other words, they would be put off by presenting the information associated with the uncertainty embedded in the inference process that generated them. We speculate that the by-product effects could be any of the following, a reduction of the trust in the system, emergence of indecision, frustration, increased decision time and potential selection of wrong options.

Claim 6 *The pilots' decision behaviour during ADR would be subject to the framing effect; more specifically, presenting ADR information in terms of its* reliability *instead of its* uncertainty *would make pilots more comfortable accepting to apply the proposed configuration.*

4.1.5 Complacency

The limitations relative to storage and retrieval of information from long-term and working memory have been discussed on several occasions so far. The argument is re-considered here, in order to make new claims concerning ADR decisions.

It has been proven that, as humans often cannot remember details about the past, they are inclined to accept misinformation as accurate when it is provided by an agent they consider more knowledgeable and/or authoritative, because they lack further memory [Skitka et al. 1999; Smith and Kosslyn 2007]. This is risky behaviour during safety-critical decisions: if the information generated by the system is not correct, the consequences can be catastrophic.

There is a robust body of evidence which shows that in highly automated processes, operators may show signs of excessive trust in, and reliance on, an automated decision support system [Lee and Moray 1992; Muir 1994; Layton et al. 1994; Endsley 1996; Parasuraman et al. 1996; Parasuraman and Riley 1997; Dzindolet et al. 2003; Parasuraman et al. 2008]. There is also a history of aviation accidents linked to an over-reliance on the deck automation.

Accident 4.1 *On the* 26^{th} *June 1988 a brand new Airbus A320-100 crashed near to Mulhouse-Habsheim airport (falling into neighbouring wood) in the midst of a low-flying manoeuvre over the runaway. Three out of the 136 passengers died, 50 were severely injured. Reports acknowledge that "the A320 has new features which may have inspired some overconfidence in the mind of the Captain" [Investigation Commission of Ministry of Transport - France 1989] (page 60).*

Accident 4.2 *On the* 14^{th} *February 1990, Flight 605 (Airbus A320-231 aircraft) crashed on its final approach to Bangalore airport, killing 92 people. On final approach, the aircraft descended below the normal approach profile and kept descending until it struck the boundaries of a neighbouring golf club. Ministry of Civil Aviation - Government of India [1990] (page 39) reports that "a false sense of faith has been reposed on this system".*

This phenomenon has been defined as **Automation-Induced Complacency (AIC)**, an uncritical reliance on the automation, resulting from an inappropriate high trust in the system's reliability [Bustamante et al. 2009]. AIC was first studied in the context of cockpit automation [Billings et al. 1976; Parasuraman et al. 1993], but later research also analysed it in other domains, like the development of decision support systems [Smith and Geddes 2003].

Memory limitations are not the only cause for the emergence of AIC and are probably not even the most influential. A relevant role is played by the phenomenon of **perceived animacy**, which refers to the act of viewing automation as an independent agent [Woods 1996]. Examples of the emergence of animacy are found in commercial airline cockpits where pilots during operation are found to ask questions about flight management automation such as, "What is it doing?" and "Why did it do that?" [Cummings 2006]. Evidence confirms that when humans solve a task in collaboration with a computer, they tend to hand the responsibility to the computer, which they tend to consider a smarter "team member" [Skitka et al. 1999; Nass et al. 1996].

Parasuraman et al. [1993] bring emotions into the discussion, another source of AIC. They maintain that the tendency towards over-reliance on automation that has functioned safely in the past might be heightened by positive affect. Furthermore, decision makers are implicitly reluctant to seek out any information that might interfere with the positive mood. However, as some authors argue (see Section 4.1.3), expert decision makers should be less influenced by emotions over phenomenon like AIC.

Complacency is found to be closely related to SA. Having empirically compared full automation and interactive approaches for air traffic control tasks, Dao et al. [2009] show that SA in the fully automated condition is weaker than in the interactive condition, because of the complacent attitude of the pilots. The issues that can arise from loss of SA during ADR have been already elaborated in Section 3.2.2.

The counterpart of AIC is also documented in the literature and its risks during safety-critical processes are highlighted. More specifically, several authors have shown that experience with a system, or a negative attitude toward modern technology, can lead pilots not to use information provided by the system or follow its recommendations, even when it would have been in their best interests to do so [Lee and Moray 1992; Muir 1994; Riley 1996; Endsley and Kiris 1995; Endsley 1996; Parasuraman and Riley 1997; Dzindolet et al. 2003; Lee and See 2004].

On the basis of the material discussed so far, it is important for this thesis to understand whether pilots are complacent towards the reconfiguration advice of the ADR decision support system or not. To the best of our knowledge, this thesis proposes the first architecture for decision support for ADR; therefore, no pilot has previously had the chance to try a system of this type or to construct any sort of trust in this type of technology. This makes the investigation even more interesting, as the complexity and novelty of the problem could lead pilots to shun the critical decision and leave the responsibility to the system, especially in highly demanding situations. This topic is resumed in Section 4.2.4, after more arguments are introduced.

If AIC is found to emerge during ADR decisions, then ways to mitigate this phenomenon should be investigated. Pilots should always be able to verify *what* the reconfiguration system is suggesting and *why*, asking for example why data bus redundancy is reduced to zero in the new configuration. Increased transparency should reduce AIC.

The following speculative claim requires empirical investigation:

Claim 7 *Pilots would be subject to Automation-Induced Complacency during highly autonomous ADR. This phenomenon would be mitigated by increasing the degree of pilot involvement in the process.*

4.1.6 Conclusions

This section generated a first set of claims aimed at defining a permanent user profile for a DSS for ADR decisions that takes into account cognitive aspects of the problem. The model is not meant to be complete; the focus of this thesis is only on aspects that can either impair or improve the safety of the process and that can be empirically and realistically verified in the context of a Ph.D. programme.

It is reminded that the aim of this thesis is not to discredit the role of the human in the control of complex systems in unstructured environments, on the contrary, this thesis advocates the unique human capability to cope with unusual and unexpected situations, a capability which is missing in modern automated system control technology to the same extent as it is present in humans. Indeed, the analysis of the factors that can affect the human during ADR decisions is performed in order to design decision support technology capable of mitigating the effects of undeniable weaknesses of the human with the overall aim of achieving effective human-machine co-operation, exploiting the strengths of both of them.

The next section uses the material presented so far to define the content of effective, tailored decision support information for ADR decisions. A number of arguments placed on hold in this section are resumed and further elaborated on. This represents the first step of the "content determination" phase of the development of the proposed decision support framework (page 50). The claims put forward here and in the next section are empirically evaluated in Chapter 6; all the claims will be summarised at the end of the chapter .

4.2 Decision Support Information Content

According to the **support theory** of decision making, which complements the prospect theory (Section 2.2.2.2), humans combine beliefs from multiple sources and based on several underlying cognitive capacities into a summary of "strength of belief" [Tversky and Koehler 1994]. The theory posits that the strength of belief increases for decision options that are "unpacked" into more explicit disjunctions. For instance, consider the comparison of the following two options:

1. `Configuration-A` is better than `Configuration-B`
2. `Configuration-A` is better than `Configuration-B` *BECAUSE* EFS is active *AND* EFS is required during manual landing *AND* current mode is 'manual landing'.

According to the theory, the latter option would receive a higher strength of belief than the former.

The material discussed above casts light on the complexity of ADR decisions. The complexity is such that usually higher-order information (e.g., the consequences of applying a certain configuration) is not directly available to the pilot. Valuable help would come from an intelligent system that would "unpack" information for pilots and present it in a readily understandable format. It must be noted that this does not necessarily mean *more* information, rather the *right* information, as if the baseline information is unpacked in too many disjunctions, the pilot could be overloaded with information and the decision support system would become counterproductive.

The question of how the decision support information should be "unpacked" and framed, in order to provide effective support to pilots during ADR, does not have a straightforward answer. In Section 2.2.2.3, having briefly introduced the five most cited descriptive cognitive models, it was observed that mental simulation is a critical component for all of them. The approach to shaping decision support information taken in this thesis focuses on fostering this process. We argue and verify empirically that this approach has a positive effect on several decision biases described in the previous section.

4.2.1 Mental simulation

Humans quite often rely on mental simulation in their day-to-day life. Evidence shows that people reason by forming and transforming mental representations of possible actions and "observing" the consequences of those actions. The fact that imagery and perception share most of the same neural mechanisms supports this assertion [Smith and Kosslyn 2007].

It has been demonstrated since the early 1980s that mentally simulating an intended action without actually performing it (**motor imagery**) has a positive effect on performing that action afterwards [Feltz and Landers 1983; Woolfolk et al. 1985]. Mental simulation particularly seems to play a critical role in experts' decisions. Building on a naturalistic decision making study [Lipshitz and Shaul 1997] in which the decisions of experts and novices are compared, Elliot [2005] infers that:

- experts require more information to identify goals;

- experts expectations are more likely to be violated;
- *experts engage in mental simulation more often*.

Calderwood et al. [1987] provide the empirical support for the third implication, which is particularly relevant to this thesis. In a very recent paper on power system operators, Greitzer et al. [2010] show further evidence in support of this idea. They report that experienced power system operators perform a mental simulation by first retrieving relevant mental models from long-term memory; they then use them to run a mental simulation to check if there is consistency with the cues that are being observed.

Greitzer et al. acknowledge that quantifying the effects of the operators' actions when the system is in unusual operating condition can be very difficult. They maintain that *the use of a simulation or contingency analysis tool may help in constructing correct mental models and favour the mental simulation process*.

The ideas of Greitzer et al. can be extended with the conclusions of a study by Artman [1999*a*] who argues that experts and novices may have the same problem solving strategies available, including mental simulation, but experts can use them more effectively because of their superior knowledge base and perceptual advantage. Klein [1998] interprets poor mental simulation capabilities as a cause of decision errors. More specifically, Klein defines three main causes of decision error:

- Lack of experience of the decision-maker.
- Lack of information.
- *Poor mental simulation*.

Interestingly for this study, Endsley [1995*c*] highlights the link between mental models and situation awareness (Figure 4.2). Later studies have brought further evidence in support of the model (e.g. Artman [1999*b*]; Brehmer [1990]; Elliot [2005]) which is now widely accepted in the literature.

In Endsley's model, the process of constructing mental representations relies on good SA (discussed in Section 3.2.2); in turn, mental simulation (projected state) relies on both good mental models and good SA.

In order to favour the construction of good mental models, good SA and good mental simulation, and with reference to the support theory, this thesis advances the following claim:

Claim 8 *Decision support information for SCMS reconfiguration should be "unpacked"/framed so as to make the following cues readily understandable to the operators, for each reconfiguration option:*

- ***Explanations****:* why *the system is making a specific suggestion.*
- ***Implications****: (*what-if *type of information) what the consequences of applying a specific configuration are.*

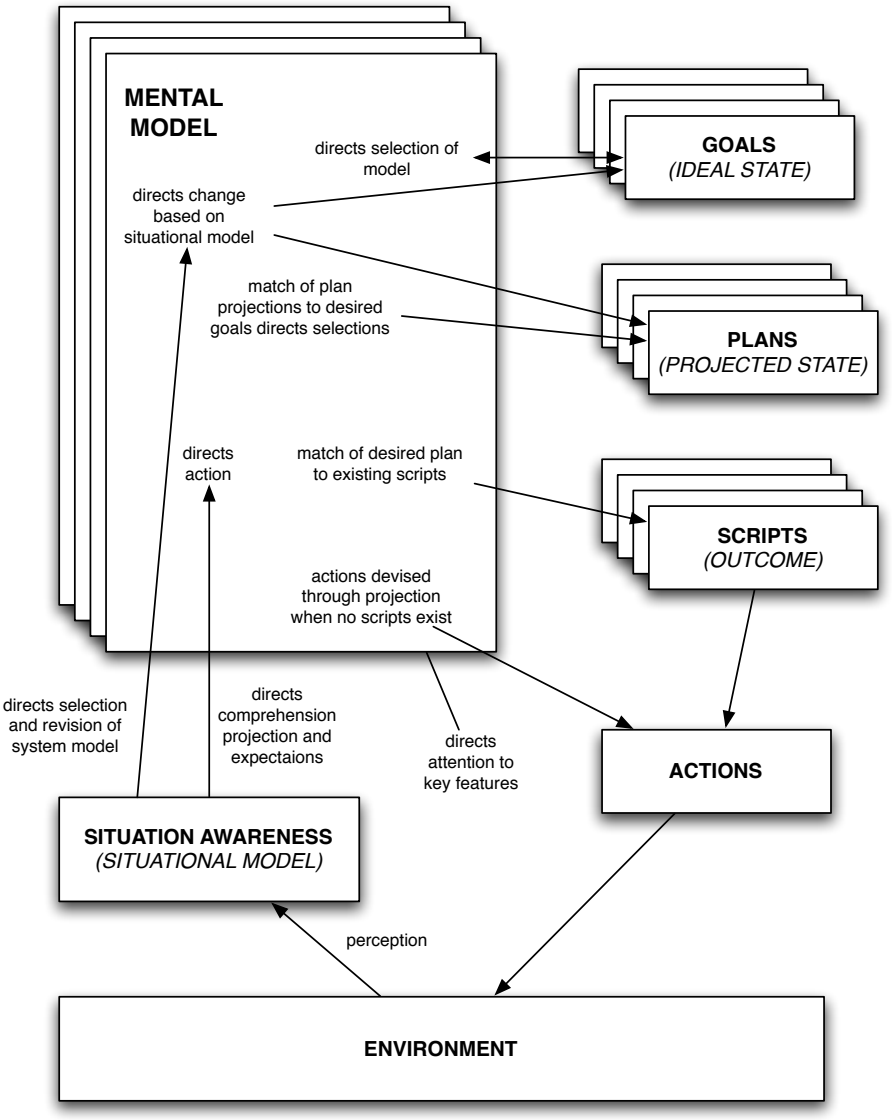


Figure 4.2: Relationship of goals and mental models to situation awareness (adapted from Endsley [1995*c*]).

- ***Reliability****:* how reliable *the advice generated by the system is.*

The rationale for the necessity of each type of information is provided in the following sections.

4.2.2 Explanations

Some situations require the SCMS operators to perform methodical actions for which they have been extensively trained (for example performing a manual landing manoeuvre in the case of an aircraft pilot). On the other hand, complex reconfigurable systems, operating in unstructured environments, sometimes require the operators to face new, unexpected problems. An 'avionics reconfiguration following an unexpected fault with an uncertain cause is a fitting example of the latter case.

Studying similar complex scenarios in the aviation field, Besnard [2004] notes that such trouble-

shooting activities, which are part of the more global objective of piloting the aircraft, "involve the construction of an *explanation* in real-time". Besnard adds that there are several factors, amongst which limited cognitive resources, time pressure and confirmation bias (a tendency to only search for information that supports the focal or preferred hypothesis [Klayman and Ha 1989]) can lead pilots to the construction of erroneous explanations.

Generating wrong explanations of facts (e.g. the reasons why avionics `Configuration-A` should be applied instead of `Configuration-B` in order to mitigate the effects of the fault which has just happened) could have catastrophic consequences in safety-critical decisions. Explanations are used by humans to construct a 'story' of the unfolding events through mental simulation; they help to refine mental models and establish relations between facts [Zsambok et al. 2002; Greitzer et al. 2010]. If these relationships are erroneous, the probability of making wrong inferences increases.

Although the explanations generated by the decision maker are decisive in making correct choices, their construction is particularly prone to errors in real, complex, time-critical decisions, especially without any bespoke support. Kaber et al. [2001] bring this problem back to the issue of *misinterpretation*, on the basis of the well documented inability of humans to assess the intention of the computer system [Mosier and Skitka 1996; Bubb-Lewis and Scerbo 1997]. The claim made in this thesis is that providing pilots with an explanation of the reconfiguration recommendations produced by the system would foster the construction of better and richer mental models and, as a consequence, the room for misinterpretations would be reduced.

Section 4.1.1 showed that perception of *risk* has its roots in a poor understanding of the functioning of the system. Intuitively, explanations of the inferences generated by the system should improve pilot understanding of the situation and, as a consequence, reduce the perception of risk. We argue that explanations should also mitigate the effect of the *ambiguity aversion* bias (Section 4.1.4). By explaining why the system is making each suggestion, the ambiguity associated with the suggestions should be reduced. This should result in a more objective evaluation of the decision alternatives and, as a consequence, the decision accuracy should improve.

Explanations should foster both mental simulation and story construction by enriching the perception of the elements in the environment (see Figure 4.2). Evidence shows that both mental simulation and story construction help people to stay alert to the dangers of upcoming events [Cohen and Freeman 1997] and organise the pieces of information in a hierarchical structure determined by their *importance* [Pennington and Hastie 1988]. With reference to his model of Figure 4.2, Endsley [1997*b*] maintains that "building a story corresponds to increasing the operator's level of SA from Level 1 through Level 3". The positive effect on SA building is another important motivation for providing pilots with system-generated explanations; Section 3.2.2 has already discussed the effects of loss of SA on ADR decisions.

The emergence of *complacency* should also be limited by explanations. Section 4.1.5 discussed the fact that humans are inclined to accept misinformation as accurate when it is provided by an agent they consider more knowledgeable and/or authoritative. By increasing the transparency of the inference process, explanations should reveal potential issues with the line of reasoning of the system and help pilots to discard wrong inferences.

In summary, with reference to the decision biases discussed earlier in this chapter, explanations should mitigate the negative effects of the following factors: perception of risk, ambiguity aversion, loss of SA and complacency. There are also other general benefits of explaining the reasoning of an automated system which would provide further value to the approach proposed by this thesis.

In the previous section, in the context of support theory, it was mentioned that "unpacking" information into more explicit disjunctions increases human confidence in the information. Koehler [1991] shows that **confidence**—the overt expression of a likelihood—has a positive influence on decision behaviour and is increased by explaining why a possibility might be true. More specifically, Koehler proposes that firstly, explanations cause changes in the way the problem is perceived, by determining which aspects are made more important; secondly, they effect the interpretation of evidence; and, thirdly, they effect the direction and duration of the search. Altogether, these mechanisms are found to increase the decision makers' confidence in their decisions.

In line with Koehler, Hogarth and Kunreuther [1995] empirically demonstrate that explanations increase confidence by allowing the decision makers to justify decisions to themselves.

Empirical evidence is also provided in an earlier study by Koriat et al. [1980], who show how "reason generation" (explaining) influences confidence in a positive manner.

Another documented benefit of explanations is their potential to mitigate the effects of the **confirmation bias**, a tendency to only search for information that supports the focal or preferred hypothesis [Snyder and Swann 1978]. According to the theory, decision makers systematically favour the decision alternative they are looking for. Explanations should bring justifications for other alternatives to the attention of the decision maker, leading to a more impartial decision. In other words, explanations should increase the degree of *plausibility* of options which, because of the confirmation bias, would not be appropriately considered.

Interestingly, in his investigation into the potential of explanations, Koehler [1991] provides an 'associative memory based' interpretation of their importance during a decision making process. The author reports that "the effects of explanation might be interpreted as a variant of the generation effect observed in many memory studies (e.g., [Slamecka and Graf 1978]) that have shown a recall advantage for self-generated items over presented items [...] By this account, explaining or imagining a possibility makes it easier to recall supporting facts retrieved from memory when the actual judgment is made".

The importance and influence of explaining the logical reasoning of on-board systems have also been recognised by aviation authorities. For instance, after the crash of the Comair Flight 3272 (Accident 3.1), National Transportation Safety Board [1997] reports the followings:

- "Aeronautica Civil urges the FAA to evaluate the curricula and flight check requirements used to train and certificate pilots to operate FMS[1]-equipped aircraft, and revise the curricula and flight check requirements to assure that pilots are fully knowledgeable in the logic underlying the FMS or similar aircraft computer system before being granted airman certification to operate the aircraft";

[1]Flight Management System - it is a specialised computer system that automates a wide variety of in-flight tasks, reducing the workload on the flight crew.

- "the workload could decrease with explanations: such partially understood logic may partially account for the finding that use of the FMS often increases workload during periods of already high workload".

This thesis also acknowledges the risks and potential drawbacks of explaining ADR options. Given the degree of influence that explanations have on decision behaviour, different explanations could provoke different degrees of confidence and plausibility and, as a result, could lead to the opposite effect of mitigating the confirmation bias. Pilots could be led to choose a sub-optimal or even wrong option which was supported by a misleading explanation. This argument introduces the issues of *correctness* of the decision support information and *persuasiveness* of a DSS, which leads back to the problem of complacency. This issue is empirically investigated in the series of experiments of Chapter 6.

Intuitively, in order to limit the emergence of this effect, explanations should be equally framed for each decision alternative and they should have a similar structure, which the pilot can readily understand. This supports the design choice made in this research, not to use rich, natural language framing for the ADR decision support information.

4.2.3 Implications

Mental simulation is the projection of the current state of the system *into the future*; this process allows the operator to "see" the implications of current potential actions and evaluate their applicability and/or optimality.

As mentioned in Section 4.2.1, mental simulation has a positive effect on subsequent decisions. However, Parasuraman [2000] collects a considerable number of studies that highlight the difficulties humans have with simulation when interacting with an automated system. Essentially, humans are found to be less aware of the effects of their decisions in system states when those decisions and consequent actions are also under the control of another agent (whether that agent is an automation or another human) than when they make the decisions and implement them completely alone. In addition, Koehler [1991] shows that the effect of simulation and estimation of likelihood are mediated by the ease with which this imagination process takes place (cf. [Tversky and Kahneman 1973; Kahneman and Tversky 1981]).

Regardless of the implied difficulties, evaluating the implications of decisions is critical to the reliability of the ADR process. For instance, if pilots are provided with two applicable configurations and they do not manage to foresee the consequences of applying each of them, they could incur the risk of discarding the one whose implications are less obvious, even if it is optimal.

We argue that providing system-generated implications of each reconfiguration option would reduce room for uncertainty. Therefore, it would have a positive effect on the perception of risk (Section 4.1.1).

We also argue that system-generated implications should mitigate certain biases induced by time pressure. As mentioned in Section 4.1.2, in situations of time pressure, humans tend to screen the decision alternatives for violations of the minimal level of acceptance on one or more attributes. In a safety-critical scenario, the consequences of a decision represent the most important attribute

under evaluation, much more important than the attributes related to performance, for instance. Providing pilots with readily-available implications of each decision should release the human from the task of constructing them, which should make violations of pilots' levels of acceptance more evident and, as a result, should decrease both cognitive workload and the effect of time pressure.

The construction of the implications for each decision alternative in real-time is an extremely demanding task during ADR. On the basis of the material presented in Sections 3.2 and 4.1.3 concerning cognitive demand, problem complexity and emotion-based biases, frustration is most likely to emerge when trying to build the implications of each decision alternative given the peculiar characteristics of the decision scenario. Providing pilots with system-generated implications would increase their sense of control over the process and, as a result, reduce the impact of frustration on their decision behaviour. This conclusion is also in line with the 'law of clairvoyance' (Section 4.1.3.2) from the HVHF laws proposed by Rahman [2007] to avoid emotion-based biases: "technology, where possible, should assist the human agent to predict the immediate future course of events".

Finally, by providing an insight into the consequences of each alternative, implications should intuitively reduce the emergence of complacency (Section 4.1.5).

4.2.4 Trust

The literature has extensively proven the role of trust in the human reliance on automation (see [Dzindolet et al. 2003] for a review). When a DSS for ADR generates recommendations, explanations and implications of reconfigurations, it is actually making inferences that are critical for the lives of the crew members. If the operators do not trust the logic of the system, it is probable that they will not be comfortable basing their decisions on its suggestions. This could break the co-operation between the human and the machine, with severe consequences in a safety-critical environment.

Incidentally, it is not surprising that, during the 1980s, a strange series of accidents related to the introduction of automation on-board civil aircraft occurred in France, the country in which at that time protests and strikes against the introduction of glass cockpits—provoked by pilots' *mistrust* of the automation—had been stronger than anywhere else in the world [Amalberti 1999].

Muir [1994] argues that trust *declines rapidly* when the system makes errors but *increases again slowly* as the system keeps performing, reducing the number of errors. Thus, as interaction with the automation takes place, the level of trust is expected to ebb and flow.

In a study on military strategy decisions supported by a decision aid, Dzindolet et al. [2003] found that humans deem the aid trustworthy when they know a little about the automated decision support system. When the system commits mistakes, humans move "from an unjustified high level of trust in automation to an undeserving low level of trust and rampant disuse". However, other studies in the aviation field have found contrasting results. Pilots seem to have a strong mistrust of automated aids when they are first introduced into the cockpit and they know little about them [Olson and Sarter 2000].

Given the contrasting positions concerning human trust in this type of systems in the literature,

it is interesting to investigate whether pilots are going to trust the recommendations of a DSS for ADR proposed in this research.

An obvious observation is that the *reliability* of ADR advice would increase the trust in the system. In order to observe an increase in trust due to reliability, the pilot must use the system for a certain amount of time. This work explores a new technology which has not yet been introduced on any aircraft; therefore, this type of information is not available at present.

Unrelated to the exposure time is the influence of *a priori* knowledge of the *processes* underlying the inferences generated by the aid on human trust. In line with the literature referenced above, Lee and Moray [1992] show that without a good understanding of the processes that a decision aid uses to make inferences, the operators are likely to deem the DSS untrustworthy from the beginning. In particular, they find that an operator's use of automation to control a simulated manufacturing plant was directly related to his or her momentary trust. In a subsequent study, Lee and Moray [1994] add empirical evidence which shows that the operators use the automation if their trust in it exceeds their confidence in their own ability to control the plant, but otherwise they choose manual control.

Pu and Chen [2006] recognise the importance of explanation interfaces to improve user performance and build user trust in the automated system. They put forward a set of guidelines for the development of explanation interfaces specifically aimed at trust building. This is another argument in favour of the inclusion of explanations in the ADR decision support information set; *we expect that explanations would increase pilot trust in the DSS for ADR.*

In conclusion, implications are one of the most important 'chunks' of information for ADR decisions; however, ADR suggestions supported by risky or unacceptable implications could provoke mistrust. Mistrust could be mitigated by the availability of explanations, which provide an insight into the assumptions at the root of the inferences made by the system. Certain risky implications (e.g. switching off the auto-pilot) could apparently hide an error in the inferences made by the system; the explanation of how the system reached that conclusion could convince the pilot that the configuration suggested is actually the only right option given the current circumstances.

The material presented in this section leads to the following claim:

Claim 9 *Pilot trust in a DSS for ADR would decrease when the system makes (apparently) unacceptable inferences. This phenomenon would be mitigated by providing pilots with an explanation of the inference process.*

4.2.5 Uncertainty

"An absence of information is not the same as information about an absence"
from 'Consciousness' (page 88), Susan Blackmore, 2010

In broad terms, uncertainty has a two-fold nature:

- **Aleatoric uncertainty**: results from the fact that a *system can behave in random ways*. It is also known as stochastic uncertainty, irreducible uncertainty, variability, or Type A uncertainty.

- **Epistemic uncertainty**: results from a *lack of knowledge about a system* and is the property of the system engineers who design and develop the automation. It is also known as subjective uncertainty, ignorance, reducible uncertainty, or Type B uncertainty.

Since SCMS of the type considered in this thesis operate in unstructured environments, aleatoric uncertainty is non-negligible in the design of on-board automation. The fact that the operating environment is unstructured and that modern, reconfigurable SCMS such as next generation aircraft avionics are extremely complex machines also makes epistemic uncertainty non-negligible. In fact, it seems highly improbable (to use an euphemism) for engineers to acquire complete knowledge of the system and manage to forecast all the possible unexpected events that could happen during its operation.

The result is that on-board modern aircraft, pilots must accept the uncertainties hidden in the logic guiding the automated processes and rely upon probabilistic information to evaluate what they cannot access directly [Wickens and Flach 1988]. Section 4.1.1 provides a description of the effects of uncertainty on the SCMS dynamic reconfiguration process. Here the analysis returns to the argument. It is interesting for this research to investigate what the effect of providing pilots with *figures about the uncertainty* embedded in ADR inferences would be without leaving them to guess those figures.

Finance studies demonstrate that providing decision makers with figures on the uncertainty embedded in each decision option has a positive effect on the decision behaviour because decision makers use this information to hedge decisions away from large losses [Reckhow 1994].

Aviation and military studies reveal compatible results. Banbury et al. [1998] conducted an experiment asking military pilots to respond to a machine-identified target with a 'shoot/no shoot' decision. The experiment revealed that decision making behaviour changes when the system explicitly identifies a friendly aircraft as a secondary target. The authors noted that prior willingness to fire on a target with a high level of uncertainty disappeared. Furthermore, the decision time was also found to be influenced by the uncertainty of the targets.

With respect to decision time, similar conclusions are reached in a later study on a DSS for aircraft anti-icing, in which Sarter and Schroeder [2001] found out that accurate information from the decision aid led to improved handling of the icing encounter. However, when inaccurate or uncertain information was presented, performance dropped below that of the baseline condition.

The availability of uncertainties embedded in automated inferences seems to make the human more 'cautious' with the automation and spend more time collecting information before acting. However, this could be difficult to assess without any automated support in a complex system that handles information from various sources (e.g. modern sensor data-fusion) and under time pressure.

In this regard, in a detailed analysis of the U.S.S. Vincennes accident (3^{rd} July 1988), Gruner [1990] reports that officers and system operators "could not make better decisions because they did not have time to confirm or deny the information *uncertainties* presented to them". On that occasion the Command-And-Control team had three minutes and forty seconds to make a decision, including the time to perceive and interpret sensor data and make judgements [Roberts and Dotterway 1995]. Giving a parallel to ADR decisions, the task of figuring out how reliable/uncertain

the information generated by the system is (which is based on probabilistic processing of sensor data) could overcome pilot cognitive capabilities in a typical ADR scenario. Providing pilots with system-generated figures about the uncertainty (when available), embedded in the automated inference, should reduce their workload in situations in which the information available to the system is not fully reliable (e.g. sensors readings).

Dzindolet et al. [2003] provide empirical evidence that decision makers who are given a reason why the aid might make a mistake are likely to trust the decisions of the aid more and are more likely to rely on the aid than decision makers who are not provided with this type of information. The authors generalise this and previous studies, arguing that "optimising the performance of the automated aid will not be successful in optimising human-computer team performance. Understanding the processes that humans use to determine whether or not to depend on their automated aids will help to improve performance of the human-computer team".

On the other hand, uncertainty figures represent an additional 'chunk' of information to process in real-time and they could even provoke indecision and mistrust, hence, their theoretical benefit could actually become counterproductive. It makes sense to investigate the effect of reliability figures on pilot trust and performance during ADR decisions.

Another factor that requires attention is the way uncertainty information is framed. This section uses the terms uncertainty and reliability about the information directed to pilots interchangeably, referring to the same concept but from opposite ends of the same scale; full reliability implies no uncertainty, full uncertainty implies no reliability. The concreteness principle asserts that "a decision maker tends to use only the information that is explicitly displayed in the stimulus object, and will use it only in the form in which it is displayed" (see Section 4.1.4). Furthermore, the decision maker tends to be more risk averse when negative or uncertain decision support information is provided [Schie and Pligt 1995]. These arguments lead to the speculation that pilots would "feel more comfortable" accepting a suggested reconfiguration accompanied by high reliability figures, instead of low uncertainty figures. Reliability is a positive feature, uncertainty is negative (see the discussion about the influence of emotions on ADR decisions in Section 4.1.3).

One last argument worth considering is that when testing a hypothesis, decision makers tend to look for features that are extreme: they find it easier to process alternatives that are either very likely or very unlikely under the focal hypothesis [Skov and Sherman 1986]. This argument seems to suggest that pilots would find suggestions characterised by 'medium reliability' more complicated to process than low or highly reliable options.

To summarise, reliability figures should mitigate the effects of the following decision biases: trust, time pressure and complacency. The following speculating claim is made to motivate our empirical tests:

Claim 10 *Providing* reliability *figures would influence pilot decision making performance in the following ways:*

- *the framing effect would emerge when providing pilots with reliability or uncertainty terms; more specifically, pilots would feel more comfortable applying a configuration associated with high reliability than with low uncertainty;*

- *evidently wrong ADR suggestions, when associated with low reliability, would be more easily spotted and avoided than without any reliability figures;*

- *low and high reliability options would both be easier to process than medium reliability options, i.e. the decision time would increase with medium reliability options.*

4.2.6 On the number of alternatives

The previous sections define the characteristics of the decision support information that could help in mitigating the effects of a number of decision biases that are likely to emerge during ADR. So far, not enough attention has been paid to how many reconfiguration alternatives pilots should be provided with during a typical ADR.

Some issues about problem complexity and cognitive limits have been already discussed in Section 3.2. These arguments led to the conclusion that some degree of automation is necessary in order to make the problem realistically tractable in real-time; therefore, only one or at most a few pre-processed options should be presented to the pilot.

Naturalistic decision making models reveal that mental simulation is decisive in making correct decisions (Section 2.2.2.3). When two or more decision alternatives are provided, pilots need to generate explanations and then simulate the consequences of applying each of them in order to choose the best one. Koehler [1991] (cf. [Jones and Goethals 1972; Ross et al. 1975; Nisbett and Ross 1980]) reveals that when explanations are generated for opposite decision alternatives, the arguments generated for the side considered first prevail. Once the first alternative acquires enough evidence in its favour, some kind of inertia makes the implicit decision difficult to reverse. Koehler also argues that this phenomenon is akin to the notion of *mental set* or *fixedness* found in theories of problem solving.

It must be noted that other studies have failed to reveal the 'primacy' decision bias in question, e.g. Anderson and Sechler [1986].

Empirical investigation is required to better characterise the limits of configuration options that can be realistically handled by a pilot in real-time. This research aims at assessing the average number of configuration options that are considered by pilots in both normal conditions (for example, whilst in en-route cruise) and in critical, time pressure scenarios (for example, after a fault in the descent phase). This analysis forms the *necessary* and currently unavailable know-how for future, more sophisticated investigations.

4.2.7 Graphics

The focus of the material presented thus far has been on the design of effective *textual* decision support information. However, there is largely accepted evidence that in certain cases, graphical representations offer several advantages over the textual format, which cannot be ignored. Amongst the advantages of graphics, most researchers agree that they require: a) a reduced amount of mental computation to perform the task; and, b) less time spent searching for the information needed [Larkin and Simon 1987; Mayer and Gallini 1990].

A review of the literature returns a large amount of information concerning the development of effective graphical information for the support of safety-critical decision making activity. In the context of the analysis performed in this research, it is interesting to understand whether the provision of graphical decision support information, along with textual information, influences the ADR decision making process in any way.

The main focus of this work remains the investigation of the effects of explanations, implications and reliability figures in text format on ADR decision behaviour. However, with specific reference to the aerospace field, it is worth noting that recent research in cockpit automation has revealed a significant improvement in real-time, fault management decision performance when textual information is associated with a graphical description of a fault. A relevant example is provided by the NASA Fault Management Support System (FAMSS) [Hayashi et al. 2006; 2009; Huemer et al. 2005; McCann and Spirkovska 2005].

Hayashi et al. [2006] describe a study on fourteen highly experienced commercial airline pilots who assumed the role of spacecraft operator during the launch and ascent phase of eight spacecraft missions. The simulated missions were designed to be performed by a single operator, with a reconfigurable cockpit and no ground support. FAMSS provides a graphical representation of the fault that assists the pilots during the troubleshooting activity. The introduction of FAMSS resulted in improved malfunction resolution accuracy by 43%, reduced malfunction resolution time by 54%, and decreased cognitive workload of between 27% and 37%, depending on the type of malfunction being worked out.

The results obtained with FAMSS are in line with research on Adaptive Automation (AA) systems by Scerbo [1996], who predicted that the success of such reconfigurable systems would have been determined in large part by the capability of interfaces design to include all techniques of information exchange (e.g., visual, auditory, haptic, etc.).

The literature also contains evidence of issues with graphical representation of causal information and reconfigurable cockpit displays. Dale and Reiter [1995] notice that abstract concepts like *causality* may be difficult to convey by graphical means, while it is straightforward to achieve in text. Causality is particularly important during a real-time fault management decision process.

Petre [1995] brings attention to the fact that, although graphics are sometimes presented as an 'easy-to-understand' way of presenting information, research in applied psychology demonstrates that in many cases a considerable amount of expertise and background knowledge is necessary to interpret a graphic correctly. It can be argued that pilots are expert decision makers, so they should not be subject to this type of issue. However, aircraft are complex machines and pilots do not have a pervasive knowledge of each component and the interaction between them. The benefits of graphics over textual information are not obvious.

Wiener [1988] expounds on the potential towards display clutter and ill-considered symbols, text and colour in many graphical display designs for complex systems in many domains. Sarter [1995] associates lack of function transparency in reconfigurable graphical displays to loss of SA. Scerbo [1996] also provides an insight into several issues in terms of human-computer communication and overall system performance that emerge with reconfigurable graphical displays.

As previously mentioned, this research does not address the problem of efficient graphical rep-

resentation of fault management information on modern aircraft and spacecraft. Notwithstanding, it is interesting to collect information about the potential improvement in the decision performance of pilots during ADR decisions due to the introduction of a *graphical representation of the fault* which triggers the reconfiguration. Based on previous research, we speculate that this kind of information would favour the construction of good mental models and the process of mental simulation. For the time being, the framework proposed in this thesis does not employ any sophisticated graphic engine or design.

In conclusion, we advance the following speculating claim:

Claim 11 *A graphical representation of the fault that triggers an ADR would improve decision performance and accuracy.*

4.3 Claims and Hypothesis

The hypothesis for this research was given in Section 3.3. This chapter introduces several claims concerning the behaviour of pilots during ADR decisions and how decision support information should be designed in order to improve the effectiveness of the interaction and the process in general. All the claims made are consistent with the overall objective of fulfilling the hypothesis.

In order to provide the reader with a clear, comprehensive 'picture' of the problems addressed in the course of this Ph.D. programme and how the claims made relate to the hypothesis, the objective of fulfilling the hypothesis statement is dissected using Goal Structuring Notation (GSN) [Kelly and Weaver 2004] as shown in Figure 4.3. For reasons of space, the GSN chart has been divided into multiple charts, so that each strategy box in Figure 4.3 is elaborated in Figures 4.4 to 4.7. The GSN has been enriched with a coloured notation in order to signify the current state of work. In this regard, it must be noted that when a goal is marked as 'completed', this is only in relation to the objectives that were originally set for the thesis; obviously, no claims are made about having completely understood complex phenomena as the effect of the decision support information generated by the system proposed here on intricate human mental models like situation awareness.

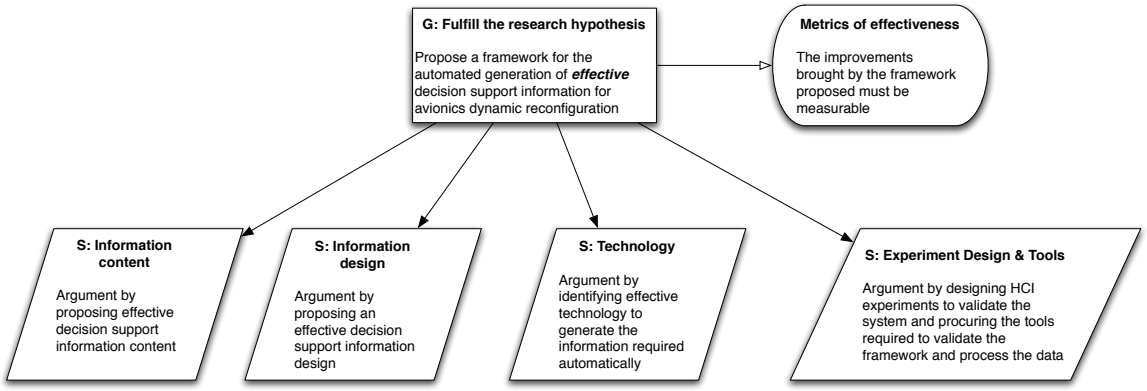


Figure 4.3: Research hypothesis dissected using the Goal Structuring Notation [Kelly and Weaver 2004].

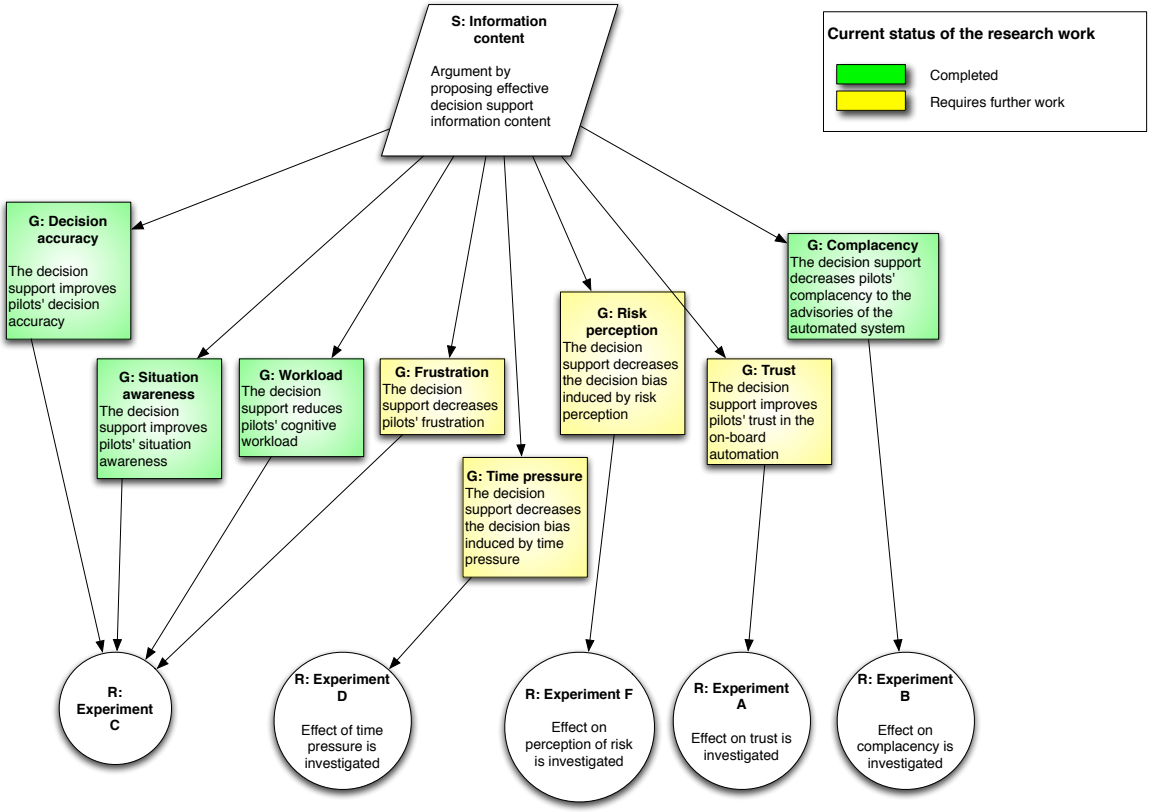


Figure 4.4: Elaboration of the 'Information content' strategy box from the GSN chart of Figure 4.3.

The GSN charts should be used as a reference to put into context which portion of the overall research agenda is addressed in which portion of the remainder of the thesis. As a general guideline, the objective of proposing a novel, effective DSS framework for automated generation of decision support information for ADR is addressed through four main strategies (Figure 4.3):

- *S1: Information content*. This chapter made a number of claims about the content of the decision support information generated by the proposed DSS; they are empirically investigated in Chapter 6.

- *S2: Information design*. Similarly, this chapter made a number of claims about the design and framing of the information produced by the proposed DSS; they are empirically investigated in Chapter 6.

- *S3: Technology*. The technology required to automatically generate the type of information implicitly defined by the claims made is discussed in Chapter 5. A series of algorithms and methods are critically reviewed; a novel algorithm for the automated resolution of data conflicts during the search for applicable configurations is presented and empirically evaluated.

- *S4: Validation tools*. Technology to validate both the performance of SaIRA and the effectiveness of the decision support it provides are required. Throughout this Ph.D. programme, the technology required to perform this task has been developed. This is the case with the

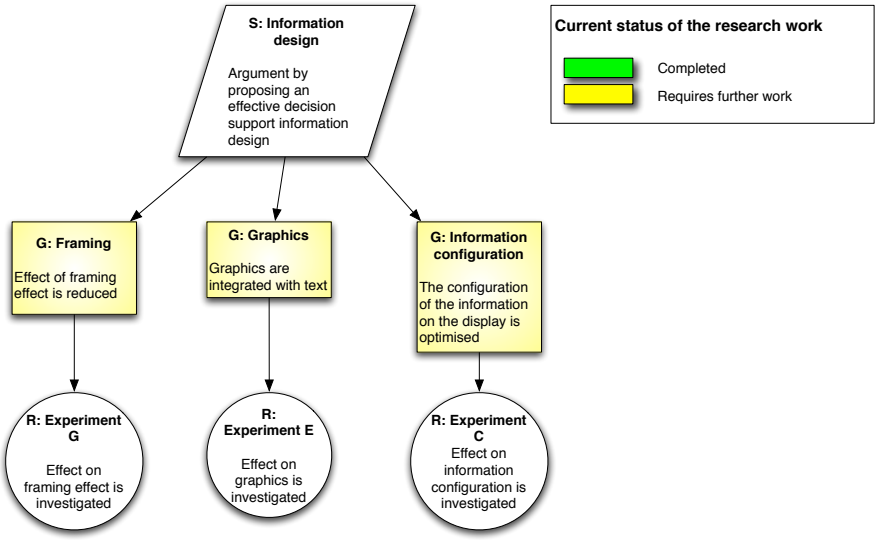


Figure 4.5: Elaboration of the 'Information design' strategy box from the GSN chart of Figure 4.3.

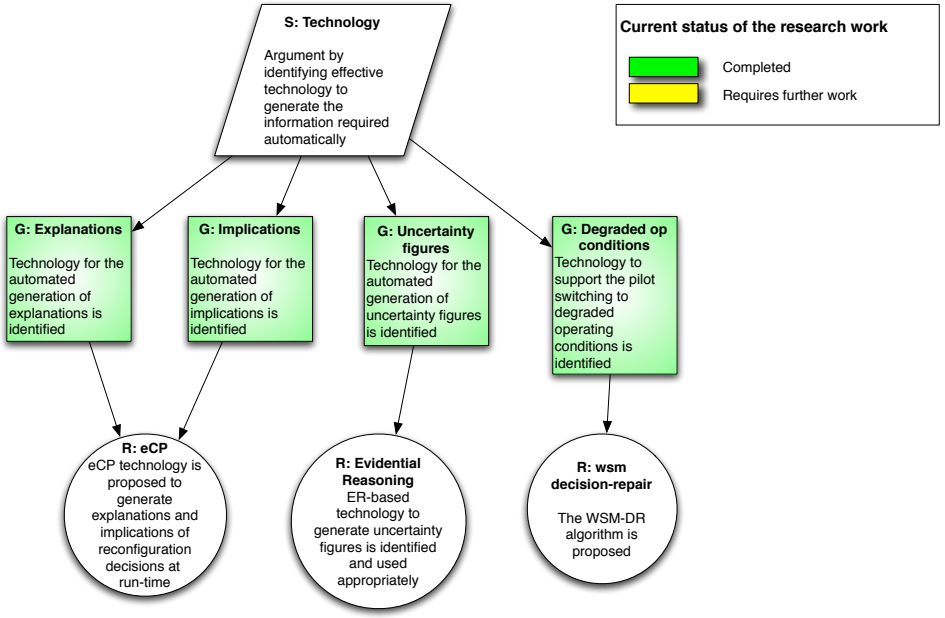


Figure 4.6: Elaboration of the 'Technology' strategy box from the GSN chart of Figure 4.3.

eye-tracking system (named SaIRA Eye-Tracking System, SETS) which has specific features that could not be found in other eye-tracking systems currently available (e.g. the possibility to integrate the system with the flight simulation software to mention but one). Furthermore, a new methodology for validating the performance and effectiveness of the system has been developed (combining objective and subjective metrics), together with the tools to analyse the data collected during the experiments (e.g. eye-movement data post-processing). The validation tools are described throughout Chapters 5, 6 and in Appendix A.

The main goal (i.e. 'G: Fulfil the research hypothesis') has an associated context box, 'Metrics

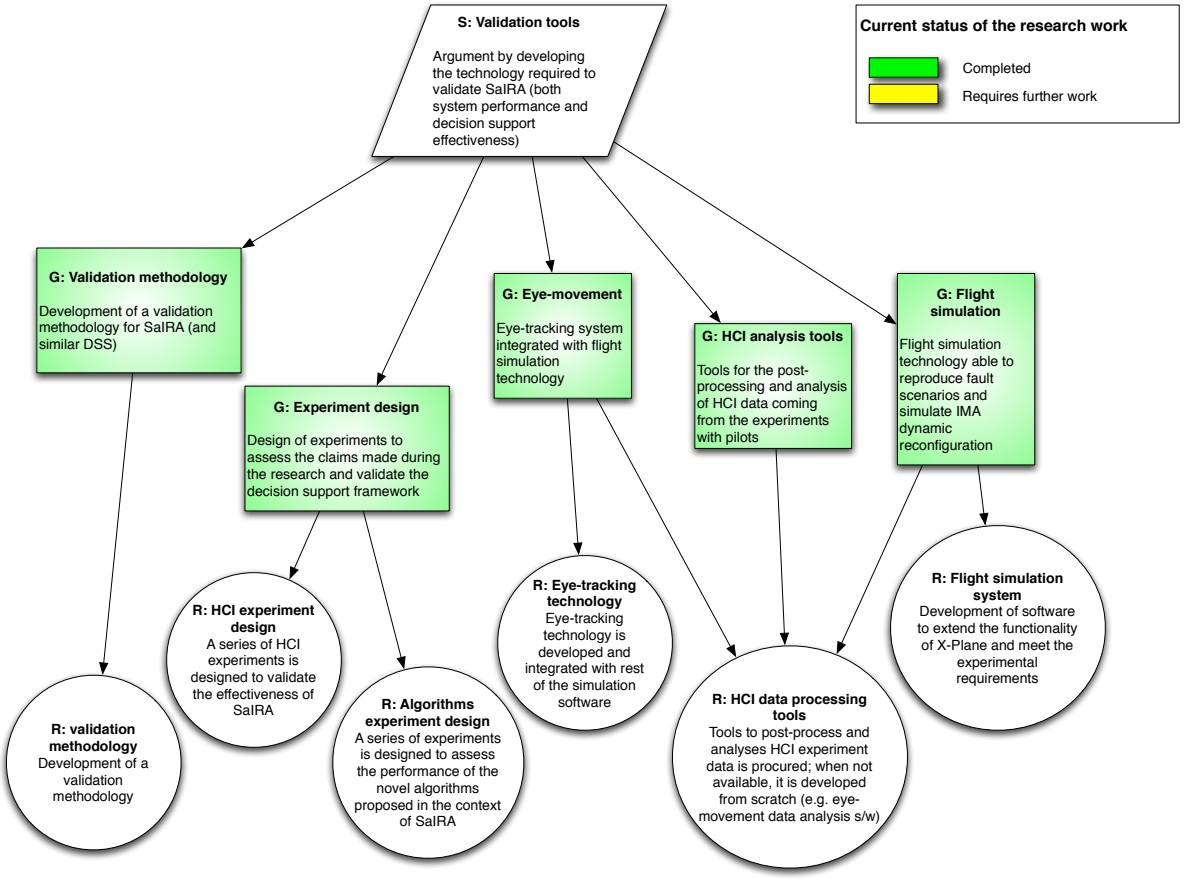


Figure 4.7: Elaboration of the 'Validation tools' strategy box from the GSN chart of Figure 4.3.

of effectiveness'. In order to assess the effectiveness of the proposed DSS, a number of subjective and objective metrics of decision support effectiveness are defined and used in the experiment presented in Chapter 6. This is necessary to quantify the improvements brought by the proposed DSS compared to basic decision support provided by fault warning and alarms currently available on standard cockpit.

4.4 Chapter Summary

This chapter covered the following topics:

- The Avionics Dynamic Reconfiguration (ADR) process is considered to be a specific instance of the more general SCMS dynamic reconfiguration process. A number of influential factors from the physical, emotional and temporal domains, which could bias pilot decisions during ADR, are discussed. Specific claims concerning the way each factor could possibly bias pilot decisions are advanced on the basis of past and current research in cognitive psychology and cognitive system engineering. These claims aim to characterise a persistent user profile for the DSS framework for proposed ADR decisions. All the claims will be empirically tested (Chapter 6).

- In the light of the hypothesised user profile, it is argued that effective decision support information for ADR (i.e. able to mitigate the effects of the identified biases) should contain the following three types of information:
  - *Explanations*: *why* the system is making a specific suggestion?;
  - *Implications*: (*what-if* type of information) what the consequences of applying a specific configuration are?
  - *Reliability*: *how reliable* the advice generated by the system is?
- Given the extent of the problem being addressed, it is not possible to cover all the aspects of the domain in the context of a Ph.D. programme. A GSN chart is provided which represents a snapshot of the current state of research; the chart allows the identification of which facets of the problem have been addressed in detail and which require further work. Furthermore, the chart provides an organic link between all the claims made in this chapter and the hypothesis statement given in Chapter 3.

For the sake of clarity, the complete set of claims advanced in this chapter is gathered here.

- **Claim 1 - Risk perception**: If prompted by the system, pilots would reconfigure the avionics in situations which are not particularly risky (e.g. whilst cruising). They would refrain from doing so in situations of pressing risks (e.g. before landing, when the system is more unstable and a change to the current state could become a catalyst for catastrophic consequences).
- **Claim 2 - Time pressure**: Pilots would react to time pressure by means of a) acceleration, b) selection of information, and c) alteration of information search pattern.
- **Claim 3 - Time pressure**: Time pressure would decrease pilot decision accuracy.
- **Claim 4 - Time pressure**: Time pressure would affect pilot decision time. Amongst the other effects, time pressure would reduce the number of configuration options considered and the way the option information is explored.
- **Claim 5 - Stress and frustration**: Heightened states of stress during ADR decisions can possibly lead to frustration. The negative effect of frustration would be mitigated by providing pilots with *effective* decision support information.
- **Claim 6 - Framing effect**: The pilot decision behaviour during ADR would be subject to the framing effect; more specifically, presenting ADR information in terms of its *reliability* instead of its *unreliability* would make pilots more comfortable with accepting the proposed configuration.
- **Claim 7 - AIC**: Pilots would be subject to Automation-Induced Complacency. [Dekker 2000] notices that practitioners from within the organisation that launched the Space Shuttle Challenger in 1986 reported that on that occasion, complacency was based on a *justified*

assumption of satisfactory system state, since there was no evidence of the contrary. As a consequence, we hypothesise that reliability figures would limit the emergence of *complacency*. We expect pilots to be less complacent to wrong ADR advice when they are associated with low reliability. During highly autonomous ADR, this phenomenon would be mitigated by decreasing the level of autonomy of the process.

- **Claim 8 - Information type**: Decision support information for SCMS reconfiguration should be "unpacked"/framed so as to make the following cues readily understandable to the operators for each reconfiguration option:
  - *Explanations*: *why* the system is making a specific suggestion;
  - *Implications*: (*what-if* type of information) what the consequences of applying a specific configuration are;
  - *Reliability*: *how reliable* the advice generated by the system is.
- **Claim 9 - Trust**: Pilot trust in a DSS for ADR would decrease when the system made (apparently) unacceptable inferences. This phenomenon would be mitigated by providing pilots with explanations of the inference processes.
- **Claim 10 - Reliability figures**: Providing *reliability* figures would influence pilot decision making performance in the following ways:
  - the framing effect would emerge when providing pilots with reliability or uncertainty terms; more specifically, pilots would feel more comfortable applying a configuration associated with high reliability than with low uncertainty;
  - evidently wrong ADR suggestions, when associated with low reliability, would be more easily spotted;
  - low and high reliability options would both be easier to process than medium reliability options, i.e. the decision time would increase with medium reliability options.
- **Claim 11 - Graphics**: A graphical representation of the fault that triggers an ADR would improve decision performance and accuracy.

Chapter 5

A Framework for Interactive Dynamic Reconfiguration

The material presented in the previous chapters fulfils three main tasks, a) it discusses the necessity for human involvement during the SCMS dynamic reconfiguration process, b) it examines and characterises the process in question from the point of view of naturalistic decision making, c) it generates a set of claims about the characteristics of pilot behaviour during ADR and about the characteristics of effective decision support information.

In order to realistically evaluate the decision making ideas proposed, technology was developed as part of this research to manage the dynamic reconfiguration process of the SCMS. This technology is designed to perform the following two tasks autonomously:

- generating configurations for IMA equipped with multi-sensor data network technology (inspired by modern, real avionics);
- generating decision support information relative to the configurations proposed to the system operator, according to the claims put forward in the previous chapters.

The purpose of this chapter is to describe both the process of design and the characteristics of the decision support technology developed in the context of this thesis. All the proposed technology is integrated with methods and ideas (e.g. heuristics) already available in the literature to form the Safe and Interactive Reconfiguration Architecture (SaIRA); SaIRA has been used to perform the experiments set out in Chapter 6. Two novel algorithms for automated generation of decision support information have been developed as part of this thesis; they are briefly and qualitatively described in this chapter but their details, including performance analysis, are set out in Appendix B.

The decision support technology discussed in this chapter is based mainly on the Constraint Programming paradigm, therefore some propaedeutic information is provided with the double purpose of (a) explaining the design decisions taken and (b) clarifying the nature of the information the pilots are provided with. First, an ontology for the avionics dynamic reconfiguration problem is introduced; the ontology is subsequently translated in a Constraint Satisfaction Problem (CSP) which is programmatically used by SaIRA to generate decision support information. Once the

nature of the decision support information is outlined and the constraint set by the CSP technology are clarified, the way the user interface of SaIRA has been designed—with the support of two pilots—is discussed in detail.

5.1 An Ontology for the Reconfiguration Problem

The analysis performed so far generated a body of knowledge that is now going to be structured into a framework for a dynamic reconfiguration system for SCMS.

The Knowledge Engineering domain offers several tools for knowledge representation and organisation (see DKE Committee [2010] and KER [2010] for a review of the field). The development and use of an **ontology**[1] for small and large-scale knowledge-based systems has been investigated in the past [Gruber 1995; Swartout et al. 1996; Studer 1998; Staab et al. 2005; Abecker and Elst 2009]. A consistent body of research in this field focuses on information reuse, sharing, formalisation and encoding within ontologies.

On the basis of these studies, the effectiveness of an ontology to structure the information required to engineer complex processes is taken for granted. As an extension to current research, this thesis advocates the use of an ontology to proceed organically from the analysis phase towards the development of SCMS reconfiguration management systems. More specifically, a contribution is given by:

- producing a basic, flexible ontology for the SCMS dynamic reconfiguration problem;
- proposing an approach to translate this ontology into a Constraint Network, which represents the knowledge base of SaIRA, and which can be programmatically accessed using Constraint Programming techniques;
- using the ontology to support the definition of the characteristics of the decision support information directed to the operator of the system.

5.1.1 SaIRA Ontology

In information science, an ontology is usually introduced as a formal representation of knowledge, made up of a set of *concepts* within a domain and the relationships between those concepts. The **SaIRA ontology** can be interpreted as a meta-model for an interactive SCMS dynamic reconfiguration process. It responds to the need to describe aspects of the SCMS domain that are relevant to the construction of the framework in a format that can subsequently be represented in a computing system. In other words, the ontology allows the mapping of human-interpretable descriptions of the system to the Constraint Network (CN) model introduced later in this chapter (Section 5.3).

SaIRA is a generic architecture for reconfigurable SCMS. The ontology presented here is kept simple in order to permit much stronger assumptions about the characteristics of the system being analysed, but it can easily be extended to other systems. In this regard, it is worth noting that the

[1] The Greek philosopher Parmenides began the tradition of using ontology as an instrument to study the organisation and the nature of the world *independently of the form of our knowledge about it*.

CN paradigm has been purposely chosen to formally represent the knowledge-base and 'drive' the reconfiguration process, because it allows the introduction of new concepts at any time during the development process, e.g. the CN-based reconfiguration algorithms used in SaIRA are not sensitive to initial conditions and do not require setup.

In the remainder of this section, 'concepts' belonging to the ontology are in the following font: `CONCEPT`. The concepts and the relationships between them are described, and the resulting framework is then translated into a constraint network which can be implemented in a real computing system.

5.1.1.1 Baseline ontology

Section 2.1.3 listed several definitions of 'dynamic system reconfiguration' from the literature, which capture different aspects of this concept. Here, previous definitions are collated into a general statement which allows a direct mapping to the seven ontological concepts that the SaIRA ontology pivots around.

Definition 5.1.1 *A **SCMS dynamic reconfiguration** is an `EVENT`-driven process of transitioning between two different `CONFIGURATIONS`, which aims at adapting the behaviour of the system to a change of `FUNCTIONS` required or `RESOURCES` available, whilst meeting a number of pre-defined functional and non-functional `REQUIREMENTS` and the preferences of the `OPERATOR`.*

Figure 5.1 contains a graphical representation of the baseline SaIRA ontology. Each ontological concept is informally described hereinafter. The description provided is generic enough to allow further specialisation, permiting the adaptation of the framework to other systems.

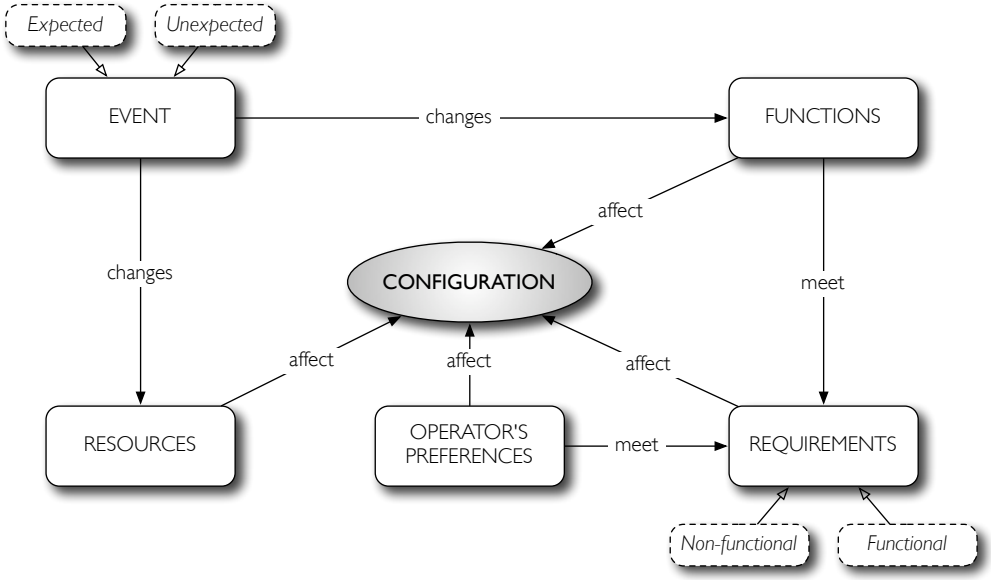


Figure 5.1: Baseline, intuitive ontology of the SCMS dynamic reconfiguration problem.

EVENT

In SaIRA, an EVENT is an observable, not necessarily extraordinary, occurrence that affects either the set of system RESOURCES available or the system FUNCTIONS required, or both.

An EVENT can be either 'expected' or 'unexpected'. An example of an expected EVENT is a request for a change of operating mode triggered by the operator (i.e. non-extraordinary EVENT). This kind of reconfiguration is generally referred to as *planned reconfiguration* in the literature, as opposed to *unplanned reconfigurations*—a typical example of this is a reconfiguration triggered automatically to recover from a fault.

An EVENT has the following properties:

- **Criticality**: EVENTs have different criticality which, to a certain extent, changes with the operating conditions. For instance, a fault in the lighting system of an aircraft is less critical than a fault in the hydraulic system. However, the criticality of a fault in the lighting system is higher during a landing manoeuvre at night than in daylight with good weather conditions. EVENTs with different criticalities can possibly trigger different types of reconfigurations, with different timings. For example, a reconfiguration in response to a fault with very low criticality can be deferred or even not performed. Criticality is a prominent property of an EVENT in the context of an SCMS.

- **Reliability** (only applies to unexpected EVENTs): dynamic reconfiguration is a recovery action, being part of the Fault Detection, Identification and Recovery (FDIR) process.

  Faults must first be detected and identified by the system, before it can proceed to generate the right configuration for the given conditions. Sensors are used to detect faults; modern SCMS employ sensor data-fusion techniques to merge the information coming from different sensors, characterised by different technical features and degrees of *detection reliability* (this is dealt with in more detail in Section 5.5).

  More specifically, in a sensor data fusion network, the same sensor can be used to detect and identify several types of faults. However, the readings will have a different degree of reliability, depending on how suitable that sensor is to detect/identify the fault in question [Hall and Llinas 1997]. As a result, here reliability is intended to mean "how reliable the assessment of an EVENT made by the system is".

RESOURCE

A RESOURCE is any physical or virtual component of limited availability within the reconfigurable system. A SCMS is composed of *computing modules* (called Line Replaceable Modules (LRM) in the Integrated Modular Avionics literature), *sensors* and *actuators*. These can all be regarded as specialisations of the RESOURCE concept.

In an early study in the domain of dynamic reconfiguration of safety-critical distributed systems, Nicholson [1998] introduces the dynamic reconfiguration process in terms of a General Topology Problem (GTP). In the author's view, a topology consists of a configured set of units,

employed to fulfil a given set of logical control actions. In this logic, the reconfiguration process entails: (a) determining an allocation of software units to individual hardware units, and (b) determining which sensors and actuators may be shared by which services. Nicholson's concept of resource is important to this research because it highlights the link between the physical components of the system and the function they fulfil within the overall architecture. In line with Nicholson, each RESOURCE in SaIRA fulfils one or more functions.

The RESOURCE concept is subject to specialisation. For instance, the memory installed on an LRM is a RESOURCE of the LRM. A fault could affect only some of the memory modules of an LRM, but not the whole module; in such a situation the LRM could continue to operate but under degraded functionality.

In this regard, it is worth noticing that the *degree of specialisation* of the ontology strongly influences the *effectiveness* of the reconfiguration process. For example, a higher specialisation of the ontological concepts generally allows the characterisation—and therefore handling—of smaller changes to the state of the system.

The set of available resources can possibly be altered (i.e. restricted, expanded or modified) at run-time by an EVENT. Consequently, the set of available RESOURCEs *limits* the applicability of any CONFIGURATION to the system in the present state. For instance, after a fault to one of the four Trent 900 engines of a Airbus A380 aircraft, all configurations that rely on four engines are no longer applicable.

FUNCTION

A FUNCTION is a *process* or *task* performed by the system. A single FUNCTION can be accomplished by more than one software application. The set of applications that compose a FUNCTION require a set of RESOURCEs in order to execute.

In this ontology, an application, or software application, follows the definition given by the Aeronautical Radio Incorporated (ARINC-653) [2006] standard: it is computer software running in a single partition with assured allocation of processing time and memory. Each application is mapped to a single computing module during the reconfiguration process and is associated with a set of sensors and actuators required for execution. Inter-dependencies exist amongst applications and, as a result, amongst FUNCTIONs.

A FUNCTION is associated with a set of functional and non-functional REQUIREMENTs (see the next sub-section) that must be met in order to assure its accomplishment, e.g. memory requirements, inter-dependencies, timing requirements. The REQUIREMENTs associated with a single FUNCTION are taken in consideration during the reconfiguration process only if the FUNCTION in question is required by the target CONFIGURATION.

REQUIREMENT

The concept of REQUIREMENT in SaIRA is similar to the common interpretation from the Requirements Engineering domain. It is "a statement that identifies a capability or function that is needed by a system in order to satisfy its customer's needs" [Bahill and Dean 1999].

A functional REQUIREMENT describes *what* a system must do; a non-functional REQUIREMENT (also referred to as 'performance requirement' in the literature) describes *how* the system should operate.

Both the reconfiguration process and the result (a configuration) must meet a set of pre-defined functional and non-functional REQUIREMENTs. For instance, consider the following statement taken from the Boeing 737 aircraft operations manual: "a single Flight Management Computer is not certified as a sole source of navigation system. It is certified to navigate accurately in conjunction with an accurate radio navaid environment" [The Boeing Company 2002] (page 672). This statement introduces at least two safety-related REQUIREMENTs on any configuration, a) a configuration shall have two FMCs operating, b) if only one FMC is available, then an accurate radio navaid environment must be available. On the other hand, the 'maximum time to complete a reconfiguration' is an example of a requirement on the *process* of reconfiguration.

The set of functional REQUIREMENTs should not be confused with the set of FUNCTIONs required by a CONFIGURATION. The former refers to the overall system, it is defined at design time (i.e. prior to operation) and does not change at run-time. The latter refers to a single CONFIGURATION and changes dynamically depending on the operating conditions. The set of FUNCTIONs required are affected by both the occurrence of EVENTs and by the OPERATOR's preferences (see Figure 5.1); in contrast, neither an EVENT nor the OPERATOR modify the REQUIREMENTs defined by system architects.

For example, the OPERATOR (pilot) of a Boeing 737 can decide to have the Distance Measuring Equipment (DME) active instead of the VHF Omni-directional Radio Range (VOR); however, the pilot cannot switch off the applications handling laser gyros in a configuration that uses the Inertial Reference System (IRS) because this FUNCTION requires them to operate (dependency REQUIREMENT).

In summary, REQUIREMENTs, amongst other objectives such as ensuring performance, guarantee the integrity of the overall system. For this reason, both the set of FUNCTIONs that are activated in a CONFIGURATION and the preferences of the OPERATOR must meet the pre-defined REQUIREMENTs (as shown in Figure 5.1).

OPERATOR

The OPERATOR of a SCMS is the human who operates the system. In SaIRA, only one OPERATOR is allowed to control the system and no co-operative scenarios are taken into account; furthermore, as this framework relates to manned systems, no remote control scenario is considered.

The OPERATOR concept relates to a *real* human being (i.e. not a model), whose rationality is not perfect and whose decisions are subject to a number of cognitive biases. The OPERATOR can express preferences during the process of reconfiguration by choosing between a reduced set of reconfiguration alternatives provided by the decision support executive of SaIRA. In order to preserve the safety of the process, the OPERATOR must confirm the application of any configuration.

CONFIGURATION

The CONFIGURATION concept in SaIRA gathers all the ontological concepts introduced so far as described by Definition 5.1.1. It follows that the features of a CONFIGURATION are *affected* by the RESOURCEs available at any time, the FUNCTIONS required, the pre-defined functional and non-functional REQUIREMENTS and by the preferences elicited by the OPERATOR at run-time.

As previously seen, Nicholson [1998] introduced the concept of CONFIGURATION in the context of a topology problem. Such a CONFIGURATION can be autonomously generated by a computer. In SaIRA, Nicholson's idea of CONFIGURATION is still valid but the problem of generating it is extended by involving the OPERATOR in the process, which makes fully autonomous approaches and algorithms inapplicable.

5.2 Configuration, Reconfiguration and Constraint Programming

The problem of **configuration** has its roots in the 1980s. Hadzic and Andersen [2004] trace it back to Mittal and Frayman [1989], who defined configuration as "a design activity of assembling an artefact that is made of a fixed set of well defined component types where components can interact only in predefine ways". If the configuration problem is about "assembling" an artefact, the **reconfiguration** problem is about *transitioning* between two different artefacts, usually minimising the cost.

A review of the literature reveals that both the problem of configuration and reconfiguration have been extensively investigated in the domain of Constraint Programming (CP). CP-based technology has been developed for configuration problems in various domains, retail product configuration, power supply restoration services, configuration of university degree courses, distributed network services, aircraft furnishing, task scheduling, personal computer configuration and satellite payload configuration *inter alia* [Mø ller et al. 2001; Van Der Linden 2002; Subbarayan et al. 2004; De Givry et al. 2002; Hadžić et al. 2005].

Part of the ADR problem concerns allocating tasks to resources in accordance with a set of scheduling requirements and constraints on routing messages on the network data buses. A large body of research has tackled these problems in domains other than aviation, confirming the effectiveness of CP-based techniques [Cambazard et al. 2004; Cheng and Smith 1997; Chun et al. 1997; Carlsson et al. 1998; Frei and Faltings 1999; De Givry et al. 2002; Elkhyari et al. 2002; 2004; Hladik et al. 2005; 2008; Leeuwen et al. 2002; Pang and Goodwin 1996].

Two interesting studies are proposed by Weibenbacher et al. [2005] and Frei and Faltings [1999] who introduce a CP-based technology for reconfiguration problems in safety-critical contexts. Whilst these studies are interesting because they focus on the safety of the process, human involvement is not taken into account, hence the ideas presented are not fully applicable to this thesis.

Hladik et al. [2008] discuss the advantages of tackling complex combinatorial problems with constraint programming, including task allocation over distributed networks. The following advantages make this technique particularly appealing for the ADR problem:

- *declarativity*: the variables, domains and constraints are simply described;
- *genericity*: it is not a problem-dependent technique, general rules are mechanically performed during the search;
- *adaptability*: each constraint can be considered as independent and a model could be simply extended by merging the different constraints;
- *non-parametric ability*: no sensitivity to initial parameters (i.e. temperature and cooling for Simulated Annealing), no training, easiness of extensions of the method to new models;
- *completeness*: if there is no solution, the CP algorithm is able to prove it (contrary to heuristic methods, which are unable to decide);
- *utilization*: CP has been effectively used for a large range of combinatorial problems;
- *performance*: CP has proved to perform well in task scheduling and resource allocation, which is a fundamental part of the ADR problem.

Interactive (re)configuration technology is designed for (re)configuration problems in which the human is actively involved in the search for a solution. More specifically, Hadžić et al. [2005] define interactive configuration as "an application of Constraint Satisfaction Problems that assists a user in her search for a valid variable assignment (a configuration) in a combinatorial problem". A relevant body of research in this domain has been published recently [Papamichail and French 1999; Frayman 2001; Jussien 2003; Madsen 2003; Jensen 2004; Hadžić et al. 2005; van der Meer et al. 2006; Tiedemann et al. 2006].

Ideas from Constraint Programming theory, (re)configuration problem techniques and recommendation technology converge in a specific class of recommenders which were briefly introduced in Chapter 2, called Constraint-Based Recommenders, with SaIRA being classified as such a recommender. As well as the advantages of the CP paradigm previously listed, the rationale for this design decision is that, unlike collaborative and content-based methods, constraint-based systems do not suffer from cold start problems, and do not require the generation of a meaningful history of pilots preferences; both features are necessary in the application domain pertaining to this thesis. In fact, parameter tuning is known to be an extremely difficult and error-prone process with complex combinatorial problems characterised by large search space [Ridge and Kudenko 2007] (which is the case of the ADR dynamic reconfiguration problem). Furthermore, constraint-based systems allow the specification of preferences between the decision alternatives *a priori* (for example, using pre-defined weighting functions devised by the system designers on the basis of domain-dependent knowledge) which do not require a history of the outcome of the decision-maker's choices in past scenarios.

The remainder of this chapter introduces the **Safe and Interactive Reconfiguration Architecture (SaIRA)**, developed by the author. It has three main components:

- *Reconfiguration Executive (RE)*: this component autonomously generates a set of applicable system configurations designed to mitigate the effects of the event that triggered the reconfiguration;

- *Sensor Fusion Executive (SFE)*: fuses the information coming from a multi-sensor data network using Evidential Reasoning techniques. The inferences of the SFE provide input data for the RE;

- *Decision Support Executive (DSE)*: this component implements Constraint-based Recommendation technology. It uses information produced by both the RE and the SFE to generate readily understandable decision support information for the system operator.

The constraint-based recommender proposed with SaIRA is different from many other systems of this type, due to its design focus for quick, critical decisions. In fact, instead of issuing, retracting or modifying *each* constraint in isolation, the user interacts with the system only by "exploring" a set of predefined and applicable solutions (i.e. configurations) and choosing the one that is most appropriate for the current operating conditions. In other words, the user chooses between a small number of 'artefacts' produced by the system.

In this regard, Frayman [2001] elaborates a list of user-interaction requirements for efficient implementations of interactive constraint satisfaction systems. The author develops a framework that supports the following user gestures: (a) select an item, (b) retract the selection, (c) reselect another item, (d) specify (i.e. I do not want this item), (e) restrict range of values desired.

The model proposed by Frayman is designed for situations in which the decision maker has a lot of time to make the choice. This is not the case for avionics reconfiguration decisions, therefore, Frayman's work does not fit well with this research.

To the best of our knowledge, SaIRA is innovative inasmuch as it is the first constraint-based recommender designed to provide effective naturalistic decision support by paralleling the cognitive strategies of the pilot. This is done by implementing algorithms that automatically generate explanations and implications, as well as an assessment of the reliability of each piece of reconfiguration advice.

5.3 Reconfiguration as a Constraint Satisfaction Problem

Slightly different notations are used in the literature to describe Constraint Satisfaction Problems (CSP). For this reason, whilst discussing the characteristics of SaIRA and the design choices made, this section introduces some necessary terminology, concepts and definitions.

The CSP material is intended as an introduction to SaIRA and the framework design choices. For detailed information on CSP, the reader should refer to Tsang [1993].

A CSP can be described in terms of a constraint network:

Definition 5.3.1 *A **Constraint Network (CN)** is defined by a triple $R = (X, D, C)$ where:*

- *X is a finite set of variables;*

- *D is a function that maps each variable $x \in X$ to a finite set of values, $D(x)$, which it is allowed to take. $D(x)$ represents the domain of x, and it is sometimes denoted as D_x;*

- *C is a set of constraints on the variables in X. Let $S = \{x_k, \ldots, x_l\} \subseteq X$. Each constraint $C_S \in C$ is a relation with scheme S and instance C_S. The set S is the* scope *of the constraint. The arity of the constraint is denoted as $|S|$. Each tuple in the instance $C_S \subseteq D_{x_k} \times \ldots \times D_{x_l}$ specifies a combination of values which the constraint allows.*

Section 5.1.1 defines an informal ontology for SaIRA. Here the ontology is restructured around the concept of 'constraint' and formalised as a CSP. As shown in Figure 5.2, each concept in the ontology is translated into constraints. For instance, the notion that, at any time, the state of a power generator can be either 'on' or 'off' is implemented through the following constraint:

$$C_{p\_gen} : p\_gen \in \{0, 1\} \tag{5.1}$$

where 0 codifies the state of 'off' and 1 the state of 'on'.
Consider the following `REQUIREMENT` for a generic IMA:

> *On any Line Replaceable Module, the sum of the memory consumption of all tasks running at the same time must always be less than the total memory installed.*

Let LRM_k be a generic Line Replaceable Module, let Mem_k be the total memory available on it, let c_j^k be a boolean value representing whether a generic task is active or inactive on LRM_k, and let m_j be the memory required by the task in question. The requirement in question is translated into the following constraint:

$$Mem_k \geq \sum c_j^k \times m_j \qquad \text{where} \qquad c_j^k \in \{0, 1\} \tag{5.2}$$

An `EVENT` in SaIRA is also modelled as a constraint. For instance, if the power generator mentioned in Constraint 5.1 is affected by a fault, the following constraint is issued:

$$C_{fault\_p\_gen} : p\_gen = 0 \tag{5.3}$$

meaning that the power generator in question is not working anymore.

Constraint 5.3 is restrictive to the point where the variable is limited to taking a single value. In other words, the constraint implicitly leads to the *assignment* of a value to the variable. This introduces the assignment and instantiation operations.

Definition 5.3.2 *Let $R = (X, D, C)$ be a constraint network. An **assignment** of the value $a \in D_x$ to the variable $x \in X$ is denoted $\langle x, a \rangle$. An **instantiation** of a set of variables $\{x_k, \ldots, x_l\} \subseteq X$ is a simultaneous assignment of values to the variables $\{x_k, \ldots, x_l\}$ and is denoted $\{\langle x_k, a_k \rangle, \ldots, \langle x_l, a_l \rangle\}$.*

Definition 5.3.3 *An instantiation $\bar{a}$* satisfies *a constraint C_S if $\bar{a}_{|S} \in C_S$. Let $R = (X, D, C)$ be a constraint network. An instantiation $\bar{a}_T$, where $T \subseteq X$, is* consistent *relative to R if, and only if, $\bar{a}_T$ satisfies all constraints $C_S \in C$ such that $S \subseteq T$.*

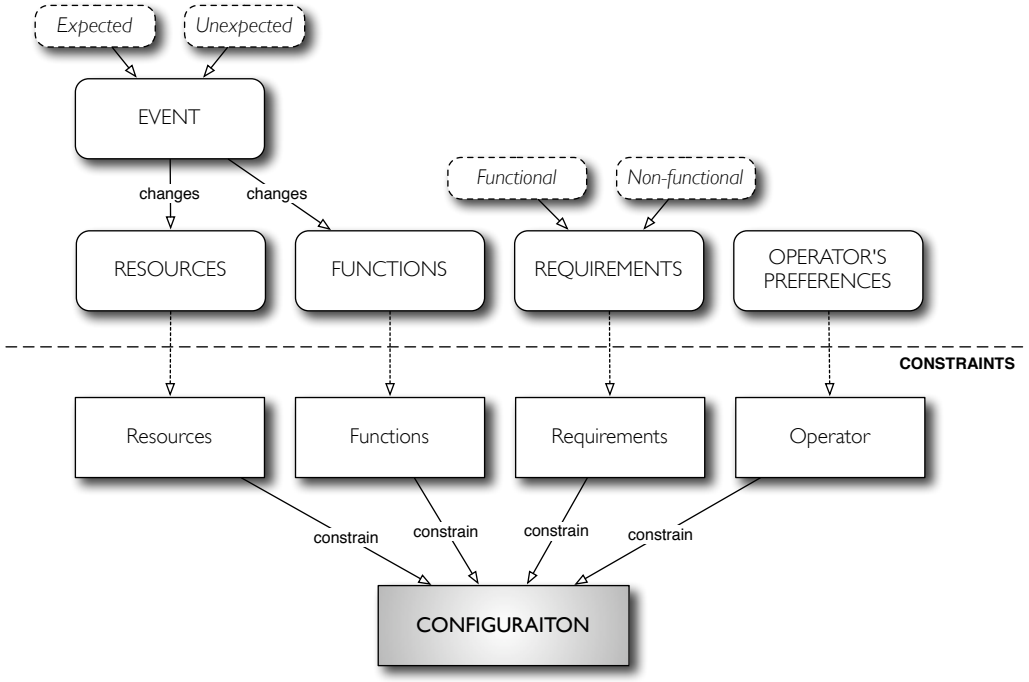


Figure 5.2: Constraints-based ontology of the SCMS dynamic reconfiguration problem.

Constraint 5.3 makes Constraint 5.1 *redundant*: a constraint is redundant if its removal does not change the set of solutions.

Constraint 5.3 represents the assignment of a single variable. When all the variables of the CN are included in an instantiation, and a value can be successfully assigned to each and every variable, a solution to the problem is found, which in SaIRA represents an applicable configuration. More specifically:

Definition 5.3.4 *A **solution** of the constraint network $R = (X, D, C)$ is an instantiation of all the variables in X which is consistent relative to R. $Sol(R)$ is the set of all the solutions to a constraint network R.*

A CN is *satisfiable* if, and only if, $Sol(R) \neq \emptyset$; it is *unsatisfiable* if, and only if, $Sol(R) = \emptyset$.

In general, after having verified that the CN is satisfiable, finding a solution to it typically corresponds to one of the following three problems:

a) finding *a solution* without any specific preference;

b) finding *all the solutions* (the whole set *Sol(R)*);

c) finding *one or more (sub)optimal solutions* (requires an optimality function).

During ADR, it is a requirement for SaIRA to generate one or a few (if available) configuration alternatives from which the pilot can select the one that best matches his/her preferences. As a result, approach (c) is adopted in SaIRA.

The case in which one or more solutions that satisfy all the constraints are available translates, in ADR terms, into what Spitzer [2000] defines as 'minor reconfiguration': all the requirements

are met and the functionality of the aircraft does not change after the reconfiguration. A minor reconfiguration is totally 'transparent' to the pilot, apart from a short time period due to the actual implementation of the reconfiguration (e.g. switching off some functions in order to reload them on different computing modules).

Clearly, there are cases in which the consequences of the `EVENT` are such that the resulting CN becomes unsatisfiable. In this case, the CSP is *over-constrained*. There are two possibilities in such a situation:

1. one or more constraints can be relaxed, hence a sub-optimal solution can still be found (defined by Spitzer [2000] as 'major reconfiguration');

2. no constraints can be relaxed, the consequences of the `EVENT` are such that the functionality of the aircraft is irremediably compromised.

A great deal of research has been conducted to develop *automated* techniques to handle over-constrained CSPs. These include associating priorities with constraints (the constraint with lowest priority is relaxed first), fuzzy logic, Multi-Attribute Utility Theory, cost functions, weighted CN and Bayesian approaches [Schiex 2005; Felfernig and Burke 2008].

Smith et al. [2005] observe that "typically an automated algorithm is better suited to conducting repetitive search steps that are not possible for a human user, while a user typically has more specific knowledge about the target domain that is difficult to formalize in general terms to be used by an algorithm".

The observation from Smith is particularly relevant in the ADR problem given its safety-critical context. As discussed in Chapter 4, it is not possible to codify pilot intentions in the logic of the avionics, as a certain degree of epistemic uncertainty is always going to characterise the reconfiguration problem. As a result, over-constrained situations are the stage of the search for applicable configurations in which full automation is "broken" and SaIRA calls upon the pilot's intervention.

5.3.1 Interactive Reconfiguration

The overall set of constraints forming the knowledge base of SaIRA is divided in *hard* and *soft* constraints. Hard constraints cannot be relaxed because this would make the problem inconsistent. An example of a hard constraint is a scheduling constraint on a flight control application. On the other hand, soft constraints can be relaxed at run-time. An example of a soft constraint is `Elevator Feel System ACTIVE`: the EFS functionality is not critical to fly the aircraft, so for instance, it could be acceptable to disable this function in situations when there is not enough power for all the on-board sub-systems.

On the basis of the definition of CN given in Definition 5.3.1, it is now possible to introduce the Dynamic Constraint Network (DCN) [Amilhastre et al. 2002; Madsen 2003]:

Definition 5.3.5 *A **dynamic constraint network (DCN)** is a quadruple $\Delta = (X, D, Z, H)$ where (X, D, Z) is a satisfiable constraint network with a static set of variables X, domains D and a static set of constraints Z. The set H is a dynamic set of constraints on the variables in X.*

Definition 5.3.6 *An instantiation $\bar{a}$ is a **solution** to a DCN $\Delta = (X, D, Z, H)$ if, and only if, $\bar{a}$ is a solution to the constraint network $(X, D, Z \cup H)$. The set of all solutions to Δ is denoted $Sol(\Delta)$. The DCN Δ is **satisfiable** (respectively unsatisfiable) if, and only if, the constraint network $(X, D, Z \cup H)$ is satisfiable (respectively unsatisfiable).*

Chapter 2 discusses the necessity of structuring the interaction between the pilot and the system as a *selection* problem. With reference to the definitions given above, if the problem was structured as a *configuration* problem, the pilot would have the possibility of adding, retracting and modifying the constraints in Z directly. As already discussed, the granularity of this type of interaction is not suitable to ADR operating conditions in the majority of situations.

A selection problem characterised by a large number of decision alternatives to choose from is likely to become intractable for the pilot in a reasonable time. Having denoted the maximum number of configuration alternatives that can be presented to the pilot using k, the goal of the ADR problem can be summarised in the following two steps:

1. add all the constraints in Z to the CN;
2. repeatedly add the constraints in H to the CN until $|Sol(\Delta)| \leq k$.

Constraints are added and retracted through the enumeration process:

Definition 5.3.7 *An **enumeration** is a sequence of constraint additions and retractions* (backtrack).

In order to guarantee that the systems proposes only applicable avionics configurations, the CN must be designed in a way that the complete enumeration of the set of hard constraints, Z, never leads to an unsatisfiable state. This problem has been already addressed in the literature and several approaches have been developed (e.g. consistency checks [Tsang 1993], finite state automata [Hadzic et al. 2007]).

A more interesting topic for this research is how to switch to degraded operating conditions interactively, by relaxing one or more constraints belonging to H. This problem is addressed after the general structure of the ADR problem is introduced.

5.3.2 ADR Problem Structure

The approach proposed in SaIRA to model the ADR CSP is based on Benders' decomposition [Benders 1962], which has already been used to structure complex combinatorial CSP [Cambazard et al. 2004]. The CSP is divided into two distinct problems (Figure 5.3). The master problem handles hard constraints only and is tackled autonomously by the CSP solver, using constraint programming techniques. This problem is solved first and an applicable but incomplete configuration is produced. The incomplete configuration is refined in the sub-problem, which handles the dynamic constraints and is solved using a mixture of constraint programming techniques, heuristics and and involves the pilot. A novel approach to the solution of the sub-problem is proposed later in this chapter.

Because of the role that the master and sub-problems play in this project, for the sake of clarity the remainder of the thesis refers to the master problem as the *Automated Problem* and to the sub-problem as the *Interactive Problem* (i.e. it involves the pilot). The output of the automated problem (a partial configuration) is the input for the interactive one. It is a requirement in SaIRA that the Automated Problem is consistent:

Definition 5.3.8 *Let $\Delta = (X, D, Z, H)$ be a dynamic constraint network (DCN). Let $S(\Delta)$ be the set of all solutions of Δ. Δ is **consistent** (respectively **inconsistent**) iff $S(\Delta) \neq \emptyset$ (respectively $= \emptyset$).*

This requirement ensures that at any time during the ADR, the solutions of the Automated Problem are valid. In other words, only the propagation of dynamic constraints can lead to unfeasible solutions which require counteractions, such as constraint relaxation, user preferences elicitation.

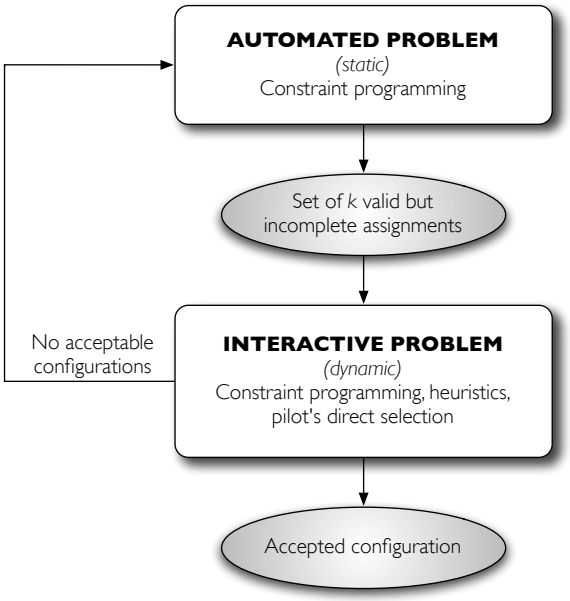


Figure 5.3: Bender's decomposition of the ADR problem. The automated problem is solved using constraint programming techniques only; the sub-problem is solved by a combination of constraint programming techniques, heuristics and pilot's preferences.

5.3.3 A note about the CSP Solver

Several tools for CSP modelling and solutions are available on the internet, for example, [ILOG 2010], [ECLiPSe 2010], [SICStus Prolog 2010]. In this thesis the ADR problem is modelled and solved using Choco [Laburthe 2000], an event-based CP solution engine based on the Claire [Caseau et al. 2002] programming language. The reasons for choosing Choco include the fact that it is open source and free. It is an event-driven engine (particularly suitable for modeling fault-management systems) and it has autonomous explanation generation features that are used in SaIRA as a baseline to generate readily understandable decision support information for the human.

The selection of a specific tool does not limit the generality of the model proposed in this work. In fact, default search algorithms provided by all the solvers previously mentioned are sufficiently similar that they provide a common context for discussing modelling choices without losing generality [Smith 2005].

5.4 Modelling and Solution of the Automated Problem

5.4.1 A model of a small IMA system

By way of example, a model of a small IMA system is described in this section in order to show how the reconfiguration problem is modelled in SaIRA (Figure 5.4).

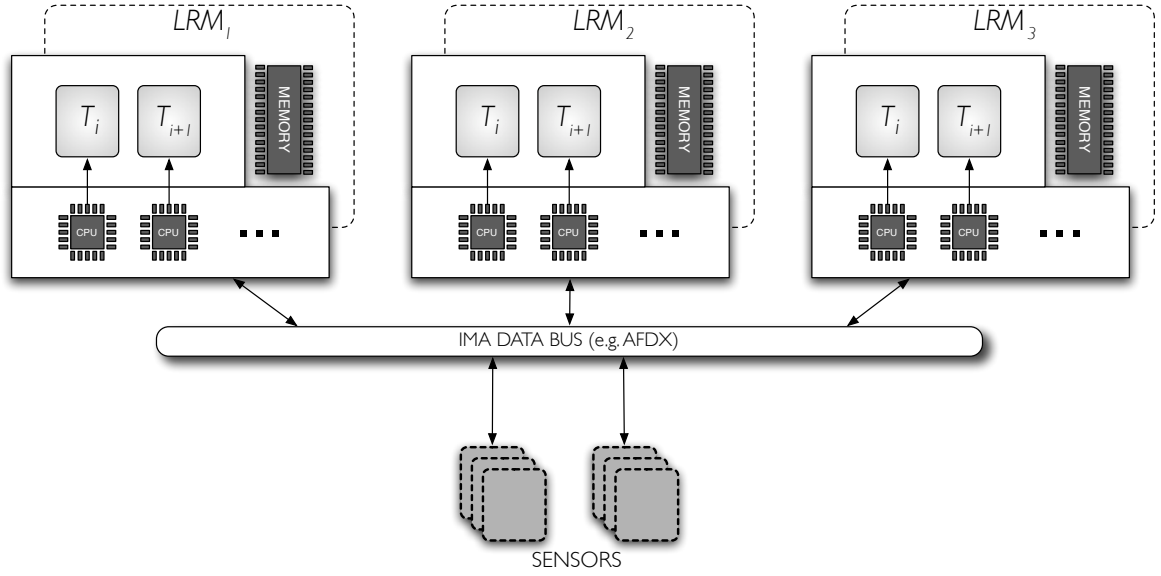


Figure 5.4: Simplified model of an IMA.

Variables

Line Replaceable Modules (LRM): The IMA is composed of three LRM_k, each of which is equipped with a variable number of processors (*cpu*). Each LRM_k has a different amount of memory Mem_k. A number of tasks (T_i) runs on each LRM_k but, for safety reasons, only one task can run on each *cpu*. In this model, cpu_i denotes the processor on which task T_i runs.

Tasks scheduling and allocation: Each task T_i has a start time S_i, a duration p_i, a deadline d_i and a memory consumption m_i. Tasks are considered non-preemptive.

The fact that task T_i is assigned to LRM_k is represented by the boolean variable C_i^k.

A very simple fixed cyclic scheduling policy is implemented in this example. More sophisticated task scheduling and allocation approaches are adopted in real, modern IMAs (e.g. Lee et al. [2000]; Bate and Burns [2003]), however, a cyclic executive allows a clearer exposition on this occasion. The specific problem of applying CP techniques to more complex task scheduling

strategies on distributed networks has been already addressed extensively (e.g. Cambazard et al. [2004]; Hladik et al. [2005]; Weibenbacher et al. [2005]).

The directed graph in Figure 5.5 represents the major execution cycle of four hypothetical tasks T_i running on the system. The small black circles represent a task execution at each iteration of the schedule. The arrows between the circles represent precedence relationships, so that the second time T_1 is executed it requires input from T_2; hence, T_2 must have completed its first execution before T_1 can start its second execution. T_i^n denotes the n-th execution of T_i within the major cycle. It follows that $T_2^1 \rightarrow T_1^2$.

Communication: Tasks communicate through the data bus which has maximum bandwidth H_{bus}. The bandwidth consumed by task T_i to send a message to task T_j is denoted as h_{ij}.

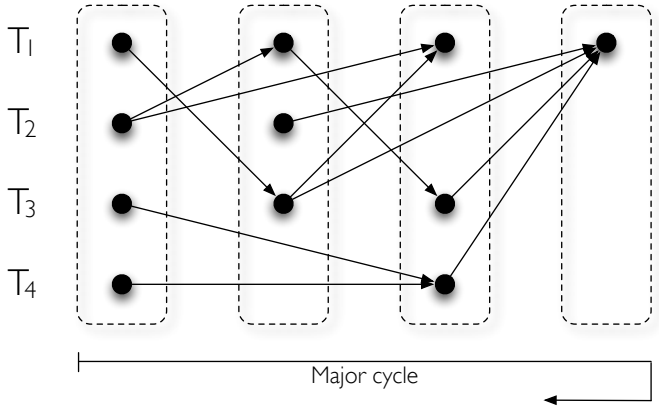


Figure 5.5: Acyclic, oriented graph representing tasks precedence relationships. Black circles represent task execution; arrows represent precedence between two tasks, e.g. $T_i \rightarrow T_j$ means that T_j is dependent on T_i, hence T_j can start only after T_i has completed.

Obviously, no bus bandwidth is consumed if the two tasks that communicate reside on the same LRM but on different processors; the variable x_{ij} contains information about the fact that tasks T_i and T_j share the same LRM (i.e. 0 means they reside on different LRM, 1 means they share the same LRM).

Constraints

For the sake of demonstrating the ease of handling the ADR problem through CP, a few explanatory constraints from the overall problem are introduced here. More sophisticated relationships can be established for real, complex systems.

Memory consumption: a constraint is defined to avoid allocating a number of tasks on a LRM which require in sum more memory than it is available:

$$\text{for each } LRM_k, \quad Mem_k \geq \sum c_j^k \times m_j \tag{5.4}$$

Bandwidth consumption: for each task T_i the following two constraints are set up to model the bandwidth consumption. As previously mentioned, the communication between two tasks that

reside on the same LRM does not require any bandwidth.

$$h_i = \sum x_{ij} \times h_{ij} \tag{5.5}$$

$$cumulative((s_0^0, d_0, h_0), \ldots, (s_0^m, d_0, h_0), \ldots, (s_l^0, d_l, h_l), \ldots, (s_l^m, d_l, h_l), H_{bus}) \tag{5.6}$$

Task allocation: the fact that each task must run on a different CPU is modelled with an *allDifferent* constraint (available in the majority of CSP solvers):

$$allDifferent\ (cpu_1, ..., cpu_n) \tag{5.7}$$

Task precedences: tasks are linked by inter-dependencies. If the input data for T_j^n is the output of T_i^m, then T_i^n must be executed before T_j^m. Denoting with s_i^n the start time of the n-th execution of task T_i (see Figure 5.5), the following constraint is set up:

$$s_i^n + d_i \leq s_j^m \tag{5.8}$$

Channelling constraints: finally, two channelling constraints are added:

$$x_{ij} = 0 \text{ iff } T_j \text{ and } T_i \text{ share the same LRM} \tag{5.9}$$

$$c_j^k = 1 \text{ iff } T_j \text{ is executed on } LRM_k \tag{5.10}$$

Occurrence of a fault

An `EVENT` is modelled as a constraint that affects either a `RESOURCE` or a `FUNCTION` (or both) of the IMA. If the sensors detect and identify a fault to some of the memory banks of LRM_2 in Figure 5.5, reducing the available memory from 100 Mb to 60 MB, this `EVENT` would be implemented by reducing the upper bound of variable Mem_2 from 100 MB to 60 MB, resulting in the following constraint being active in the constraint network:

$$Mem_2 = 60\text{MB} \tag{5.11}$$

Following the occurrence of the `EVENT`, a reconfiguration is triggered and the new value of Mem_2 is taken into consideration by Constraint 5.4. If 60 MB do not suffice for all the functions running on LRM_2, the new configurations proposed will relocate some of the tasks running on LRM_2 to another LRM.

There could be situations in which the designers want specific constraints to be considered only at the occurrence of pre-defined events. For instance, if a series of faults affects the memory banks on Mem_2 and the maximum amount of memory available falls below 30 Mb, the designers might want to deactivate the whole LRM. In SaIRA this mechanism is implemented algorithmically by means of a special class of constraint introduced by van der Linden [2001], **dynamic meta-constraints**, which are "constraints that activate other constraints". This functionality is easily implemented as a set of 'if ... then' rules in Choco.

Up to this point, three main simplifications have been made, which require more attention:

1. **Multiple, applicable configurations available**: multiple, *seemingly similar* (to the pilot) and equally applicable configurations can result from solving the ADR CSP following an unexpected `EVENT`; the pilot must be provided with only a few decision alternatives.

2. **No configuration available**: there could be no solution to the problem given the active set of constraints;

3. **Uncertainties**: uncertainty was not addressed appropriately:

   (a) an assumption was made that practitioners have perfect knowledge about the system they designed and developed (i.e. epistemic uncertainty was ignored);

   (b) it was assumed that all possible changes in the operating conditions could be predicted (i.e. aleatory uncertainty was ignored);

   (c) sensors and other onboard devices used to detect and identify unexpected `EVENT`s were assumed to be perfectly reliable and tuned.

All the limitations listed are discussed in the following sections.

5.4.2 Multiple Configurations Available

The first simplification has a relatively trivial solution. When multiple configurations are applicable, by default SaIRA reduces the options to a few (i.e. two or three) configurations *which carry minimal changes to the current one*. The two or three options selected are shown to the pilot, who has the final choice on which configuration should be applied to the system.

Any solution to the problem represents an applicable configuration; the rationale for favouring minimal changes is, intuitively, the reduction of the risk of failures during the process of transitioning between two different system states.

SaIRA uses an instance of the *Distance-Weighted k-Nearest Neighbour* algorithm to calculate the configurations that bring minimal changes to the system; this decision is debatable and it does not limit the generality of the framework; other algorithms could be employed to perform the same activity.

The algorithm assumes that all the configurations generated by the CSP solver correspond to points in a 3-dimensional space $\mathbb{R}^3$. The three dimensions are: (a) number of constraints added, (b) number of constraints retracted, and (c) the number of constraints modified. The three dimensions are metrics that allow the ranking of configurations in terms of the degree of changes they bring to the current set of active constraints (which represent the current configuration).

For each configuration x, a feature vector is calculated:

$$\langle a_1(x),\ a_2(x),\ a_3(x)\rangle \tag{5.12}$$

where $a_r(x)$ denotes the value of the rth attribute metric of configuration x. The standard

Euclidean distance between two configurations x_i and x_j is defined as follows:

$$d(x_i, x_j) \equiv \sqrt{\sum_{r=1}^{n} (a_r(x_i) - a_r(x_j))^2} \tag{5.13}$$

Let x_q be the query configuration to be classified, and let $f : \mathbb{R}^3 \rightarrow V$ be the classification function, with $V = \{v_1, \ldots, v_k\}$ is the resulting set of k configurations nearest to the current one. For each configuration to be classified, the algorithm calculates the following:

$$f(x_q) \leftarrow \arg\max_{v \in V} \sum_{i=1}^{k} w_i \delta(v, f(x_i)) \tag{5.14}$$

where:

$$\delta(a, b) = \begin{cases} 1 & \text{if } a = b \\ 0 & \text{otherwise} \end{cases} \tag{5.15}$$

and:

$$w_i \equiv \frac{1}{d(x_q, x_i)^2} \tag{5.16}$$

The term w_i allows weighting the contribution of each of the k neighbour configurations according to their distance to the current configuration, giving greater weight to closer neighbours.

In the majority of cases two or three configuration options are shown to the pilot, hence the algorithm is configured accordingly (i.e. as 2- or 3-Nearest Neighbour).

5.4.3 No Configuration Available

In over-constrained situations, one or more constraints must be relaxed in order to find sub-optimal solutions. *Chronological backtracking* is the basis of several algorithms proposed in the literature for the autonomous management of over-constrained CSP. A depth-first search is performed through the search space and the variables are instantiated sequentially; all the relevant constraints are checked for consistency after each variable is instantiated and, in the case of any violations, the algorithm backtracks to the last variable that still has one or more values available [Tsang 1993].

Several improvements and alternatives to chronological backtracking have been proposed, all of them designed for the automated management of over-constrained situations, e.g. *Dynamic Backtracking* [Ginsberg 1993], *Conflict-Directed BackJumping* [Prossner 1995], *Partial Order Dynamic Backtracking* [Ginsberg and McAllester 1994], *Dependency-Directed Backtracking* [Stallman and Sussman 1977], *Generalised Dynamic Backtracking* [Bliek 1998].

One of the standpoints taken in this research is that experienced decision makers usually possess deeper knowledge of the decision options, which goes beyond system models. The decision maker could act effectively during unforeseen situations and manage to recover from system failure, if properly informed of the specifics of the impasse that the system has encountered.

When the constraints relative to the current operating conditions make the CSP associated with the ADR problem over-constrained, it becomes necessary to guide the pilot through the complex decision of selecting which constraints, amongst hundreds, should be relaxed, in order to find the configuration that best fits the new operating conditions. In CSP terms, this corresponds to the problem of identifying the correct **repair action** [Elkhyari et al. 2002].

In SaIRA, the objective of a decision support message in support of a repair action is to help the pilot answering the question "why should I apply `Configuration A` instead of `Configuration B`?" Accepting one configuration instead of another corresponds to accepting one repair action instead of another (e.g. sacrifice the Elevator Feel System instead of the Waypoint Generator System).

The constraint-based reconfiguration system must go through the following three steps in order to provide support to the pilot:

1. Generate a number of *configurations* using the current active constraints;
2. Generate a set of *repair actions* for the inconsistencies;
3. Generate *recommendations* for the repair actions.

The first step has already been briefly discussed in the previous section; extensive research has already been performed in this domain. The second and third steps are more complicated and interesting for this thesis. The generation of recommendations for each repair action[2] encompasses the tasks of generating explanations (i.e. why should the pilot go for repair action A instead of repair action B?) and implications (i.e. what are the consequences of going for repair action A?). The accomplishment of these tasks required the development of two novel algorithms, namely `wsm decision-repair` and `SaIRA-XPlain`. For reasons of clarity of exposition, the details of the algorithms development process, the description of their features and the analysis of their performance have been relegated to Appendix B; the focus of this chapter is on the overall design of the system, however the details are given in the appendices for the sake of reproducibility.

The research hypothesis states that effective decision support information for ADR decisions should include explanations, implications and an assessment of the uncertainty embedded in the recommendations of the system. Up to this point explanations and implications have been discussed; Section 5.5 will focus on the last piece of information missing from the picture: the uncertainty assessment. As with explanations and implications, the exposition will be kept qualitative in this chapter; detailed information about the algorithms used to calculate uncertainties are provided in Appendix C.

Figure 5.6 provides an overall view of the function of each piece of decision support information in relation to the configuration suggestion proposed by the system. Both explanations and the uncertainty assessment are used by the algorithms as inputs to generate the suggestion; the implications are calculated as a result of the potential application of each configuration suggestion. The research hypothesis predicts that the contemporaneous access to all the pieces of support information should lead to an informed and effective avionics reconfiguration decision.

[2] Each repair action leads to a specific configuration.

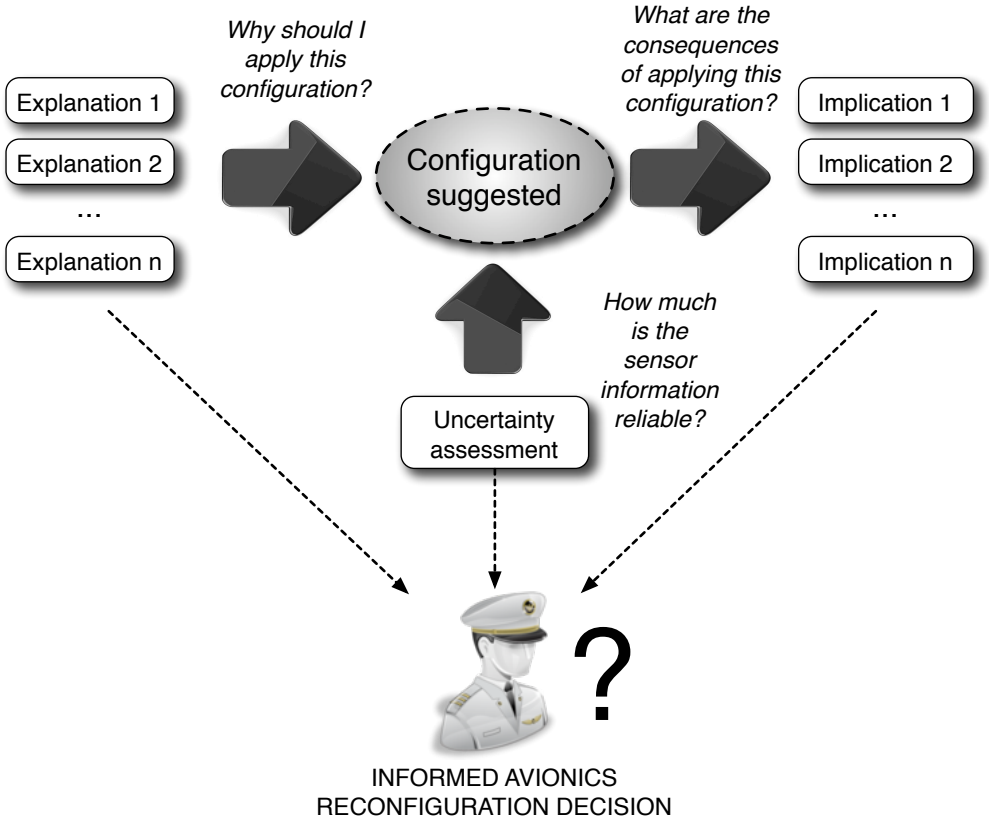


Figure 5.6: Informed avionics reconfiguration decision supported by SaIRA decision support information.

Section 5.6 will describe the overall SaIRA framework and its components. Section 5.7 will reveal how the user interface has been designed and how explanations, implications and uncertainty assessment, as generated by SaIRA, are physically presented to the pilots on the cockpit displays.

5.5 Multi-Sensor Data Fusion and Uncertainty

Data fusion techniques combine data from multiple sensors, and related information from associated databases, to achieve improved accuracy and more specific inferences than could be achieved by the use of a single sensor alone [Llinas and Hall 1998].

Murphy [1996] gives the following definition of **intelligent sensor fusion**:

Definition 5.5.1 *Intelligent sensor fusion is defined as a process which can autonomously gather observations from multiple sensors and combine them into a single, coherent percept (execution), adapt the combination process to major environmental strategies for observing the percept (configuration).*

At present, multi-sensor data fusion techniques are employed in many safety-critical engineering domains, including robotics, machine vision, monitoring of manufacturing processes, medical

applications, smart weapons, fault diagnosis, satellite and aircraft avionics (see Hall and Llinas [1997] for an extensive case history).

Besides the benefits extensively discussed in the literature, the adoption of sensor data fusion techniques on-board modern aircraft can possibly have consequences on the avionics dynamic reconfiguration process which require attention. Let us consider the following accident:

Accident 5.1 *On the 24^{th} August 2001 an Airbus A330-200 covering the Air Transat 236 route between Toronto and Lisbon developed a fuel leak in a fuel line to its right engine [Aviation-safety.net 2010]:*

1. *Flight TS 236 took off from Toronto at 0:52 (UTC)*

2. *There were 293 passengers and 13 crew members on board. The aircraft was an Airbus A330 manufactured in 1999, configured with 362 seats and placed in service by Air Transat in April 1999. Leaving the gate in Toronto, the aircraft had 47.9 tonnes of fuel on board, 5.5 tonnes more than required by regulations.*

3. *At 04:38 UTC (estimated), a fuel leak started in the area of engine no. 2 (right engine).*

4. *At 05:16 UTC, a cockpit warning system sounded and reported low oil temperature and high oil pressure on engine no. 2. There is no obvious connection between oil temperature or pressure problems and a fuel leak. At first, Captain Piché and co-pilot DeJager suspected these warnings were computer bugs and communicated with their maintenance control center.*

5. *At 05:36 UTC, the pilots received a warning of fuel imbalance and diverted fuel from the port (left side) wing tanks to the starboard tanks, which were showing close to empty. Because the fuel leak in the starboard engine had still not been diagnosed, this diversion had the effect of sending fuel to the leak and causing further loss.*

6. *At 05:45 UTC, it became clear that fuel was dangerously low.*

...

7. *At 06:13 UTC, 28 minutes after the emergency declaration and 135 miles (217 km) from Lajes, engine no. 2 on the right wing flamed out, exhausted of jet fuel.*

...

8. *At 06:26 UTC, engine no. 1 flamed out*

...

9. *Thanks to the skills of the pilots, the plane made a fortuitous emergency landing at 6:46 (UTC) with several people injured but no deaths.*

Some *evidence* of a fuel leak was detected by the network of sensors installed on the aircraft. However, the information was not fused into a single, helpful fault message and too much inference was left to the human operators. Leading aviation and space companies are investing significant amounts of money in the development of multi-sensor information fusion systems. Examples are the Multi-Sensor Integration (MSI) avionics software by EADS [Belz 2005] and Block 3.0 sensor fusion avionics software by Lockheed Martin [Caires and Stout 2002].

Different sensors, characterised by different reliability, precision and fit to assess a specific fault, could generate different, and possibly contrasting, inferences. This implicitly injects a degree of uncertainty into the conclusions reached by the fault management logic which, as seen in the previous chapters, drives the ADR system. In this scenario, a major avionics reconfiguration (in which the functionality of the aircraft is changed) could be triggered on the basis of uncertain information, depending on how the logic is implemented. In some cases, a major reconfiguration could lead to severely degraded functional arrangements with reduced safety. Should the pilot be informed of the degree of uncertainty embedded in a fault assessment made by the system before triggering a reconfiguration? Would this type of information improve his or her situation awareness?

In this regard it is interesting to mention Hunter [2006] which investigates the effects of risk perception among general aviation pilots. Trying to explain pilot behaviours that lead to accidents or incidents, Hunter brings evidence for two conclusions that are extremely relevant for our research interest:

- One explanation for behaviour that leads to an accident or incident is that the pilot did not perceive the risk inherent in the situation and hence, did not undertake avoidance or other risk-mitigating actions;

- Another explanation is that when individuals correctly perceive the risks involved in a situation, some may elect to continue because the risk is not considered sufficiently threatening. Those individuals would be described as having a greater tolerance or acceptance of risk compared to the mainstream.

The position of this thesis is that a decision support system should quantify the uncertainty to help the pilot cope with higher order reasoning processes in an unstructured environment. We claim—and empirically verify in Chapter 6—that an assessment of the uncertainty (which is made up of risk and ambiguity as shown in Section 4.1.1) embedded in safety-critical information coming from the system would lead to more informed—and hence safer—decisions and would improve pilot situation awareness.

Hidden risks were not exposed in Accident 5.1. No textual information was generated by the system, a set of alarms was triggered and pilots were directed to the Flight Manual. To worsen the situation, the Airbus A330 manual, when describing the procedure to reconfigure the fuel between tanks (which is what the pilots did on that occasion), contains the following sentence:

```
CAUTION: do not apply this procedure if fuel leak is suspected.  Refer
to FUEL LEAK procedure.
```

This information could have meant that the accident was avoided, but even if it is reported in the manual, it is presented as a caution and not as a step in the standard procedure. For this reason, in that critical context and with a limited decision time budget, the pilots did not pay attention to it.

If the system had been able to tell the pilots something like this:

```
[low oil temperature] + [high oil pressure]
                        ↓
          fuel leak [uncertainty:  30%]
```

It might have sowed a doubt in the pilots' minds and, as a consequence, the accident could have been avoided.

The problem of fusing information in multi-sensor data networks has had a lot of attention in the aerospace and defence domains in the last couple of decades. Recently a series of criticisms of the probabilistic characterisation of uncertainty have appeared in the literature [Sentz et al. 2002], claiming that traditional probability theory is not capable of capturing epistemic uncertainty (see Appendix C for more details).

The application of traditional probabilistic methods to epistemic uncertainty is often known as Bayesian probability. Bayesian approaches to Fault Detection, Identification and Recovery (FDIR) are widely employed in the aerospace domain [Paakko et al. 2001; Guiotto et al. 2003]. These techniques require practitioners to have precise information concerning the probability of all events. When this is not possible, the uniform distribution function is often used, justified by Laplace with the Principle of Insufficient Reason [Smets 1994]. This can be interpreted as, all simple events for which a probability distribution is not known in a given sample space, are equally likely. The result is hiding potentially dangerous information from the pilot.

In response to this problem, significant research has been performed on the application of Evidential Reasoning (ER) techniques. ER is a reasoning framework based on the *theory of belief functions* conceived by Dempster [1967] and further developed by Shafer [1976]. It is a generalisation of probability theory that allows the specification and handling of *degrees of precision* as well as *degrees of uncertainty*, which go beyond the limits of classic Bayesian methods.

The theory allows belief to be assigned to individual propositions in the space or to disjunctions of propositions or both. Belief assigned to a disjunction explicitly represents a *lack of sufficient information* to enable more precise distribution. This allows belief to be attributed to statements whose granularity is appropriate to the available evidence.

Murphy [1998] investigates the application of ER-based techniques to multi-sensor data fusion for robotics applications; Sarma and Raju [1991] apply the theory to the problem of multi-sensor based target identification on the battlefield; Strat [1987] addresses the problem of generating explanations of the ER-based reasoning of an automated system; Yu et al. [2004] focus on the problem of fusing multi-sensor information coming from airborne sensor networks for target tracking and identification in military applications. This short list is far from being exhaustive, but it reveals plenty of well-established work that can be applied to the problem of generating uncertainty figures related to the fault assessment information that triggers a reconfiguration and that is directed

to the pilot for decision support.

In the experiments discussed in Chapter 6, the ER-based approach for multi-sensor fusion introduced by Yu et al. [2004] is used to simulate sensor data fusion and generate the uncertainty figures.

The choice of the algorithm developed by Yu et al. is not absolute, other solutions available in the literature could be applied as well. In fact, the topic of sensor data fusion has been briefly touched on in this section in order to introduce the necessity of providing the pilot with uncertainty figures and to show that the technology to generate this type of information programmatically is already available, supporting the practicality of the approach taken in SaIRA. More information concerning how ER ideas in the literature are specifically applied to SaIRA are in Appendix C.

5.6 SaIRA

All the technology discussed in this chapter and in Appendices B and C is integrated in the architecture presented in Figure 5.7. SaIRA comprises three main components, the Sensor Fusion Executive (SFE), the Reconfiguration Executive (RE) and the Decision Support Executive (DSE).

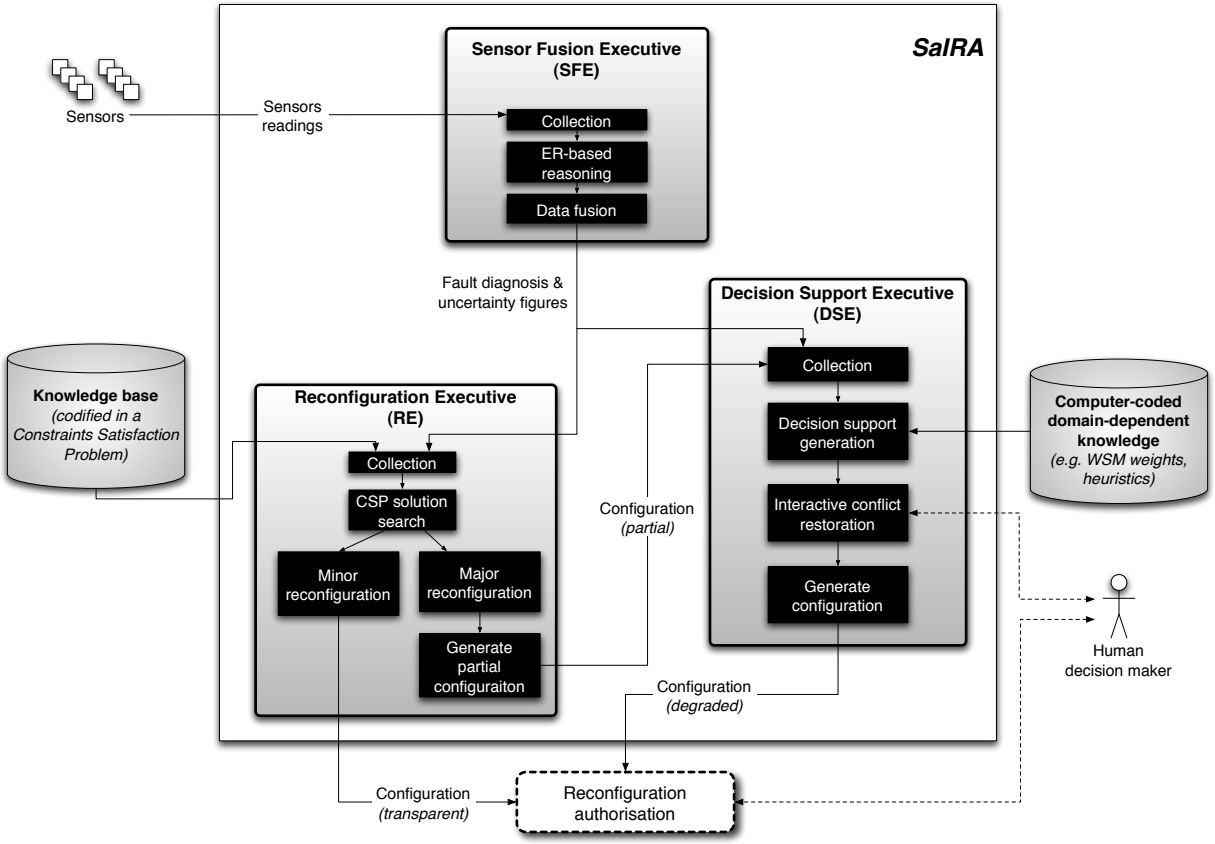


Figure 5.7: Safe and Interactive Reconfiguration Architecture (SaIRA).

The **Sensor Fusion Executive** receives readings from the on-board sensors and uses heuristics based on the Evidential Reasoning paradigm (discussed in Appendix C) to merge the data and

calculate an assessment of the reliability of the fault diagnosis. Fault diagnosis and uncertainty information are sent to both the RE and DSE for further processing.

The main purpose of the **Reconfiguration Executive** is 'generating applicable configurations'. Fault diagnosis information obtained from the SFE is integrated with domain-dependent knowledge codified in the CSP associated with the ADR problem. The constraint-based solver makes a first attempt to identify a solution (i.e. applicable configuration); in cases when it is successful (i.e. minor reconfiguration, the system functionality is not altered), no decision support information is generated, hence the DSE is not called and the pilot is asked to confirm a 'transparent' reconfiguration. In cases of major reconfiguration (i.e. the current system functionality must be altered and it is necessary to switch to a degraded operating mode), the partial solution generated by the RE is passed to the DSE for further refinement, with the support of the pilot (Sections 5.3 and 5.4).

The **Decision Support Executive** receives fault diagnosis information from the SFE and a partial configuration from the RE. Within this module, the `SaIRA-XPlain` algorithm (Appendix B, Section B.2.3), which embeds the `wsm decision-repair` algorithm (Appendix B, Sections B.1.5.2 and B.1.6), manages the interactive process of conflict repair. The DSE generates decision support information for the pilot and calculates applicable configurations on the basis of the inputs coming from SFE, RE, the pilot, and a domain-dependent knowledge base, which is codified in the constraints weights used by the `WSMrank` function (part of the `wsm decision-repair` algorithm).

For the reasons given in Section 3.2.3, the pilot is required to confirm the application of the new configuration to the system, regardless of the path followed for the generation of the configuration to apply, i.e. either through a minor or major reconfiguration.

5.7 SaIRA Interface Design

The previous sections in this chapter along with Appendices B and C describe how the decision support information is algorithmically generated by SaIRA. This section discusses how the interface of SaIRA has been designed and the decision support information is graphically presented. The work discussed in this section has been performed with the support of two Boeing 737-900ER pilots; all the design decisions taken have been preventively discussed and agreed with them, a process that led to the development of graphics which are very similar to cockpit instruments currently available on the Boeing 737-900ER aircraft.

Modern cockpits have evolved from "clock-like" instruments to "glass cockpits"—the cockpit features electronic, reconfigurable instrument displays similar to the personal computer glass monitor. Because the displays can be reconfigured, their content can be tuned whilst airborne to show flight information as required by the phase of flight and as needed by the pilot. Like other modern glass cockpit applications, SaIRA has been designed to temporarily replace the content of some of the cockpit displays with its decision support information at the occurrence of an event that triggers an avionics reconfiguration.

Some basic information about modern glass cockpit is presented hereinafter (Section 5.7.1), in preparation for the discussion of how the interface of SaIRA has been designed (Section 5.7.2).

5.7.1 The Electronic Flight Information System

The primary component of modern glass cockpits is the Electronic Flight Information System (EFIS), which is responsible for the processing and display of all the flight information during all the phases of flight. It comprises two displays: the Electronic Horizontal Simulation Display (EHSI) and the Electronic Altitude Director Indicator (EADI). The third type of reconfigurable display in modern glass cockpits is the the aircraft systems and engine performance information display: this latter display is known as EICAS (Engine Indicator and Crew Alert System) on Boeing aircraft and as ECAM (Electronic Centralised Aircraft Monitor) on Airbus aircraft. Figure 5.8 shows the glass cockpit of a real Boeing 737-900ER and the location of the displays mentioned above. The figure shows clearly that both pilot and co-pilot have their own EHSI and EADI displays whilst they share the EICAS. Because the Boeing 737-900ER is used in the experiments presented in this thesis, the remainder of this section refers to this aircraft specifically.

The EHSI displays the flight progress of the aircraft on a plan view map or the flight plan on a map oriented to true north (Figure 5.9). The display can also serve as a weather radar when the pilot activates this functionality.

The EADI displays information about attitude (pitch and roll), flight director commands, localiser deviation and glide slope deviation (Figure 5.9). Additionally, depending on the system mode, the display can also show information relating to autopilot, airplane speed, pitch limit, Mach, ground speed, radio height alert, decision height and radio altitude.

The EICAS displays engine and selected subsystems indications as well as crew alerting functions. Typical engine parameters shown in this display are revolutions per minute, temperature values, fuel flow and oil pressure.

The EFIS of the Boeing 737-900ER is called Common Display System (CDS) and is produced by Honeywell International Inc. The CDS, shown in Figure 5.8, comprises six flat panel LCD display units. The six identical ARINC D-size display units are 20.32 by 20.32 cm. Each display has a 16.97- by 16.97-cm usable display area. The four display units located outboard and inboard on the captain's forward panel and first officer's forward panel are the EHSI and EADI. The upper- and lower-center display units on the central panel are the EICAS.

5.7.2 Designing the interface of SaIRA

The EFIS shows information on a "need-to-know" basis. In line with this approach, SaIRA has been designed to display reconfiguration decision support information only when a reconfiguration is required; in nominal conditions, no SaIRA information is displayed on the cockpit.

The first design decision taken was the choice of which displays should be used to display the SaIRA information. A major requirement is to keep the information that is critical for the flight always accessible, even during avionics reconfiguration. A consultation with the pilots revealed that basic information provided by the EHSI and EADI displays is always available in their analogic counterpart on the cockpit, as shown in Figure 5.9. In fact, the HSI and ADI instruments are supposed to be used by the pilots when the EFIS fails to generate the graphics. As a result, it was decided that when a reconfiguration is required, SaIRA temporarily replaces the content of both

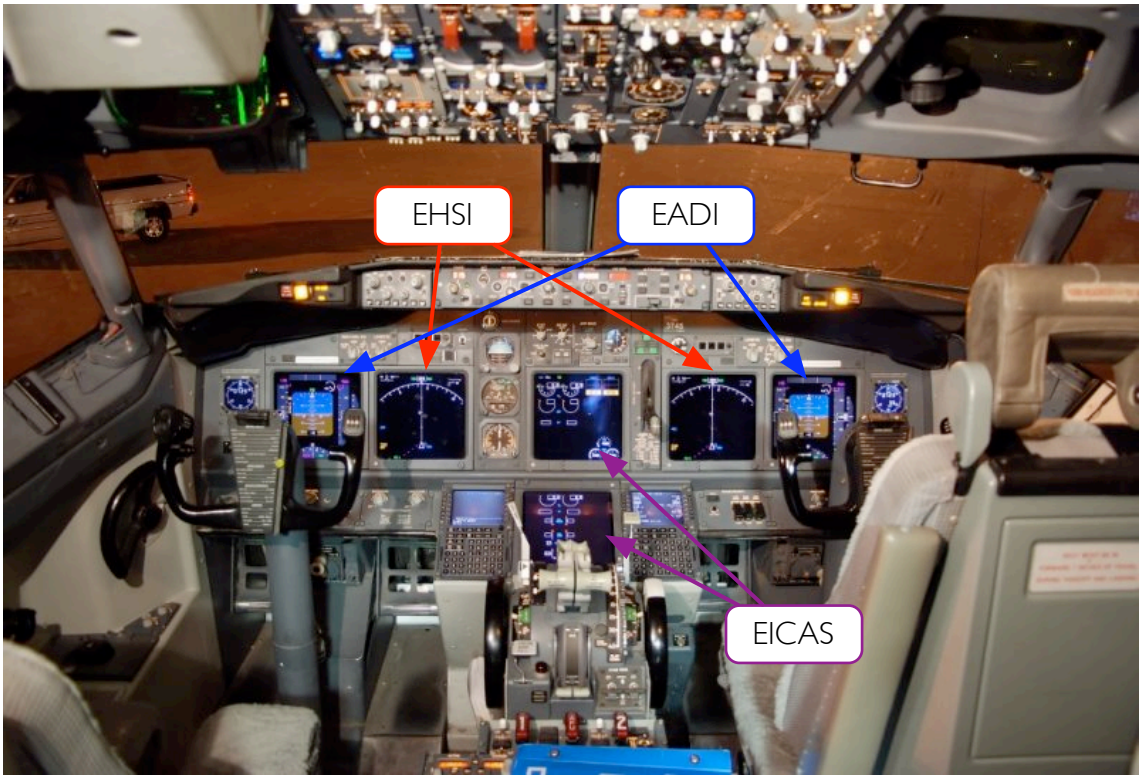


Figure 5.8: Cockpit of the Boeing 737-900ER [original image freely available at www.flightgear.org].

the EHSI and EADI displays. The EICAS displays are not modified in any way by SaIRA, leaving engine information and other warning from the avionics subsystems always acessible.

An analysis of the scaled pictures of the cockpit displays from the Boeing 737-900ER manual reveals that the font size of the text shown on both EHSI and EADI ranges between 34mm to 68mm. In line with other instruments, the standard font used by SaIRA has size 60 mm; the virtual buttons used by the pilots to select and accept the configurations (see Figure 5.11) have size 1 cm.

The Boeing 737-900ER EFIS has the following warning colours scheme [Brady 1999]:

- White - Informative text
- Red lights - Warning - indicate a critical condition and require immediate action.
- Amber lights - Caution - require timely corrective action.
- Blue lights - Advisory - eg valve positions and unless bright blue, ie a valve/switch disagreement, do not require crew action.
- Green lights - Satisfactory - indicate a satisfactory or ON condition.

The information displayed by SaIRA has been adapted to the scheme above. White is used for the informative text. The fault description is written in red colour. The buttons to accept a satisfactory configuration are green and the buttons to switch between the suggested configurations are magenta, to differentiate them from the acceptance buttons.

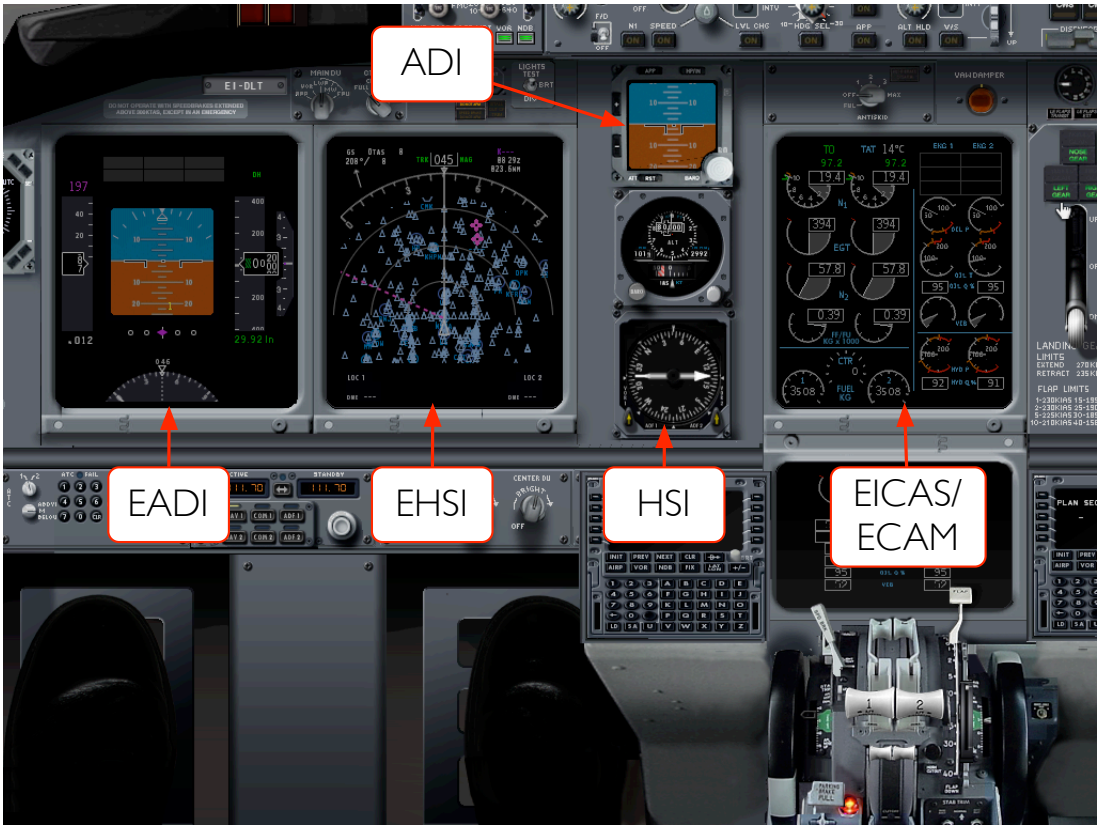


Figure 5.9: Captain-side of the cockpit of the Boeing 737-900ER (X-Plane flight simulator). Both the Attitude Director Indicator (ADI) and the Horizontal Situation Indicator (HSI) have an electronic counterpart—EADI and EHSI respectively—in modern glass cockpits.

The previous sections of this chapter discussed how the decision support information is generated in SaIRA and the importance of explanations, implications and an assessment of the reliability of the sensor information. A way to present this information on the cockpit displays in natural language, for each decision alternative, in an effective way, must be identified.

A review of the literature reveals that the majority of Natural Language Generation (NLG) systems employed in interactive problems and decision support structure the information using one of the following ways, or a combination of them:

- **Elaboration**: a set of messages elaborates on the information in another set of messages;
- **Exemplification**: a set of messages provides an example of the fact stated in another set of messages;
- **Contrast**: a set of messages provides contrasting information to that provided in another set of messages;
- **Narrative sequence**: a set of messages communicates a time-ordered sequence of events.

As discussed with the pilots, sophisticated constructions represent a risk for SaIRA, because the message delivered could become too complicated to be processed in real-time. Given that

the pilot is required to choose between two or more configurations, it was decided to structure the messages using *contrasts*; the pilot is provided with the characteristics of each configurations, facilitating an informed comparison between them.

Another important aspect of the structure of recommendations is the approach adopted for the content design, *bottom-up* or *top-down*. In the former case, all the low-level parts of the information that needs to be realised for the user are merged into a final message. On the constrary, in the top-down approach, a semi-fixed pattern is superimposed and the pieces of information are adapted to it.

The information delivered by SaIRA must have a standard structure for the reasons given above, therefore the top-down approach is naturally more suitable. SaIRA follows a *schema-based* method to structure the information. Dale and Reiter [1995] define a **schema** as "a pattern that specifies how a particular document should be constructed from constituent elements, where those constituent elements may be individual messages or, recursively, instantiations of other schemas". Schemas allow the enforcing of a pre-defined skeleton for the message, to deliver and refine it at runtime; this approach is enough for the needs of the decision support information generation mechanism of SaIRA described so far.

The schema shown in Figure 5.10 is defined for the decision support messages delivered by SaIRA. As shown later in this section, buttons are available under the schema which allow the pilot to switch from the description of one configuration to another, enabling them to *contrast* the content of two configuration suggestion messages. The order of the pieces of information, from the top of the *schema* to the bottom, was devised in collaboration with the pilots. The first information that pilots want to read is the nature of the fault which has just happened followed by the assessment of the reliability of the diagnosis; this allows for the rapid understanding of what is happening to the system and enables decisions on how to handle the following information on the display. The next preferred piece of information are implications: understanding what are the consequences of each configuration alternative is critical to make an informed decision rapidly. Finally, explanations should be particularly effective to increase situation awareness but they will likely require more time to be processed (this is empirically assessed in Chapter 6), therefore it was decided to leave them as the last piece of information presented.

The 16.97- by 16.97-cm usable display area represents a limit to the amount of text that can be written on the display. With reference to Figure 5.10, on the top of the display one or more evidence of fault from the sensors are connected by the "+" symbol and are linked to the fault diagnosis using a vertical arrow to signify causality. The fault diagnosis block occupies up to 4.5 cm (vertical) of space. It was decided to display up to three implications and up to three explanations for each configuration alternative. Appendix B shows that each decision alternative could potentially have a large number of explanations and implications; the decision support information generation algorithms developed for this thesis are designed to filter this information to extract the most important information for the current situation and, for the experiments of Chapter 6, they have been tuned to show up to three explanations and up to three implications.

Appendix B reveals that the DSS algorithms developed use a pre-defined number of metrics to evaluate the impact of each configuration alternative on the system functionality. Typical metrics

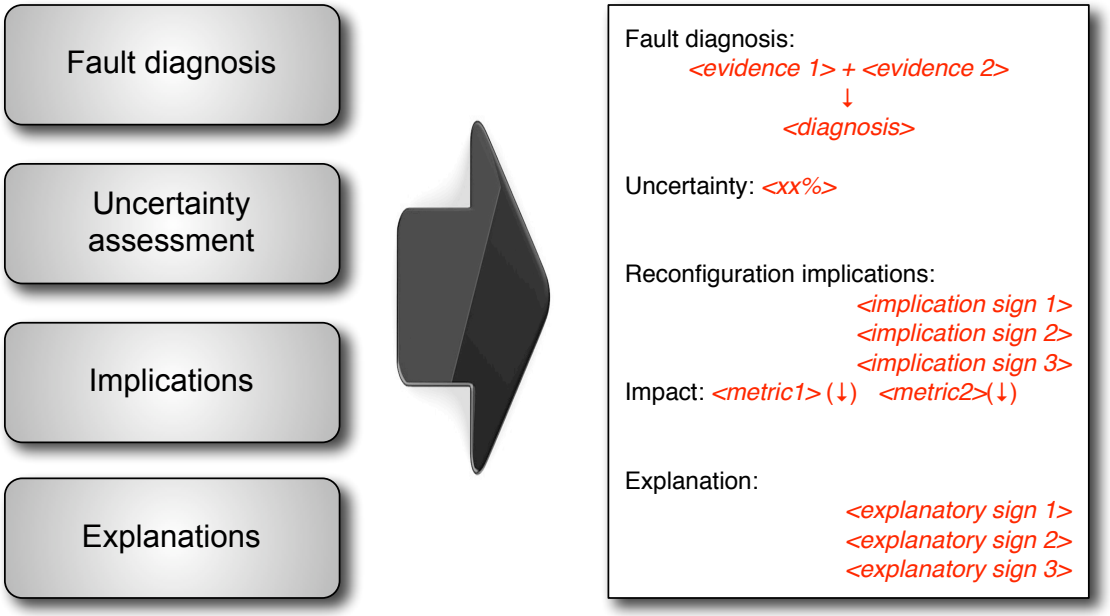


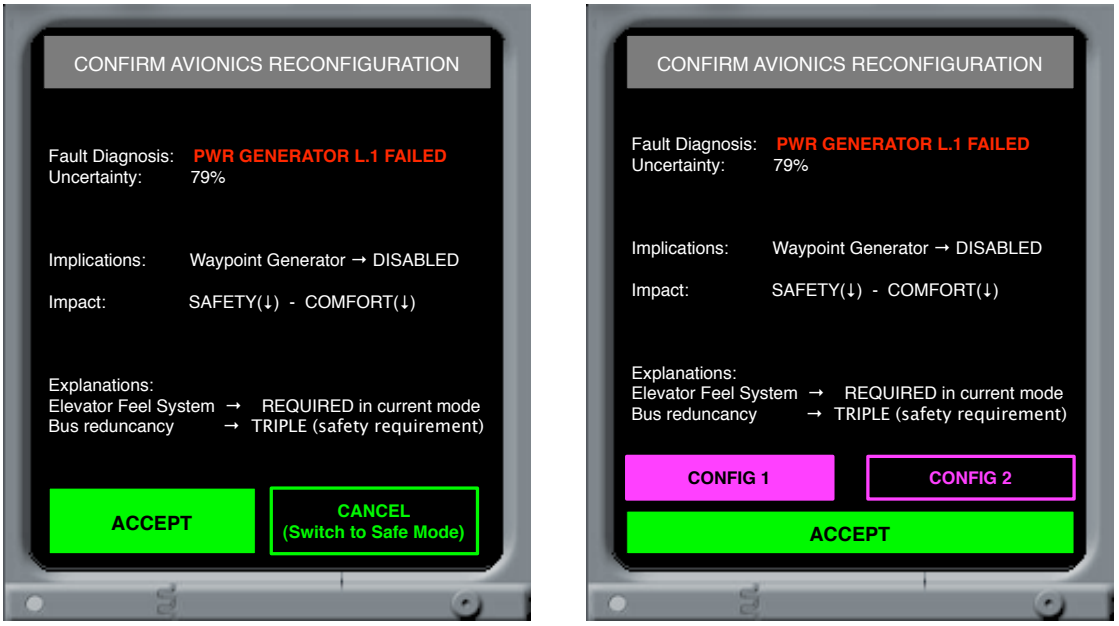
Figure 5.10: Decision support information schema used in SaIRA. The portions in red colour are filled with information generated at run-time.

are safety, performance, instrument usability. Because of the space constraints, the algorithms have been tuned to show up to two metrics and the direction of their impact; for example, is the suggested configuration decreases the degree of performance of the system, the following string will be shown: `PERFORMANCE (↓)`.

The design described so far led to the interface shown in Figures 5.11.a and 5.11.b, which are taken from one of the experiments discussed in Chapter 6. Figure 5.11.a represents the case in which the pilot is provided with one configuration only: she can decide to either apply it or switch to Safe Mode. Figure 5.11.b is representative of the case in which two configuration alternatives are available: the decision support information is identical to the Figure 5.11.a but two magenta, virtual buttons appear in the bottom of the display that allow the pilot to browse the characteristics of each configuration alternative; when she is ready to reconfigure, the pilot clicks on the green 'ACCEPT' button to trigger the reconfiguration process.

In addition to the textual information described so far, which is shown in the EHSI, SaIRA also provides the pilot with a graphical representation of the sub-systems affected by the fault (in the EADI display). A typical example is provided in Figure 5.12, in which a fault caused by an over-heat to the Integrated Drive Generator (IDG) on the left-1 engine is depicted.

Pilots stated that for complicated faults they sometimes consult the aircraft manual in order to have a better understanding of the situation and of the consequences on other sub-systems. In this light, SaIRA has been designed to automatically show schematics of the sub-systems affected by the fault as they appear in the aircraft manual; actually, the schematics shown in Figure 5.12 are taken from the Boeing 737-900ER aircraft manual. Additionally, the sub-systems directly affected by the fault are circumscribed by a red rectangle.



(a) One ADR alternative plus Safe Mode. (b) Two ADR alternatives.

Figure 5.11: SaIRA decision support information on the EHSI display when the pilot get to choose between (a) one alternative and Safe Mode, or (b) between two alternatives. The fault in question is the failure of the power generator driven by the left engine.

5.8 Chapter Summary

This chapter deals with the following topics:

- The 'SaIRA ontology', a meta-model for an interactive SCMS dynamic reconfiguration process was introduced. The ontology facilitates the mapping of a human-interpretable description of the ADR process onto computer-encoded information (i.e. a Constraint Network) which is used as a knowledge-base for the ADR algorithms proposed.

- After introducing background information about state-of-the-art constraint programming technology, the approach used in this thesis to structure and solve the ADR problem is qualitatively discussed. For the sake of clarity of exposition, all the details about how the ideas have been technically implemented are provided in Appendices B and C, allowing for the reproducibility of the experiments. The details also include the presentation of two novel algorithms for automated generation of decision support information and the performance analysis of one of them.

- The problem of generating information about the uncertainties embedded in the fault management information produced by the system is briefly discussed; the technology required to achieve this objective is already available in the literature but, to the best of our knowledge, it has never been applied in the context of safety-critical decision support on-board modern aircraft. The technology has been reviewed, discussed, adapted and integrated with SaIRA to show the practicability of the approach proposed. Qualitative information is provided in

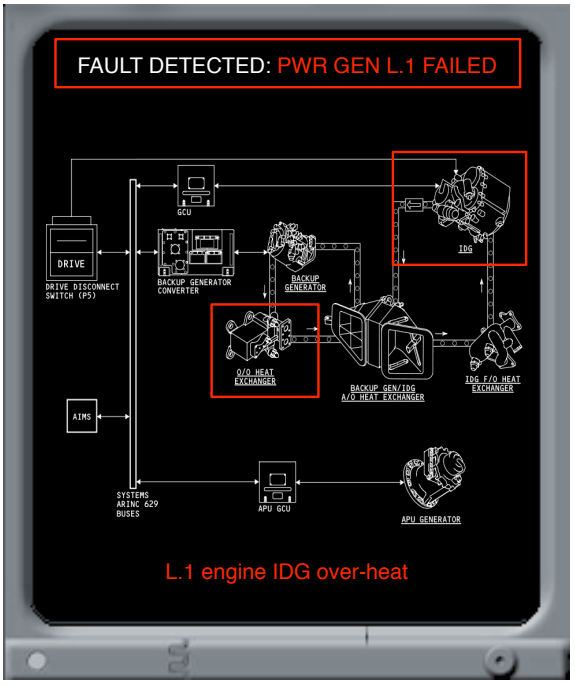


Figure 5.12: Schematics that describe the sub-systems mainly affected by the fault. This information is shown on the EADI display of the Boeing 737-900ER cockpit.

this chapter; the technical details have been relegated to Appendix C.

- All the technology discussed in the chapter is organically amalgamated into the SaIRA framework, which is finally presented from a general point of view. The functionality of the three main components is described and linked to the material previously presented.

- The process of design of the user interface of SaIRA is discussed in detail. The representation of the decision support information on the cockpit displays has been designed with the support of two aircraft pilots qualified to fly the Boeing 737-900 aircraft. The user interface has been designed in accordance to the graphical shape and logic of current glass cockpit applications in order to provide the novel decision support tool proposed in this thesis with a realistic appeal.

Chapter 6

Empirical Evaluation

"The most savage controversies are about matters as
to which there is no good evidence either way."
—Bertrand Russell
'An Outline of Intellectual Rubbish' (1943)

6.1 Overview of the Experiments

This chapter describes the series of experiments performed to verify the claims advanced in Chapter 4 (see Section 4.4 for the complete list) on pilot decision behaviour during ADR decisions and the effectiveness of the decision support framework proposed in Chapter 5. The effectiveness of the hypothesised DSS user profile is also assessed.

As mentioned in Section 4.3, we ran seven experiments, aimed at investigating different aspects of the following general research topics:

1. *How pilots respond to ADR decisions, regardless of the design of the decision support information* (Experiments A, B, D and F). The experiments performed in this area examine the following decision biases: trust, complacency, risk perception, time pressure and stress.

2. *How the design of SaIRA decision support information influences the pilot response* (Experiments C, E and G). Different aspects of the decision support information generated by SaIRA are manipulated in order to characterise the reactions of the pilots. The aspects manipulated include information framing, information content, graphics availability and reliability figure availability.

Thirteen civilian pilots from two European airlines, certified to fly the Boeing 737 aircraft, participated in this study. At the time of writing, eleven pilots were resident in the United Kingdom and two in Italy. One of the pilots, of Italian nationality, served as a captain on the B737 and is now in retirement. All pilots were aged between 31 and 68 at the time of the experiment; twelve of them are male, one is female.

Initially, two airlines were contacted and asked for their interest in participating to this study; both of them refused. As a consequence, we proceeded to contact the pilots informally, "as indi-

viduals", exploiting social networks (i.e. networks of contacts, not online). Initially, before their consent to participate to the study, pilots were provided with a overall view of the research during informal meetings hold in public places such as pubs. All of them required complete anonymity as a core prerequisite to their participation to the study. Only after their informal consent to participate to the research, we proceeded to a more formal relationship and training, as explained in detail later in this chapter.

Unfortunately, it is extremely difficult to find experienced pilots willing to participate in research experiments. Furthermore, experiments in which objective measuring techniques like eye-movement analysis are employed require considerable time to be performed. These are two of the reasons why a limited number of individuals took part in this study. This is a general issue in the research community which, on condition that the right statistical tools are employed for analysis of the results, does not diminish the significance of the results obtained. The literature contains many examples of research performed on a very small number of pilots, which produced well-established results that are currently applied in the industrial arena (Sarter and Woods [1994]; Hutchins, Morrison and Kelly [1996]; Flemisch and Onken [2000]; Singer and Dekker [2000]; Mumaw et al. [2001]; Diez et al. [2001]; Arthur et al. [2003]; Huemer et al. [2005]; Hayashi et al. [2006]; Trujillo et al. [2008]; Dao et al. [2009]; Taylor et al. [2009]; Hayashi et al. [2009] *inter alia*). The majority of studies involving experienced pilots and employing eye-movement analysis techniques seem to be based on a small number of participants.

Section 6.2 describes the metrics used. Apparatus and materials are presented in Section 6.3. Section 6.4 accounts for the training received by the pilots for the use of the flight simulator, SaIRA, NASA-TLX (the technique we used to assess the pilot workload) and SA-SWORD (the technique we used to assess pilot situation awareness). The seven experiments are discussed in Sections 6.7 to 6.13. Finally, general conclusions are drawn in Section 6.14.

6.2 Metrics

This study used a combination of four different metrics for the assessment of our claims and the evaluation of SaIRA: a) decision performance, b) eye-movements, c) workload, and d) situation awareness. Additionally, short open interviews were conducted at the end of each simulation, which drew out more robust conclusions.

The metrics in question were specifically selected to collect both subjective and objective data from the experiments. The rationale is that by correlating between subjective (workload and SA) and objective (decision performance and eye-movements) techniques, it is possible to distinguish between subjective impressions and the actual performance of the pilots.

6.2.1 Decision performance

In this study, decision performance is regarded as a 'composite metric', made up of three sub-metrics a) decision time, b) decision accuracy and c) data exploration rate. The *decision time* is the time elapsed from the instant in which ADR information is shown on the cockpit display (demanding a decision) and the instant in which the pilot actually makes an ADR decision by

selecting a configuration option through the SaIRA interface. Decision time is highly correlated with task complexity [Hogarth 1975; Quesada et al. 2005].

Decision accuracy is the percentage of correct decisions made by pilots during each experiment. A correct decision corresponds to applying the right configuration for the current operating conditions.

Decision time and decision accuracy are classic metrics which have been extensively used in the literature. We also considered another metric related to the decision performance, *data exploration rate*. As SaIRA allows pilots to 'browse' the information related to each configuration option using the virtual buttons on the EHSI display, in some experiments we recorded the number of times the pilot clicked a button to switch from one description of a configuration to another or the overall number of configurations explored before making a decision. We argue that an increased number of switches between configurations suggested by the system or exploring an increased number of configurations are symptom of confusion; the right decision is not immediately obvious to the pilot and they spend time wavering from one alternative to another.

6.2.2 Eye movements

In the last decade, the study of human cognitive processes and accidents involving systems under human control has led to the conclusion that human error is rarely random, but can be traced to causes and contributing factors [Hutchins, Morrison and Kelly 1996]. As discussed in Chapters 3 and 4, *decision complexity*, *loss of situation awareness* and *frustration* are three factors which greatly contribute to the decrease in pilot *decision accuracy* and *performance*. The two last factors are important for the evaluation of the effectiveness of the decision support information generated by SaIRA and in the light of the overall hypothesis of this study (Section 3.3).

The literature shows that several features of eye movement (e.g. fixation duration, saccadic amplitude, visual attention distrubution) have been successfully used to draw conclusions concerning pilot workload, loss of situation awareness and frustration, and to examine the usability of cockpit instruments [Morrison et al. 1997; Diez et al. 2001; Merchant and Schnell 2001; Manhartsberger and Zellhofer 2005; Hayashi et al. 2006; Duchowski 2007]. In this light, eye movement analysis has been shown to be effective in supporting the design of new cockpit instruments capable of both *reducing human error* (e.g. Hanson [2004]) and *improving human perception and task performance* (e.g. Morrison et al. [1997]).

In this research specific features of eye movement are used to get an insight into pilot cognitive demand (related to task complexity), frustration and distribution of visual attention. As stated in the central hypothesis, the main objective is developing a framework for decision support that improves both decision accuracy and performance. However, if the framework is *also* able to reduce cognitive demand and frustration during ADR decisions, then the conclusions can go beyond the specific conditions tested in the experiments, making the research more robust and generally applicable. For instance, Section 3.2.1 shows that extreme decision complexity leads to segmentation of information, which makes the overall decision more prone to errors and strongly decreases human performance; if the decision support information generated by SaIRA is found to reduce task complexity, then it can be argued that in general, SaIRA has a positive by-product

effect on both decision accuracy and performance. In other words, *eye movement analysis is used to make the conclusions on decision accuracy and performance more robust*. Workload and situation awareness, the two metrics discussed in the following two sections, are used in the same way in this research.

For the sake of clarity, basic concepts and nomenclature of eye tracking methodologies are introduced first, followed by how eye movement data is used to corroborate the results from other objective and subjective metrics in which pilot cognitive demand and situation awareness are assessed. For a complete and up-to-date discussion of eye tracking methodology, refer to Duchowski [2007]. Further details about the eye-tracking system specifically developed and used in this research are given in Appendix A.

Eye movement is a combination of two main behaviours: **fixations**, where the eye is relatively still (for a period of 150-200 ms) and **saccades**, where the eye moves rapidly between fixations. Due to the technical characteristics of the eye-tracking system used, the concept of fixation used here is defined as a cluster of raw gazes falling in a squared area of 40^2 pixels on the screen. In order to filter out micro-saccades resulting from moving to the periphery of the previous fixations, a lower bound of 100 pixels (roughly 5 degrees of visual angle at 1 meter from the screen) was set on the 46 inches screen used for the simulations. No upper bound was defined because saccades can be quite large (e.g. up to 20 degrees).

A **scanpath** is a series of fixations and saccades. On an image of the aircraft cockpit, they are represented as straight lines drawn between consecutive fixations.

A **backtrack** is a particular type of scanpath in which a previously fixed point is revisited.

An **Area Of Interest** (AOI) is a specific target of the users' visual attention to a part of the interface.

An **AOI transition matrix** contains the frequency of transition for each pair of AOIs.

On-target fixations are the fixations falling in a AOI divided by the total number of fixations.

The **rate of fixations** is given by the time spent fixating divided by the overall observation time.

The **saccades rate** is the number of saccades per second.

The following studies are critical to understanding how eye movement data is interpreted in this research:

- Rayner [1998] argues that *mean fixation duration* is representative of information complexity and task difficulty.

- Nakayama et al. [2002] show that *gazing time* is negatively related to task difficulty and that *saccades rate* decreases when task difficulty or processing demand increases.

- As well as confirming the results of the two studies above, Goldberg and Kotval [1999] add that scanpath duration is directly related to processing demand. They also show how *transition matrices*, *saccadic amplitude* and *number of saccades* can be efficiently used to make a detailed analysis of the user visual distribution, attention and the complexity of the information to process.

Despite the successful history of eye movement analysis applications (see Rayner [1998] for a review), some researchers (e.g. Flemisch and Onken [2000]) express doubts about the effectiveness of conclusions based solely on eye-tracking techniques. For instance, they argue that fixations in a specific area of the cockpit show that the pilot is "looking" there, but the link with the cognitive processing of the information in that area is dubious. However, more recent studies on exogenous attention argue that although it is possible under experimental conditions to dissociate visual attention and eye fixations, under more natural circumstances attention and eye movements are tightly linked and may even rely on the same underlying network in the brain ([Smith and Kosslyn 2007], cf. [Corbetta and Shulman 2002]).

This is one of the reasons why, for this research, a decision was made to merge quantitative data coming from eye movement analysis with qualitative data coming from subjective metrics and post-experiment interviews. This approach enabled us to check the validity of conclusions made on the basis of eye movements against the pilots' subjective inputs. For example, having noticed a high concentration of fixations on a specific chunk of decision support information, the pilot was asked after the test why he/she was looking there, whether he/she found the information too complex to process and so on.

6.2.3 Mental workload

Mental workload (MWL) is the proportion of cognitive resources demanded by a task or set of tasks. The most prominent subjective techniques for MWL assessment are the Cooper-Harper Scale [Cooper and Harper 1969], the Bedford Scale [Roscoe and Ellis 1990; Roscoe 1987], the SWAT (Subjective Assessment Technique) [Reid and Nygren 1988] and the NASA-TLX (Task Load Index) [Hart and Staveland 1988; Hart 2006]. For a more comprehensive list, refer to Stanton [2005].

NASA-TLX and SWAT are the most cited techniques in studies of aviation psychology. However, SWAT has been criticised on several occasions for having low sensitivity [Luximon and Goonetilleke 2001] whilst NASA-TLX has been found to perform better, particularly in situations of low mental workloads [Hart and Staveland 1988; Hill et al. 1992; Nygren 1991]. For these reasons, NASA-TLX is used in this thesis.

Subjective data obtained through NASA-TLX were merged with physiological information coming from the eye movement analysis, in order to improve the robustness of the conclusions.

As well as fixation duration, saccade length and other parameters from the list in Section 6.2.2, endogenous *blink rate* has been found to decrease with increasing MWL [Wierwille and Eggemeier 1993; Salvendy 1997]. Unfortunately, the eye tracking system used is not able to detect blinks, therefore, it was not possible to use this parameter in support of the research.

NASA-TLX can be administered as a paper/pencil version or an electronic version using the software distributed by NASA [NASA 2010]. We found the handheld version of the NASA-TLX software developed by Cao et al. [2009] from Wayne State University particularly useful and fast to administer. When applicable, at the end of the experiment, pilots were required to complete the NASA-TLX assessment on a Compaq iPaq handheld (running WindowsCE).

6.2.4 Situation awareness

Section 3.2.2 introduces and discusses SA in relation to the SCMS dynamic reconfiguration problem. Endsley [1995*b*] provides a review of approaches to SA measurement which includes subjective rating scales (self and observer rating), questionnaires (on trial and post-experiment), freeze techniques (e.g. SAGAT [Endsley 1995*b*]), physiological techniques (e.g. eye-tracking), performance measurement, external task measurement and embedded task measures.

Some SA measurement techniques are tailored for certain specific application domains. To mention two, the Mission Awareness Rating Scale (MARS) [Matthews and Beal 2002] was developed for military scenarios and SASHA was developed by Eurocontrol for air traffic controllers' SA in automated systems [Jeannott et al. 2003]. These techniques are not suitable for this work because they are specifically designed for contexts which are significantly different from SCMS dynamic reconfiguration.

As an alternative, the Situation Awareness Subjective Workload Dominance (SA-SWORD) [Vidulich and Hughes 1991] technique was adopted for this study; the technique was specifically conceived to assess and compare pilot SA when using two or more different cockpit displays or interfaces. SA-SWORD is a variation of the original Subjective Workload Dominance (SWORD) technique [Vidulich et al. 1991] which rates the dominance of one task over another in terms of the workload imposed.

SA-SWORD was first used by Vidulich and Hughes [1991] to compare two F-16 cockpit displays (the FCR and HSF displays) in an aircraft simulator. Given the experimental similarities with the experiments presented here, this technique seemed particularly suitable.

6.2.5 Post-experiment open interviews

Following each simulation, an open interview was conducted with each pilot. The eye-tracking system allows access to basic statistics about the eye movements straight after a simulation, e.g. number of fixations per AOI. This allowed the experimenter to quickly identify possible singularities and further investigate them with the pilot in the interview. For instance, having noticed an anomalous concentration of visual attention on a specific instrument, the experimenter could ask the pilot if the information contained in that area of the cockpit was particularly difficult to process or not easily understood.

When applicable, open interviews were used to corroborate the results obtained through SA-SWORD. Pilots were asked questions that addressed their understanding of the situation, such as "Why did SaIRA suggest switching off the Landing Waypoints Generator system?" or "Why did two computing modules stop working?"

6.3 Apparatus and Materials

The architecture of the system used for this series of experiments is shown in Figure 6.1. The bespoke software runs on two computers: the Simulation Computer (SC) runs the flight simulator, the SaIRA inference engine and the eye-tracking system. Pilots interact with the SC. The Control

Computer (CC) runs the SaIRA Control software which allows the experimenter to control the experiment execution flow in real-time, e.g. issuing faults, triggering avionics reconfigurations, activating/deactivating aircraft functions. Furthermore, the software running on the CC allows off-line post-processing of the results.

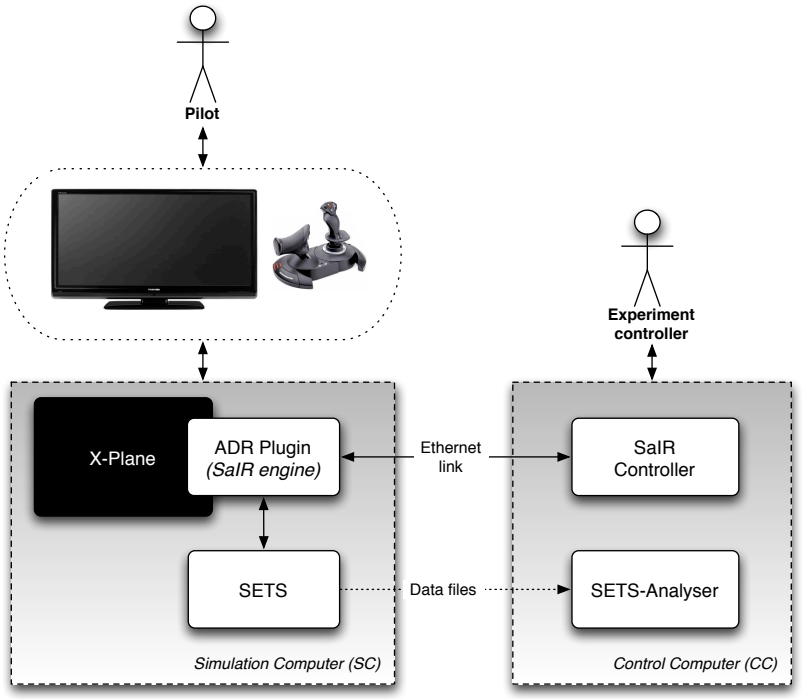


Figure 6.1: Simulation system architecture.

The controlling system merges information about various events: a) simulator events (e.g. alarms firing, faults), b) pilot actions (e.g. button pressing, decision support information switch, configuration description browsing) and c) eye-tracking events (e.g. loss of eyes target). All the events are logged with a time-stamp and integrated with eye movements coordinates in real-time.

During the simulation, events unfold and are logged in Simulation Time Units (STU). One STU corresponds to 1/60 second. The STU is set according to the eye-tracking system unit, the eye-tracking frequency being 60Hz.

6.3.1 Flight simulation software

A brief description of each component of the simulation system architecture depicted in Figure 6.1 is provided hereinafter.

6.3.1.1 X-Plane flight simulator

The FAA[1]-approved X-Plane flight simulator [Laminar Research Inc. 1993] was used in this thesis. X-Plane was executed on the SC (AMD Athlon 64-bit, 2.4Ghz, 2Gb RAM) running Ubuntu Linux 9.10.

[1]Federal Aviation Administration, http://www.faa.gov

X-Plane is not open-source but it is equipped with a rich plugins interface that provides a high degree of customisation. All the experiments were performed using the Boeing 737-900ER aircraft model provided by the x737 package [European Aircraft Developer Team 2010].

The flight simulation was projected on a 46 inch Toshiba Regza LCD screen; pilots were positioned approximately 50 inches away from the screen. They used the Thrustmaster T-Flight Hotas X joystick [Thrustmaster 2010] to control the aircraft and a classic mouse to click on the virtual buttons on the ADR interface to select and apply an avionics configuration when requested.

6.3.1.2 ADR plugin

As part of this PhD programme, the ADR Plugin component for the Boeing 737-900ER model of X-Plane was developed; the software reorganises the avionics so as to resemble the IMA concept and provides avionics dynamic reconfiguration capabilities. The ADR Plugin simulates faults and implements the SaIRA concept, including SFE, RE and DSE (Figure 5.7). The user interface of the ADR plugin was described in detail in Section 5.7.

When a reconfiguration is issued, SaIRA temporarily replaces the contents of the Electronic Horizontal Situation Indicator (EHSI) display with ADR decision support information (pilots are still able to access basic aircraft control information from the analogue instruments). Two cases are possible: the pilot has to choose between applying a suggested configuration or switching to safe mode (Chapter 5, Figure 5.11(a)), or he/she has to choose between two or more configuration options (Chapter 5, Figure 5.11(b)). Pilots are provided with the former or the latter type of information depending on the needs of each experiment. Additionally, schematics about the fault detected by the sensors temporarily replace the content of the Electronic Attitude Director Indicator (EADI) display (Chapter 5, Figure 5.12). The details of the design of the information shown by SaIRA on the cockpit displays have been already provided in Section 5.7.1.

SaIRA generates the following three cockpit conditions:

- **INFO_1**: Baseline condition. Only 'Fault information' and 'Diagnosis' data is displayed (upper portion of data in Figures 5.11(a) and 5.11(b)); no 'Reliability' figures are shown. The original content of the EADI display is not modified.

- **INFO_2**: Controlled condition. EHSI contains the same information as INFO_1 but the EADI shows schematics about the fault detected by SaIRA (Figure 5.12).

- **INFO_3**: Controlled condition. Full SaIRA decision support information is displayed, including explanation, implications and reliability figures, as shown in Figures 5.11 and 5.12.

It is worth sending out a note concerning the metrics to evaluate the configurations. In the description of the `wsm decision-repair` algorithm given in Appendix B (Section B.1.5.2), the pilot is given the opportunity to dynamically change the weights of the metrics used to evaluate the configurations (e.g. give more importance to 'safety' related constraints at the expense of 'performance'). Because of the complexity of the experiments being described in this chapter, and the time-scale of a Ph.D. research programme, it was not possible to implement and empirically investigate this functionality. During the experiments, the pilots are provided with up to three

reconfiguration alternatives to choose from; the implemented SaIRA interface does not allow the dynamic modification of the metrics used by the system.

6.3.1.3 ADR controller

The ADR Controller was designed for remote control of SaIRA during the experiments. The ADR Controller is written in C++ and designed to run on both Linux 2.6 and Mac OS X; it communicates with the ADR Plugin through a TCP/IP link.

The experimenter uses this tool to start and stop the simulations and to fire faults when required.

During the experiments, the ADR Controller was executed on a Apple MacBook (Intel Core 2 Duo 2.16 GHz, 2Gb RAM) running Mac OS X 10.5.

6.3.2 Eye-tracking system

The SaIRA Eye-Tracking System (SETS) uses OpenGazer [Zielinski 2009] as the eye-tracking engine and extends its functionality to meet the needs of this series of experiments. Some additional functions (e.g. interface with the X-Plane flight simulator, output data pre- and post-processing, fixations/saccades/scanpaths processing, etc) were developed specifically for this study with the support of the OpenCV machine vision library [Intel 1999].

SETS gives higher priority to *precision* than *accuracy*. High accuracy, commercial eye-tracking systems are available on the market (e.g. TOBII [Tobii Technology 2010], Smart-Eye [Smart Eye AB 2010], Eyegaze [LC Technologies 2010]).

SETS is highly sensitive to head movements. During a preliminary series of tests we experienced a heavy loss of eye-tracking data. As a consequence, during formal tests, pilots were asked to lean their head on a stable support. This configuration change gave good results.

Since OpenGazer needed to be tuned for each pilot, before starting the first experiment, SETS went through a short calibration process which lasted approximately 2 minutes. The system showed a grey square in nine different locations on the virtual cockpit and pilots were asked to fixate on the object. This happened only once for each pilot.

6.3.2.1 Definition of the Areas of Interest

SETS superimposes a map of 7 AOIs on the B737 cockpit as shown in Figure 6.2. The AOIs on the EADI and EHSI displays are juxtaposed to isolate each specific ‘chunk’ of decision support information. Figures 6.3 and 6.4 provide a zoom on the EHSI and EADI displays respectively.

The definition of these AOIs allowed us to characterise precisely the visual attention of pilots during ADR decisions. The separation of generic fault information from explanations and implications was decisive in investigating the claims appropriately.

6.3.3 Eye-movement data analysis software

The SETS-Analyser is software designed specifically for this thesis, in order to make off-line, post-processing analysis of the experimental data collected by SETS in real-time (e.g. backtrack

Figure 6.2: Definition of the AOIs on the cockpit of the Boeing 737-900ER.

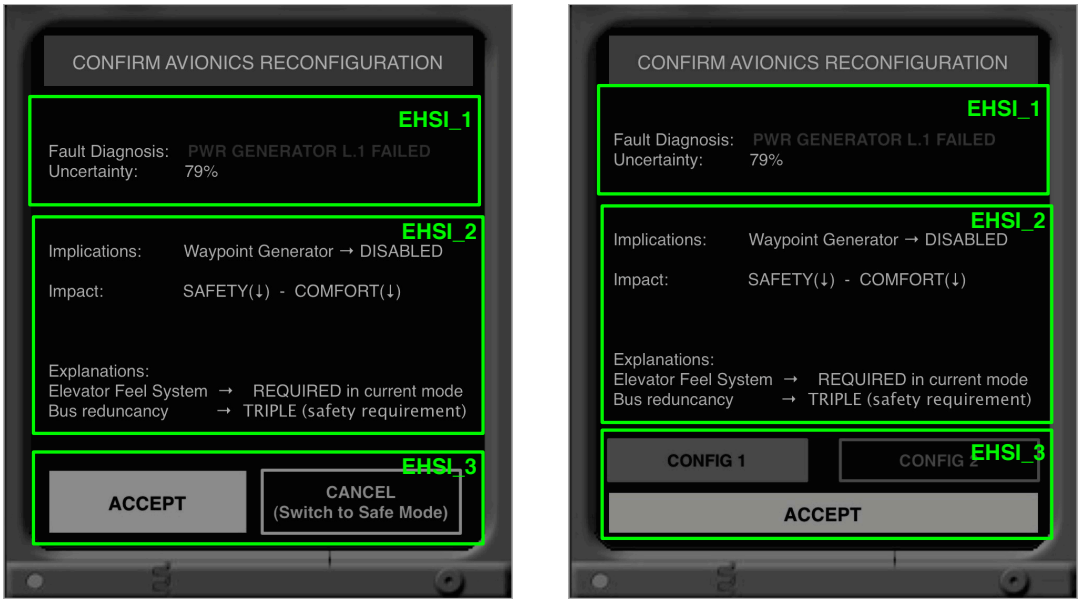
calculation, saccadic amplitude estimation etc). A subset of the data automatically generated by the SETS-Analyser was used to obtain the research conclusions (i.e. fixation duration, saccadic amplitude, number of on-target fixations); however, other parameters like hotspot maps were used in a qualitative manner in order to provide further support for the inferences made.

Using raw gaze coordinates and additional meta-information (such as event tags) recorded by SETS at simulation time, the software generates the following data: number of fixations, number of saccades, raw gazes in each AOI, gazes transition matrix, fixations transition matrix, sum of fixations time for each AOI, mean fixation duration, mean fixation duration for each AOI, rate of fixation, on-target fixations, mean saccades length, mean raw gazes per AOI, saccades rate, number of backtracks, number of backtracks per AOI, fixations/saccades rate, scanpath maps (JPEG image), backtrack maps (JPEG image) and hotspot maps (JPEG image).

A typical hotspot map is shown in Figure 6.5. This kind of representation provides a qualitative but clear picture of the overall visual attention of the pilots during ADR.

Numerical data is stored in 'Comma Separated Value' format files that are compatible with Microsoft Excel, OpenOffice.org and IBM SPSS. Depending on the needs, a combination of these three tools was used to perform statistical analysis on the data generated by SETS-Analyser.

SETS-Analyser is written in C++ and it is designed to run on Linux 2.6.



(a) One ADR alternative plus Safe Mode.

(b) Two ADR alternatives.

Figure 6.3: Definition of AOIs on the EHSI display when the pilot get to choose (a) between one alternative and Safe Mode, or (b) between two alternatives.

6.4 Pilots' Training

None of the pilots involved in this study had prior experience with the avionics reconfiguration technology developed in this research. Attention was given to providing all pilots with the same degree of training and information about the system and the experiments.

All participants were provided with: a) written instructions to study before the tests; b) four hours of training, delivered in two evening sessions, roughly one week before the start of the experiments; c) half an hour of same-day training.

The *written instructions* were distributed in the form of an A4 pamphlet containing the following information:

- general description and objectives of the experiments;
- general information about SaIRA, X-Plane and the ADR process;
- information about NASA-TLX, including a paper version of the form and instructions on how to use the handheld version of the software;
- information about SA-SWORD, including a paper sample of the form.

All the written information delivered to the pilots was discussed during the two evening training sessions. Furthermore, in the second training session, all pilots were given the chance to attend a preliminary, assisted flight simulation with the aim of getting acquainted with the system. On this occasion, no SaIRA support was provided in order to preserve the integrity of the experiments aimed at assessing pilot *a priori* trust in the system.

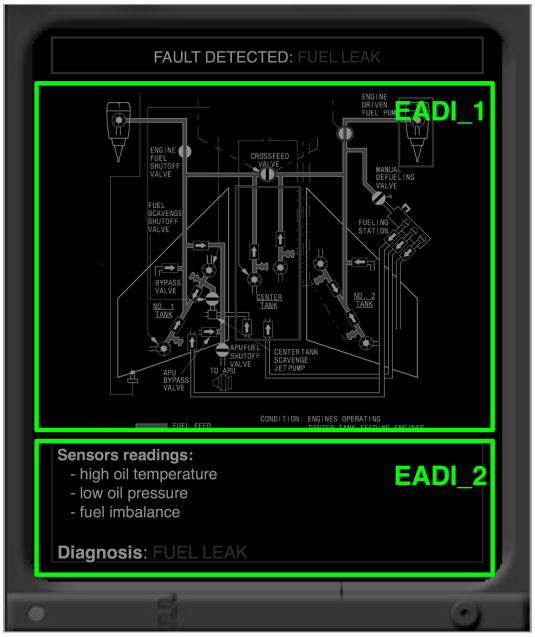


Figure 6.4: Definition of the AOIs on the EADI display.

Finally, pilots were given a half hour recap before starting the experiment, in which they had the opportunity to put forward any final questions.

Attention was paid to provide the pilots with a "good-enough" presentation of the system without selling it as an infallible device. It was important to establish pilots' trust in SaIRA without incurring the risk of overly biasing the experimental results. The complexity of the problem being tackled by SaIRA was clarified to pilots; they were explicitly informed that their role was critical for the success of the reconfiguration process and that SaIRA was designed to *support* them, not perform the reconfiguration for them. It was clearly stated that *the pilot was always in control and responsible for the operation of the aircraft*. The fact that pilots were not overly biased by the pre-experiments presentation of the system is confirmed by the statistically relevant results obtained in the experiments about complacency and trust discussed later in this chapter.

One last note about training: SA-SWORD does not provide a direct measure of SA but an assessment of the conditions in which SA is highest. It follows that a very clear understanding of SA is required for the technique to work. Particular care was taken during all training phases to verify that all pilots had mastered the concept of SA and how it was going to be assessed during the study.

6.5 Design

The experiments were designed following the *gold standard statistical argument* [Cairns and Cox 2008]; in particular, both independent and dependent variables were identified *a priori* and the expectations for the outcome of each experiment were clearly stated in advance.

Table 6.1 gives an overview of the design of the experiments. For the clarity of exposition,

Figure 6.5: Definition of the AOIs on the EADI display.

more information about the design of each experiment, including the levels of the variables and the expectations, are provided in the following sections, when each experiment is described in detail.

6.6 Method

Each experiment had its own design characteristics, however, they all shared a general method which is briefly described here. Experiment-specific differences are reported in the following sections.

Each experiment consisted of a set of *tests*, each of which was usually targeted on a specific claim. Each test comprised one or more *flight simulations* (also referred to as *simulations*, *flight scenarios*, or simply *scenarios* in the remainder of this chapter). The subdivision of each experiment into a number of tests pertains only to the organisation and description of the experiments; the pilots were not aware of this.

During each flight simulation, pilots faced an unexpected event (e.g. a fault) which required them to reconfigure the avionics in real-time. When prompted with a reconfiguration, pilots had to make a decision between the available alternatives by clicking with the mouse on the virtual buttons that appeared on the EHSI display (see Figures 5.11(a) and 5.11(b)).

Pilot behaviour was analysed under the effect of a series of condition variables, including, type of decision support information, maximum time budget, correctness of decision suggestions and

EXPERIMENTS DESIGN			
Experiment Identifier	**Independent Variable(s)**	**Dependent Variable(s)**	**Design Type**
Experiment A	Information Correctness, Explanations	Trust self-assessment	Within-subjects
Experiment B	Information Correctness	Decision accuracy	Within-subjects
Experiment C	Information Type	Decision accuracy, decision time, nr of clicks, fixation duration, workload and frustration self-assessment, situation awareness self-assessment	Within-subjects
Experiment D	Time pressure, Implications	Decision accuracy, fixation duration, decision time, nr of alternatives explored, visual attention distribution	Within-subjects
Experiment E	Information Correctness (within-subjects), Information Reliability (between-subjects)	Workload self-assessment, fixation duration, decision time	Mixed factorial
Experiment F	Perception of Risk	Pilot accept/refuse to reconfigure	Within-subjects
Experiment G	Reliability Framing	Pilot accept/refuse to reconfigure, fixation duration, comfort self-assessment	Within-subjects

Table 6.1: Design of the experiments.

information framing. Objective and subjective data was collected during and after each experiment.

The pilots performed each experiment individually. Each flight simulation lasted between 3 and 6 minutes. There was a total of 25 flight simulations for the whole study (including all seven experiments), adding up to an overall average testing time of two hours (excluding configuration and scenario switching time). The seven experiments were split into two sessions which took place on two different days.

The risk of carry-over effect was minimised by differentiating the characteristics of each reconfiguration (e.g. different types of fault) whilst keeping similar operating conditions in the set of simulations, according to the requirements of each experiment (e.g. weather conditions, time of day).

The eye-tracking system was calibrated before each pilot started using the system on each day.

The simulation was stopped and the system was reset between consecutive flight simulations (the SETS calibration was not lost during this operation). The X-Plane flight simulator allows the

configuration of *scenarios*; scenarios allow the simulation to be started at any phase of flight, e.g. it was possible to run a set of tests starting in mid-air, without the need for take-off each time. All the scenarios used in an experiment were configured prior to the start of the experiment.

It is helpful to give a few notes before presenting the detailed discussion of each experiment. Either because it was not possible to verify the parametric behaviour of the dependent variables under consideration, or because there was simply not enough relevant knowledge (SaIRA is an entirely new system), non-parametric tests were used to analyse the results of the experiments in the majority of cases. Data generated through the NASA-TLX and the SA-SWORD techniques were an exception; the parametric nature of this data was assumed due to the huge amount of literature accrued in the last couple of decades that process this type of data using the ANOVA method.

Finally, all pilots judged the simulations to be realistic in the light of their purpose. Furthermore, having been questioned about the workload generated by the experiments, they commented that it was similar to the circumstances they usually face in this kind of real-time fault management problem.

6.7 Experiment A – Information Correctness and Explanations

Description and aim

Experiment A aimed at investigating the effect of information correctness and explanations availability on pilots' *trust* in a hypothetical DSS for ADR. This experiment encompasses Claim 9 which is repeated here for the sake of clarity:

> **Claim 9 - Trust**: Pilot trust in a DSS for ADR would decrease when the system makes (apparently) incorrect inferences. This phenomenon would be mitigated by providing pilots with an explanation of the inference processes.

There are situations in which the solution to a problem is not straightforward and the correct course of action might either not be clearly visible to the pilot or it might seem inapplicable or seriously hazardous. A correct suggestion coming from the system might seem wrong for all sorts of reasons (e.g. loss of SA, framing effect) and be erroneously avoided. For example, consider the following case:

> *The pilot of a Boeing 737-900ER is preparing for landing. A series of inter-dependent faults affects both the Fuel Monitoring Unit (FMU) and some computing modules. The system suggests a configuration in which the way-points generator is not active. This recommendation could seem wrong, at least without knowing that a fault to the FMU leads to the automatic disengagement of the VNAV*[2] *and, when VNAV is unavailable, way-points are not available either. An explanation of the reasoning of*

[2] Vertical NAVigation – a function of the autopilot which directs vertical movement of the aircraft either according to pre-programmed FMS flight plan (during cruise) or according to the Instrument Landing System (ILS) glideslope (during approach).

the system could make the recommendation more transparent and enable the pilot to make the right decision more quickly (this example is one of the cases tested in this experiment).

In this experiment a reconfiguration recommendation of the type just described is referred to as 'apparently wrong'. Here the objective is gathering empirical evidence to show that whilst 'apparently wrong' suggestions provoke pilots' mistrust in the system, when explanations of those suggestions are provided (which show their soundness), pilots' trust in the system is restored and, statistically, they are more prone to accept the suggestion generated by the computer.

Procedure

X-Plane was configured to run 3 tests characterised by similar flight conditions. Before starting the experiment, the characteristics of SaIRA were briefly reviewed; the experimenter took particular care to present the system as highly reliable, in an effort to increase the pilot's trust in it. The pilot was then asked to complete any potential real-time fault management procedures correctly and in the shortest time possible, with the support of SaIRA. The three tests had the following characteristics:

- **Test 1:** the system showed *evidently right* information of type INFO_3 *without explanations* (baseline case);
- **Test 2:** the system showed *apparently wrong* information of type INFO_3 *without explanations*;
- **Test 3:** the system showed *apparently wrong* information of type INFO_3 *with explanations*.

In all cases, SaIRA was configured to provide only one ADR configuration; the pilot could either apply it or switch to safe mode.

After Test 2 a short interview was conducted with the participant, in order to understand whether he/she had noticed anything odd about the decision support information provided or not and to inform him/her that the following simulations would be supported by a wider range of decision support information, including explanations.

At four pre-defined times (between the simulations, not during them), the pilot was asked to give feedback in terms of how much he/she trusted the suggestions produced by the system on a scale from 0 to 10, using the form displayed in Figure 6.6. The pre-defined times were as follows:

- **Time 1:** following the training session, after the pilot had received a complete description of the functioning of the system, but before starting any simulations assisted by SaIRA. This question is about *a priori* trust in the DSS;
- **Time 2:** after Test 1;
- **Time 3:** straight after Test 2, before the interview;
- **Time 4:** after Test 3.

This was a within-subjects experiment.

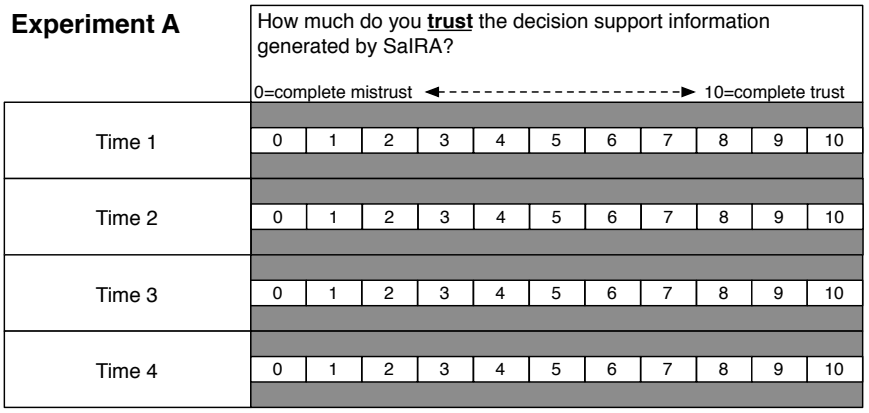


Figure 6.6: Questionnaire format for the assessment of pilots' subjective ranking of their trust in the decision support information generated by SaIRA.

Expectations

The expectations were as follows:

- E1: pilot trust in the system would decrease at the first occurrence of a wrong ADR suggestions (Time 1);
- E2: by showing *why* the system arrived at a 'dubious' conclusion, an explanation would provide an insight into the system inference process, would reduce the feeling of uncontrollability over the situation and, as a result, would mitigate the effect hypothesised by E1, partially re-establishing pilot trust in the system (Time 4);
- E3: in this experiment pilots were provided with only one configuration recommendation, they could either apply it or refuse to apply it, instead of switching to safe mode. Explanations would increase pilot acceptance of the recommendations of the system.

Results

Table 6.2 and Figure 6.7 report the descriptive statistics concerning the pilots' subjective ranking of their trust in the decision support information generated by SaIRA at different stages in the experiment series.

As expected, trust strongly decreases after the wrong information is shown (E1) and is partially recovered when explanations are provided (E2). During the short interview after Test 2, all pilots reported that, in their opinion, the system had generated unsuitable decision suggestions.

The statistical significance of the results is proved by the Friedman's test ($\chi^2(3) = 30.84$, $p < 0.001$, N=13).

Concerning E3, the results about the number of pilots who decided to apply the configuration suggested by SaIRA (nrAccept) follows the same trend of trust, as shown in Figure 6.8 and Table 6.3. Twelve pilots applied the configuration in Test 1, none of the pilots did in Test 2 and, as a result of increased quality of explanations, eight pilots accepted the reconfiguration in Test

Pilot trust in the system				
	Minimum	**Maximum**	**Mean**	**Std. Dev.**
Time 1 (trust *a priori*)	3	10	6.54	2.14
Time 2 (trust after correct info.)	5	10	8.31	1.38
Time 3 (trust after apparently wrong info, no expl.)	0	4	2.23	1.36
Time 4 (trust after apparently wrong info and expl.)	5	8	6.38	1.19

Table 6.2: Pilots' subjective ranking of their trust in the decision support information generated by SaIRA at different stages of the series of experiments.

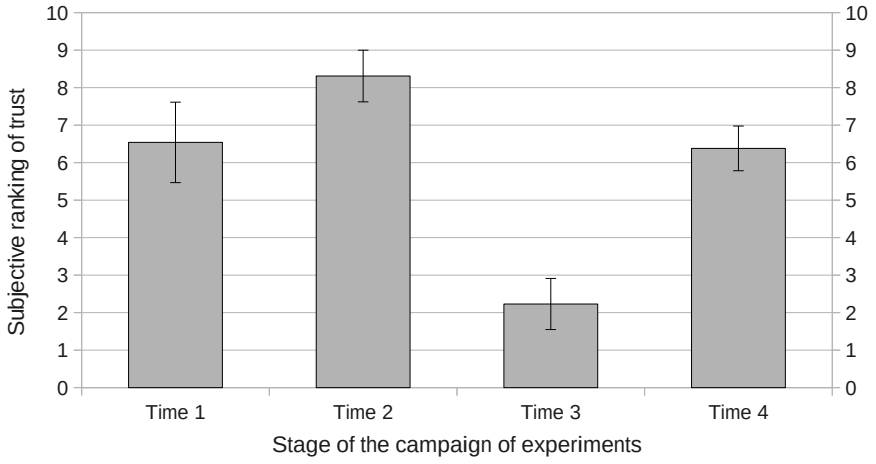


Figure 6.7: Subjective ranking of pilots' trust in the decision support information generated by the DSS at different stages of the series of experiments.

3. The main effect on 'nrAccept' is statistically significant (Cochran's Q test: $\chi^2(2) = 18.667$, $p < 0.001$, N=13).

	Accepted	**Refused**
Test 1 (Time 2)	12	1
Test 2 (Time 3)	0	13
Test 3 (Time 4)	8	5

Table 6.3: Number of pilots who accepted or refused the reconfiguration alternative suggested by SaIRA.

Interestingly, a strong correlation between pilots' trust in the system at Time 1 and their age was noted (Spearman's test: $\rho = -0.739$, $p < 0.004$, N=13). In other words, older pilots were more reluctant to trust *a priori* the information generated by the system than their younger colleagues. More information concerning pilot age is shown in Table 6.4.

	Minimum	**Maximum**	**Mean**	**Std. deviation**
Pilots' age	31	68	43.31	11.43

Table 6.4: Descriptive statistics for pilots' age.

The correlation between pilot age and trust was not found for the rankings recorded at Time 2

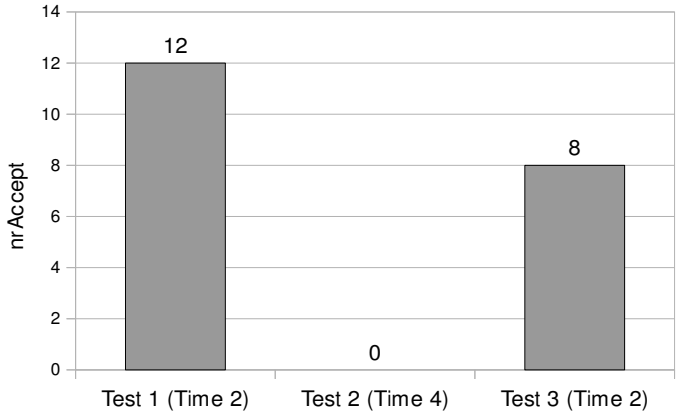


Figure 6.8: Number of pilots who accepted the reconfiguration alternative suggested by SaIRA.

(Spearman's test: $\rho = -0.546$, n.s.), Time 3 (Spearman's test: $\rho = 0.233$, n.s., N=13) and Time 4 (Spearman's test: $\rho = -0.455$, n.s., N=13); therefore, there is a correlation only with the *a priori* trust.

Discussion

Notwithstanding how careful designers are, expert systems make mistakes. This technological limit must be taken into consideration when developing a DSS for safety-critical applications.

The preparation of the participants in this experiment aimed at provoking a sense of *a priori* trust, through a purpose-made, misleading presentation of the system as an infallible support to safety-critical decisions. Jian et al. [2000] show that pilots usually mistrust the automation when they do not know whether it is accurate or not. However, at least in this experiment, the high ranking of trust at Time 1 (before using the system) shows that pilots were successfully persuaded by the way the system was presented to them.

It is true that there is no way to tell if this result was caused by how the system was presented to the pilots or if they were biased by the overall experience of testing novel, airborne technology. Probably, in a real scenario their ranking of trust would be lower on average, because the perception of risk would be stronger. However this is not important for this study, as here the aim was making sure pilots had a reasonable trust in the system before starting the experiments.

We turn now to the impact that explanations of the automated inference process have on human trust in the system. The results show that by explaining how and why the system reaches certain conclusions, human trust in a DSS is enhanced and the system-human cooperation improved. Both the ranking of trust and 'nrAccept' decrease from Time 2 to Time 3 (provision of 'apparently wrong' suggestions) and increase from Time 3 to Time 4 (provision of explanations).

SaIRA proved effective in bringing to the attention of the pilots ADR alternatives that they would not have considered without its support but that in a real scenario could have avoided catastrophic consequences.

It is worthwhile commenting on the correlation between *a priori* trust and pilots' age. In this respect, it is interesting to note the following two facts in combination, a) younger pilots seemed

to be more 'naively' trusting towards SaIRA; b) post-experiment interviews revealed that neither of the two younger pilots amongst the participants were aware of the history of protests and strikes against the introduction of glass-cockpit automation that characterised the aviation history of the 1980s (see Amalberti [1999] for a review of the main facts) nor were they aware of the issues and controversies that characterise this research domain and its key actors. Most probably, a better knowledge of these facts would change their reactions.

In conclusion, the results from Experiment A provide support to the claims concerning the importance of explanations and mental simulation during ADR decisions put forward in Chapter 4; furthermore, they provide evidence of the overall effectiveness of SaIRA and, specifically, of the importance of including explanations of the automated line of reasoning of a DSS for safety-critical applications in the decision support information set.

6.8 Experiment B – Effect of Information Correctness on Complacency

Description and aim

The objective of Experiment B was to assess the potential emergence of *complacent behaviour* by the pilots about the ADR recommendations generated by the automation, especially in the light of the results about trust obtained from Experiment A. The potential emergence of a complacent behaviour is assessed by manipulating the correctness of the information provided by the system.

This experiment aimed at collecting evidence for the assessment of Claim 7, which is reported here for the sake of clarity:

> **Claim 7 - AIC**: Pilots would be subject to Automation-Induced Complacency during highly autonomous ADR. This phenomenon would be mitigated by decreasing the level of autonomy of the process.

Procedure

X-Plane was configured to run 4 similar scenarios. Before running the tests, the pilot was told that in rare cases the system could generate wrong suggestions, due to technological limitations. If that happened, he/she should avoid the suggestion and switch to safe mode.

The pilot was asked to complete any potential real-time fault management procedures correctly and in the shortest time possible. The experiment was structured into two distinct tests; three simulations were run in Test 1 and one simulation was run in Test 2. SaIRA was configured to generate and display INFO_1 for all simulations.

This was a within-subject test, each pilot ran all the simulations listed below.

Test 1

Each pilot was asked to perform the first three scenarios.

During each scenario, a fault was simulated between 2 and 5 minutes after the start of the simulation.

SaIRA generated *correct* decision support information, suggesting an optimal configuration that could effectively mitigate the effect of the fault in question. The right decision in these first 3 scenarios was accepting the suggested reconfiguration (not switching to safe mode).

Test 2

Each pilot was asked to perform the fourth scenario.

A fault was simulated between 2 and 5 minutes after the start of the simulation.

SaIRA generated *wrong* decision support information, suggesting a configuration that was either not applicable or would evidently put the safety of the crew at risk, e.g. switching off the external positioning systems before landing.

Expectations

As a result of the emergence of AIC, pilots were expected to accept the ADR advice in Test 2 even if it was wrong. In other words, the *decision accuracy* would be lower in Test 2 than in Test 1.

Results

The primary result is that only 3 out of 13 pilots refused the wrong advice in Test 2. The results for both tests are reported in Table 6.5; additionally, the proportion of correct decisions is given as a percentage and interpreted as *decision accuracy (DA)*. The decrease of DA from Test 1 to Test 2 is statistically significant (Chi Square test: $\chi^2(1) = 13$, $p < 0.001$, N=13).

Test	Right	Wrong	Decision accuracy
Test 1 (3 simulations, 39 cases)	36	3	92.3% (s.d. 4.3%)
Test 2 (1 simulation, 13 cases)	3	10	23.1% (s.d. 12.2%)

Table 6.5: Right decisions, wrong decisions and decision accuracy (percentage of right decisions) for Test 1 and Test 2.

It is worth mentioning other ancillary results in support of the claim concerning complacency regarding *decision time (DT)* and *fixation duration (FD)* (see Figures 6.9 and 6.10):

- there was no statistical effect for the *correctness of the suggestion provided* to pilots (i.e. Test 1 vs Test 2) on both DT (Wilcoxon Signed-Rank test: $Z = -0.454$, n.s., N=39) and FD (Wilcoxon Signed-Rank test: $Z = -0.105$, n.s., N=39);

- within Test 2, there was a statistical effect for *"spotting the incorrectness of the ADR suggestion provided"* on both DT (Mann-Whitney U test: $Z = -2.846$, $p < 0.04$, N=39) and FD (Mann-Whitney U test: $Z = -2.793$, $p < 0.05$, N=39); see Table 6.6. In other words, those pilots who made the right decision (i.e. those who spotted the wrong suggestion and switched to safe mode) had higher DT and FD than those who made the wrong decision (i.e.

pilots who accepted and applied the wrong configuration). There is a correlation between DT and FD (Spearman's test: $\rho = 0.824$, $p < 0.001$, N=13).

	DT	FD
Overall (52 cases)	46.12 (s.d. 16.82)	370.67 (s.d. 27.54)
Wrong decision (13 cases)	41.19 (s.d. 3.45)	361.174 (s.d. 6.2)
Right decision (39 cases)	90.1 (s.d. 8.36)	422.38 (s.d. 2.51)

Table 6.6: Descriptive statistics for DT (in seconds) and FD (in milliseconds) related to: a) all cases, b) cases in which the pilots accepted the wrong ADR advice (wrong decision); b) cases in which the pilots refused it (right decision).

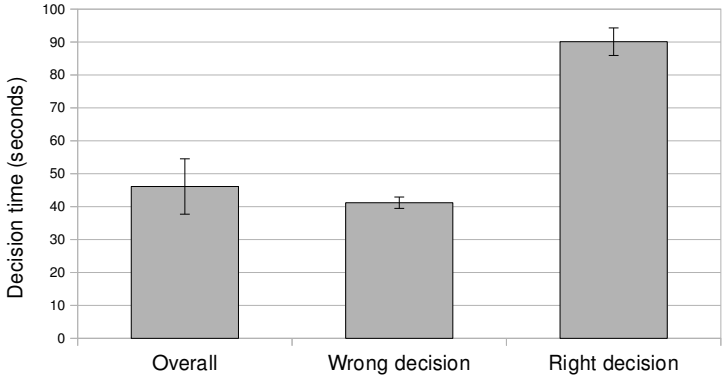


Figure 6.9: Decision time (in seconds).

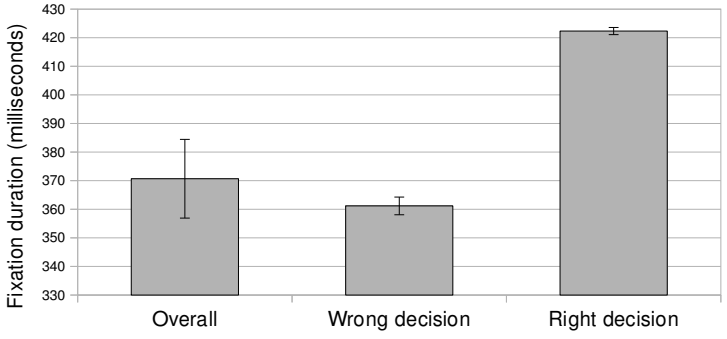


Figure 6.10: Fixation duration (in milliseconds).

Discussion

The main result is the drastic drop (−69%) of DA from the case in which the decision support information is right and the case in which it is wrong. This result strongly supports claims concerning the emergence of AIC during ADR decisions. (It is obviously assumed that pilots do not want to make wrong decisions intentionally).

The fact that there was no statistical physiological variation (such as eye movements) in the decision behaviour as a result of the *correctness of the suggestions provided* can be explained in

two ways (a) pilots didn't notice the wrong advice (this eventuality would support the claim of emergence of AIC) or (b) they did notice it, but this didn't provoke any observable physiological reaction.

Theoretically, the latter option is possible with highly experienced decision makers like pilots; however, it must be noted that a statistical variation of DT and FD was actually found for those pilots who succeeded in spotting the wrong information and modified their decision behaviour accordingly. This argument invalidates option (b) for the ADR decision scenario. Hence, the statistical lack of physiological reaction to subtle, wrong information supports the claim of emergence of AIC.

Wrong information led to lower DT and FD, probably because it made the decision problem simpler. As soon as pilots spotted a 'bad apple' amongst the available reconfiguration alternatives, they avoided it and went for the 'safe' backup option. This speculation is supported by the post-experiment interviews, as seen in the following excerpts:

- "I just dropped it. I had no time to think and the machine was clearly wrong"
- "The backup mode was available" [...] "I was asked to make a quick decision and switching off the FMC was just stupid" [...] "I couldn't figure out why so I cut it out and kept going".

The evidence collected in this experiment verifies Claim 7: pilots are complacent about the ADR advice provided by SaIRA.

6.9 Experiment C – Information Type

Description and aim

Experiment C investigated the effect of *different types of decision support information* on pilot *decision making experience*. More specifically, different sets of information, namely INFO_1, INFO_2 and INFO_3, were empirically compared and their effect was investigated in terms of decision accuracy, decision performance, frustration, workload and situation awareness. This analysis encompasses Claims 5, 8 and 11:

> **Claim 5 - Stress and frustration**: Heightened states of stress during ADR decisions can possibly lead to frustration. The negative effect of frustration would be mitigated by providing pilots with *effective* decision support information.
>
> **Claim 8 - Information type**: Decision support information for SCMS reconfiguration should be "unpacked"/framed so as to make the following cues readily understood by the operators for each reconfiguration option:
>
> - *Explanations*: *why* the system is making a specific suggestion;
> - *Implications*: (*what-if* type of information) what the consequences of applying a specific configuration are;
> - *Reliability*: *how reliable* the advice generated by the system is.

Claim 11 - Graphics: A graphical representation of the fault that triggers an ADR would improve decision performance and accuracy.

Unfortunately, because of a bug in SETS, the eye movement data for one of the pilots was lost. The bug was removed but it was not possible to repeat the experiment. The eye movement analysis performed in this experiment therefore uses the data of 12 pilots instead of 13. All the other data (decision time, decision accuracy etc) is available for all 13 pilots.

Procedure

X-Plane was configured to run 6 similar scenarios. Similar flying conditions were set for all the cases. The pilot was asked to complete any potential real-time fault management procedures correctly and in the shortest time possible.

Between 30 and 120 seconds after starting the scenario, a fault was simulated and a reconfiguration was automatically issued.

Two reconfiguration options were provided, one of which was wrong. The two reconfiguration options were always such that they required a choice between switching off one of two critical functions (e.g. 'Elevator Feel System' or 'Landing Points Generator').

The experiment was structured into 3 distinct tests; 2 simulations were run in each test. As it was a within-subject test, each pilots ran all the simulations listed below:

- **Test 1:** pilots performed the first 2 simulations. SaIRA provided the pilot with INFO_1;
- **Test 2:** pilots performed the following 2 simulations. SaIRA provided the pilot with INFO_2;
- **Test 3:** pilots performed the last 2 simulations. SaIRA provided the pilot with INFO_3 (always showing 'FULL reliability').

It is reminded that SaIRA is designed to generate three types of decision support information characterised by different amount and type of information; they are referred as INFO_1, INFO_2 and INFO_3 throughout the thesis. The definition of these three types of decision support information was given in Section 6.3.1.2.

Immediately after the final test, both the NASA-TLX and the SA-SWORD questionnaires were submitted to the pilot. With reference to SA-SWORD, the pilots were asked to compare their level of SA when performing an ADR with INFO_1 (baseline), INFO_2 and INFO_3 using the questionnaire in Figure 6.11, which was purpose-made for this experiment.

Expectations

INFO_1 is the baseline condition. As a result of better decision support, the following expectations were set:

- E1: *decision accuracy (DA)* would progressively improve with INFO_2 and INFO_3;

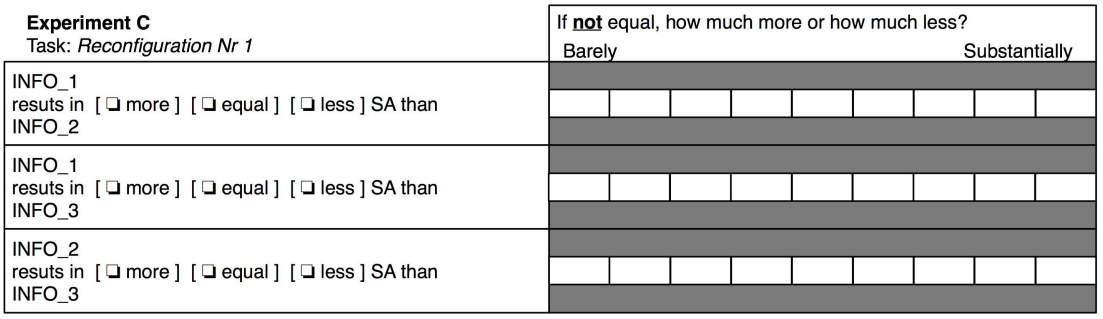


Figure 6.11: SA-SWORD post-simulation questionnaire format for Experiment C.

- E2: it was not possible to make any precise forecast concerning the *decision time (DT)* when the experiment was designed. On one hand, better decision support should reduce the time required by pilots to complete the procedure, as found in FAMSS [Hayashi et al. 2006]; on the other hand, having more information to process could increase the DT;
- E3: the *number of clicks on the reconfiguration buttons (nrCL)* would progressively decrease with INFO_2 and INFO_3. We speculate that the number of times pilots switch from one configuration to another to explore its characteristics would be indicative of their confusion. Better decision support would decrease pilot confusion, therefore this value should also decrease;
- E4: *fixation duration (FD)* would progressively decrease with INFO_2 and INFO_3;
- E5: *workload (WL)* would progressively decrease with INFO_2 and INFO_3;
- E6: *frustration (FR)* would progressively decrease with INFO_2 and INFO_3;
- E7: *situation awareness (SA)* would progressively improve with INFO_2 and INFO_3.

Altogether, expectations from E1 to E7 combine with the general expectation of obtaining improved decision performance with INFO_2 and even more so with INFO_3.

Results

E1: decision accuracy (DA)

Cochran's Q test reveals a statistically significant difference in terms of DA amongst INFO_1, INFO_2 and INFO_3 ($\chi^2(2) = 7.091$, $p < 0.029$, N=26). A pairwise comparison using continuity-corrected McNemar's tests shows that the main improvement over INFO_1 (baseline) is provided by INFO_3. This result is evident from Table 6.7, which contains the descriptive statistics.

E2: decision time (DT)

A significant effect of the type of decision support information on DT is revealed by Friedman's test ($\chi^2(2) = 13$, $p < 0.02$, N=26). A post-hoc test using Wilcoxon Signed Rank tests with

	Right	Wrong	Decision accuracy
INFO_1	15	11	57.69%
INFO_2	20	6	76.92%
INFO_3	22	4	84.61%

Table 6.7: Decision accuracy under the effect of different types of decision support information. Columns 'Right' and 'Wrong' contain the number of pilots who made the right or wrong decision respectively.

Bonferroni correction shows that the stronger decrease in DT is given by INFO_3 ($Z = -2.984$, $p < 0.003$, N=26).

The descriptive statistics are provided in Table 6.8.

	Decision time
INFO_1	36.78 (s.d. 6.36)
INFO_2	35.02 (s.d. 5.97)
INFO_3	28.63 (s.d. 8.61)

Table 6.8: Decision time (in seconds) under the effect of different types of decision support information.

E3: number of clicks on the reconfiguration buttons (nrCL)

The statistical difference in terms of nrCL amongst the three conditions is confirmed by Friedman's test ($\chi^2(2) = 26.297$, $p < 0.001$, N=26).

As expected, a progressive decrease of nrCL with INFO_2 and INFO_3, with respect to the baseline (INFO_1), is revealed by a post-hoc test performed through a series of Wilcoxon Signed Rank tests (INFO_2 vs INFO_1: $Z = -3.326$, $p < 0.001$, $r = 0.652$; INFO_3 vs INFO_1: $Z = -3.968$, $p < 0.001$; INFO_3 vs INFO_2: $Z = -2.057$, $p < 0.04$). These tests show that the biggest decrease of nrCL w.r.t the baseline is provided by INFO_3. The descriptive statistics are provided in Table 6.9.

	nrCL
INFO_1	3.81 (s.d. 1.17)
INFO_2	2.88 (s.d. 1.07)
INFO_3	2.27 (s.d. 0.72)

Table 6.9: Number of clicks on the reconfiguration buttons under the effect of different types of decision support information.

E4: fixation duration (FD)

Friedman's test reveals a significant influence of the independent variable on the FD ($\chi^2(2) = 17.583$, $p < 0.01$, N=24). The descriptive statistics are shown in Table 6.10.

The biggest decrease of FD is provided by INFO_3 over INFO_1, as statistically confirmed by the Wilcoxon Signed Rank post-hoc test with Bonferroni correction ($Z = -4.229$, $p < 0.001$, N=24).

	Fixation duration
INFO_1	410.01 (s.d. 10.55)
INFO_2	379.13 (s.d. 9.32)
INFO_3	354.89 (s.d. 7.04)

Table 6.10: Fixation duration (in milliseconds) under the effect of different types of decision support information.

E5 and E6: workload (WL) and frustration (FR)

Table 6.11 reports the results of the NASA-TLX test for each type of decision support information.

Workload (NASA-TLX) - Test results			
	INFO_1	**INFO_2**	**INFO_3**
MD	71.92 (3.42)	63.46 (3.9)	50.00 (4.38)
PD	1.92 (1.21)	1.54 (0.87)	1.15 (0.61)
TD	32.31 (3.47)	27.69 (2.81)	31.15 (3.01)
PE	54.62 (3.94)	58.08 (4.1)	78.85 (2.34)
EF	54.23 (5.71)	46.54 (3.37)	34.23 (3.66)
FR	61.23 (5.72)	52.31 (4.03)	25.00 (2.59)
OWL	*52.23 (2.91)*	*47.15 (1.77)*	*39.1 (1.83)*

Table 6.11: NASA-TLX data under the effect of different types of decision support information.

A one-way ANOVA test is run on each parameter of the NASA-TLX test (i.e. Mental Demand (MD), Physical Demand (PD), Temporal Demand (TD), Performance (PE), Effort (EF) and Frustration (FR), and on the Overall Workload (OWL)). A strongly significant effect for the independent variable is found on all the parameters except PD and TD (see Table 6.12).

Workload (NASA-TLX)	
Workload parameter	**ANOVA result**
MD	$F(2, 37) = 7.95$, $p < 0.001$
PD	$F(2, 37) = 0.171$, n.s.
TD	$F(2, 37) = 0.597$, n.s.
PE	$F(2, 37) = 13.605$, $p < 0.001$
EF	$F(2, 37) = 5.32$, $p < 0.009$
FR	$F(2, 37) = 19.219$, $p < 0.001$
OWL	$F(2, 37) = 8.802$, $p < 0.001$

Table 6.12: Results of the one-way ANOVA test on the NASA-TLX results.

The Tukey HSD post-hoc test (Table 6.12) reveals that INFO_3 provides more improvement then INFO_2 on the baseline INFO_1. Furthermore, a statistical improvement of INFO_3 is confirmed on INFO_2 for PE, FR and OWL.

As one of the parameters of the NASA-TLX method is frustration (FR), this technique allows analysis concerning expectation E6. Tables 6.11 and 6.12 reveal that FR decreases statistically with both INFO_2 and INFO_3, confirming the effectiveness of the decision support information produced by SaIRA.

Workload (NASA-TLX)		
Workload parameter	**INFO_3 vs...**	**Tukey HSD**
MD	INFO_1	$M = -21.923,\ p < 0.01$
	INFO_2	$M = -13.462,\ n.s.$
PE	INFO_1	$M = 24.231,\ p < 0.01$
	INFO_2	$M = 20.769,\ p < 0.01$
EF	INFO_1	$M = -20,\ p < 0.007$
	INFO_2	$M = -12.308,\ n.s.$
FR	INFO_1	$M = -36.231,\ p < 0.01$
	INFO_2	$M = 27.308,\ p < 0.01$
OWL	INFO_1	$M = -13.128,\ p < 0.01$
	INFO_2	$M = -8.051,\ p < 0.039$

Table 6.13: Results of the Tukey HSD post-hoc test for the NASA-TLX test. INFO_3 provides the biggest statistical difference for all the combinations except w.r.t INFO_2 in relation to MD and EF.

E7: situation awareness (SA)

SA-SWORD does not provide a direct measure of SA but it is designed to give an assessment of which type of information gives the highest SA. As expected, INFO_2 and INFO_3 provide progressively better SA with respect to INFO_1, as shown in Figure 6.12.

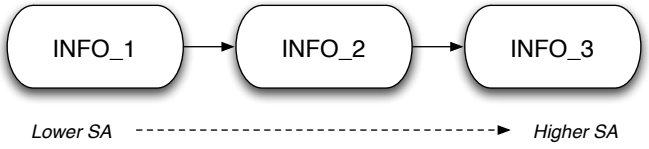


Figure 6.12: Result of SA-SWORD test for Experiment C.

A one-way ANOVA reveals the strong effect of the independent variable on the subjective assessment of SA ($F(2, 37) = 1860.943$, $p < 0.001$). The Tukey HSD post-hoc test shows that INFO_3 gives the greatest improvement.

The results of this experiment are summarised in Table 6.14 and Figures from 6.13 to 6.17.

Experiment C						
	DA (%)	**DT (sec)**	**nrCL (count)**	**FD (msec)**	**WL**	**SA (ranking)**
INFO_1	69%	35.22 (3.56)	3.54 (0.37)	410.01 (10.55)	52.23 (2.91)	3 rd
INFO_2	58%	35.6 (2.01)	3.08 (0.36)	379.13 (9.32)	47.15 (1.77)	2 nd
INFO_3	92%	28.52 (2.72)	2.15 (0.19)	354.89 (7.04)	39.1 (1.83)	1 st

Table 6.14: Overall results for Experiment C.

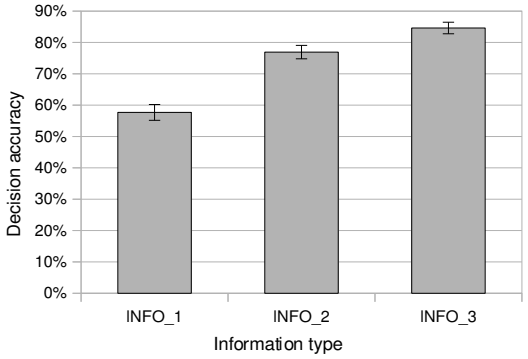


Figure 6.13: Decision accuracy (percentage of right decisions)

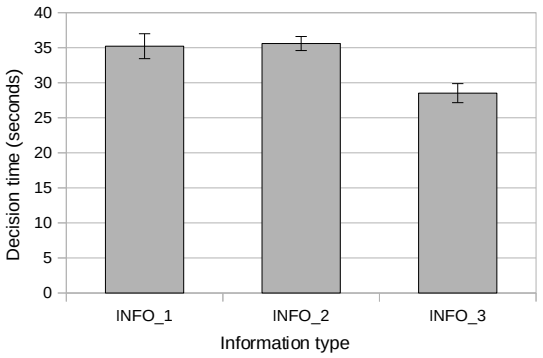


Figure 6.14: Decision time (in seconds)

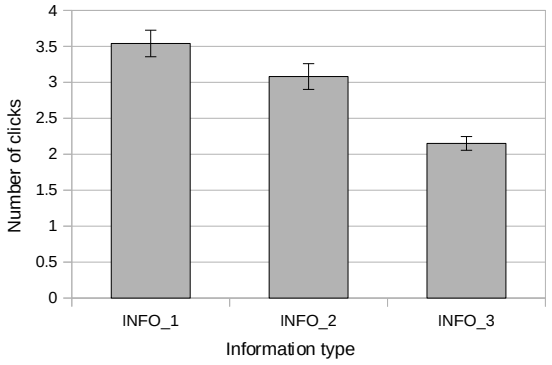


Figure 6.15: Number of clicks on the virtual buttons for configuration switch.

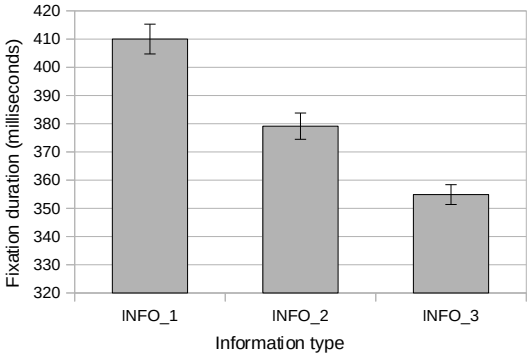


Figure 6.16: Fixation duration (in milliseconds)

Discussion

The main result is that in general the effectiveness of the complete set of decision support information generated by SaIRA (i.e. INFO_3) is strong in terms of all the dependent variables considered. To a certain extent, DA, nrCL, FD, WL, FR and SA all behaved as expected, providing evidence of a significant improvement in all aspects of pilot decision experience during ADR. An improvement is also found in terms of DT, which was not predicted, for reasons given at the beginning of

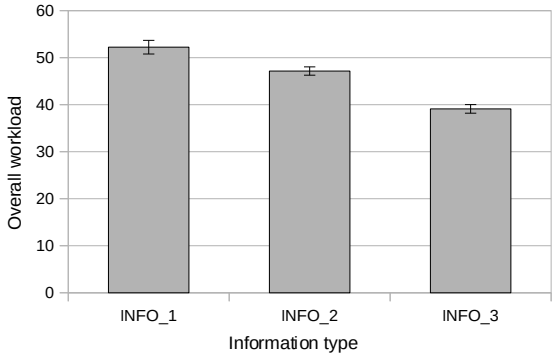


Figure 6.17: Workload ('Overall Workload' parameter from the NASA-TLX test)

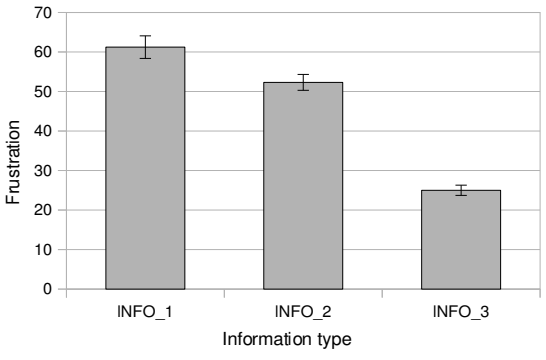


Figure 6.18: Frustration (as recorded by the NASA-TLX test)

this section.

The improvement brought by a graphical representation of the fault over the baseline textual information set (i.e. INFO_2 *versus* INFO_1) is not as strong as in other studies such as Hayashi et al. [2006], which evaluated the FAMSS architecture for the next generation spacecraft cockpit (see Section 4.2.7). It must be noted, however, that projects like FAMSS are specifically targeted at the design of effective graphical representations of fault management information, whilst this study is tailored to the analysis of the effects of explanations, implications and reliability information on the interactive fault management process. One possibility is that the graphical information generated by SaIRA is not as sophisticated and effective as the information produced by more advanced graphic engines like FAMSS. It would be interesting to analyse a combination of the two approaches.

An unexpected result comes from the NASA-TLX; pilots ranked their performance higher in the scale with INFO_3 than with other information formats. In this regard, Fox and Tversky [1995] argue that feelings of competence occur when people have clear rather than ambiguous knowledge. INFO_1 and INFO_2 provide less information than INFO_3, therefore, there is a possibility that the former two types of information leave room for ambiguity in the pilots' minds. With reference to the *support theory* of reasoning [Tversky and Koehler 1994], the content of INFO_3 is "unpacked" into more explicit disjunctions, a fact that, according to the theory, increases the "strength of belief" of the decision maker and decreases the ambiguity. We speculate that, as a result of this phenomenon, pilots would feel more competent and give themselves a higher performance score.

It is particularly important for this research to comment on the positive effect of INFO_3 on frustration, which provides further support to the effectiveness of the SaIRA concept.

Finally, a remark should be made about **cognitive readiness**, which has been defined by Morris and Fletcher [2002] as the mental preparation (including knowledge, skills, abilities and attitudes) an individual needs to establish to sustain competent performance in complex and unpredictable operational environments. Morris and Fletcher identify ten psychological components or theoretical mechanisms underlying the concept of cognitive readiness: situation awareness, memory, transfer of training, metacognition, automaticity, problem solving, decision-making, mental flexibility and creativity, leadership and emotion. The empirical results obtained in this experiment reveal that the decision support approach proposed for ADR decisions improves human decision accuracy, decision performance and situation awareness. The improvement of these quantities reflects, by definition, a general contribution to the improvement of the cognitive readiness of the decision maker.

It is observed that the definition of cognitive readiness previously given is applied in the literature to different stages of the preparation for a task. Several studies focus on manipulating cognitive readiness through training the decision makers well before performing the task. Here cognitive readiness is manipulated by means of decision support information, shown at a relatively close distance from the decision-making deadline. This interpretation seems appropriate for the domain and the problem being addressed here.

In conclusion, the full set of SaIRA decision support information is found to verify all the claims about its effectiveness and as a consequence to provide a strong, statistical improvement

in pilot decision experience during ADR, confirming the user profile proposed in the previous chapters.

6.10 Experiment D – Time Pressure and Implications

Description and aim

Experiment D aimed to investigate the effect of *time pressure* on ADR decisions, encompassing claims 2, 3 and 4. Furthermore, evidence was collected about the effect of the availability of the *implications* of each reconfiguration alternative in time pressure circumstances. The claims are reported below:

> **Claim 2 - Time pressure**: Pilots would react to time pressure by means of a) acceleration, b) selection of information and c) alteration of information search pattern.
>
> **Claim 3 - Time pressure**: Time pressure would decrease pilot decision accuracy.
>
> **Claim 4 - Time pressure**: Time pressure would reduce the number of configuration options considered and the way the information is explored.

As reported in Section 4.1.2, some studies question the order of appearance of the three strategies mentioned by Claim 2 and even their actual appearance in real unstructured decision contexts. This experiment does not question the conclusions of Ben Zur and Breznitz [1981]; Edland and Svenson [1993*a*] and Johnson and Payne [1995], which are at the basis of the claim; instead the interest is in investigating the applicability of these ideas to the ADR problem, in order to characterise effective decision support. In addition, this experiment only investigates the potential emergence of any of the strategies in question, not their hypothetical order of appearance.

Two independent variables were examined in this experiment: *time pressure* and *implication availability*. The pilots performed ADR under normal conditions and under time pressure; when under time pressure, simulations were performed with and without the availability of 'implications'. The rationale for this decision was elaborated in Section 4.2.3. In brief, we argue that in severe time pressure situations, pilots would discard a great portion of the decision support information, but would look for the implications of their decisions.

The *decision time*, *decision accuracy*, *fixations duration* and the *distribution of visual attention* on the cockpit displays (dependent variables) were used to draw conclusions about the case.

Procedure

X-Plane was configured to run three flight simulations characterised by similar flying conditions. The pilot was asked to complete any potential real-time fault management procedures correctly, in the shortest time possible.

In addition to full SaIRA decision support information (i.e. INFO_3), a *predictive countdown timer* showing the time remaining to complete the ADR process before the advent of serious consequences was shown in the top-right corner of the EHSI display. Pilots were told that missing the deadline equated to a failure.

Pilots were also informed that, in some cases, the system could fail to generate the whole set of decision support information that constituted INFO_3. This didn't necessarily mean that the reconfiguration option suggested was wrong; it just lacked a piece of information that could not be computed for technical reasons. Pilots did not know which chunk of information would be missing in advance. In point of fact, for all cases, the information in question was the 'implications'; as mentioned in Section 4.2.3 (page 94), we argue that the implications of reconfigurations would be particularly effective in improving the decision performance especially under time pressure conditions.

A safety-critical fault was simulated between 1 and 2.5 minutes after the start. The system generated 20 equally applicable reconfiguration options.

Three scenarios with the following characteristics were executed (in random order):

- **Test 1**: with 30 seconds of maximum decision time available *(time pressure)*;
- **Test 2**: with 120 seconds of maximum decision time available *(no time pressure)*;
- **Test 3**: with 30 seconds of maximum decision time available but no implications shown on the display *(time pressure and no implications)*.

Test 2 represents the baseline case; Test 1 investigated the effects of time pressure alone; Test 3 investigated the effects of lack of implications under time pressure. The order is mixed in order to prevent pilots from identifying the baseline, which is of critical importance for this experiment. This is a within-subject experiment.

Expectations

The following results were expected:

- E1: *decision accuracy (DA)* would decrease with time pressure; this effect would be strengthened by the lack of implications;
- E2: pilots would react to time pressure by means of a) acceleration, b) selection of information or c) alteration of information search pattern. More specifically:
  - E2a - acceleration: the *fixation duration (FD)* would decrease with time pressure. It was more difficult to make predictions concerning the effect of the availability of implications. On the one hand, availability of implications should reduce the problem complexity, therefore reducing FD; on the other hand, implications constitute an additional piece of information to look at and this could complicate matters for the pilots. In both cases, there should be an observable effect on FD;
  - E2b - selection: the distribution of *visual attention (VA)* would show that, in the controlled simulations, pilots focus only on a specific subset of the decision support information available, which certainly includes the 'implications';

– E2c - search pattern: the *number of configuration options explored (nrCNF)* would be effected; furthermore, the *VA* would show an alteration in the way the available information was accessed, with pilots focusing more on implications and fault description, which is symptomatic of a less cognitive demanding attribute-based search in place of an alternative-based search.

Nevertheless, it is interesting to check any potential effect on *decision time (DT)* under the two test conditions in question. E1 assesses Claim 3; E2 encompasses claims 2 and 4.

Results

E1: decision accuracy (DA)

A slight decrease of DA was found as a result of time pressure (Table 6.15 and Figure 6.19) but it is not statistically significant (Cochran's Q test: $\chi^2(2) = 0.846$, n.s. N=13).

	Right	Wrong	Decision accuracy
120 seconds	10	3	76.9% (s.d. 12.2%)
30 seconds	9	4	69.2% (s.d. 9.2%)
30 seconds, no implications	9	4	69.2% (s.d. 9.2%)

Table 6.15: Decision accuracy under the effect of time pressure and implications availability.

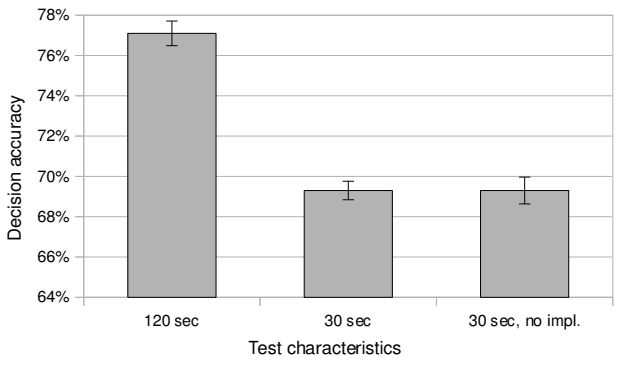


Figure 6.19: Decision accuracy (number of right decisions over the total number of decisions)

Whilst indicative, this result is not strong enough to support E1. The lack of a main effect on DA is probably due to the fact that pilots are expert decision makers, particularly accustomed to time pressure conditions, which is intrinsically part of their usual operating environment. Regardless of any physiological impact of time pressure, which is investigated below, pilots managed to make the right ADR decision even with a limited time budget.

E2: fixation duration

A statistically significant effect of the test factors is found on FD (Friedman's test: ($\chi^2(2) = 10.308$, $p < 0.006$, N=13). The test results are reported in Table 6.16 and Figure 6.20.

	FD
120 seconds	408.51 (s.d. 9.39)
30 seconds	379.16 (s.d. 10.05)
30 seconds, no implications	384.64 (s.d. 10.49)

Table 6.16: Fixation duration (FD) under the effect of time pressure and implications availability.

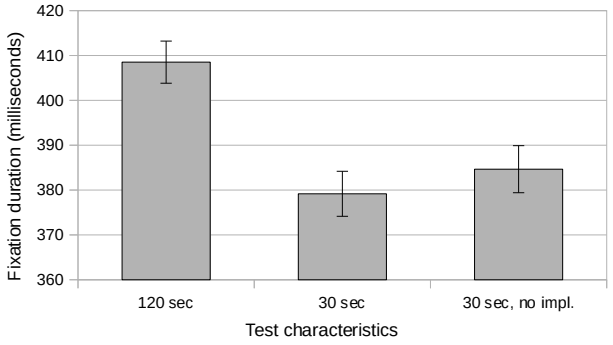


Figure 6.20: Fixation duration (in milliseconds)

Table 6.17 contains the result of the post-hoc test on FD (a series of Wilcoxon's tests with Bonferroni correction). The main discrepancy is recorded between *120 seconds* and *30 seconds* of time budget. Furthermore, implications slightly increase FD under time pressure but this effect is not statistically significant.

The tests confirm the claim concerning the emergence of the acceleration strategy as a result of time pressure; given the small number of participants, we cannot confirm that lack of implications has the consequence of slightly braking this effect.

	120 seconds vs 30 seconds	**120 seconds vs 30 seconds, no impl.**	**30 seconds vs 30 seconds, no impl.**
Z	1.231	0.846	0.385
Asymp. Sig. (2-tailed)	$p < 0.05$	n.s.	n.s.

Table 6.17: Wilcoxon Signed-Rank post-hoc test with Bonferroni correction for Friedman's test on FD (descriptive statistics in Table 6.16).

E2: decision time, number of configuration alternatives explored

Table 6.18 reports the statistics relative to the number of configuration alternatives (nrCNF) explored by pilots during ADR decisions and the decision time (DT).

Friedman's test reveals a statistical difference between the three experimental conditions for both nrCNF ($\chi^2(2) = 17.915$, $p < 0.01$, N=13) and DT ($\chi^2(2) = 14.308$, $p < 0.01$, N=13). A follow-up test using a series of Wilcoxon Signed-Rank tests with Bonferroni correction (see Table 6.19) shows that, whilst both '30 seconds' and '30 seconds, no implications' both differ statistically from the baseline, the difference *between* them is not statistically significant. In other

words, implications seem not to affect the time spent by pilots reaching a decision, even if as seen in the previous section with FD, there is some evidence of increased problem complexity.

	nrCNF	**DT**
120 seconds	3.77 (s.d. 0.28)	49.12 (s.d. 4.6)
30 seconds	2.00 (s.d. 0.11)	27.33 (s.d. 0.65)
30 seconds, no implications	2.85 (s.d. 0.15)	27.15 (s.d. 0.5)

Table 6.18: Number of configuration alternatives explored under the effect of time pressure and implications availability.

	120 seconds vs 30 seconds	**120 seconds vs 30 seconds, no impl.**	**30 seconds vs 30 seconds, no impl.**
Z	-3.11	-3.04	-0.035
Asymp. Sig. (2-tailed)	$p < 0.001$	$p < 0.001$	n.s.

Table 6.19: Wilcoxon Signed-Rank post-hoc test with Bonferroni correction for the Friedman's test on DT (descriptive statistics in Table 6.18).

It is worth noticing that the standard deviation for DT is very small for Test 2 and Test 3, whilst it is larger for Test 1, which has a greater time budget. Evidently, pilots intentionally use the full 30 second window that is available to them to complete the task; the button that confirms their decision is clicked at the very last second. None of the pilots missed the deadline in any test. These results are expected from skilled and expert decision makers like pilots. They exploit all the available time and still rarely miss the deadline.

A correlation is found between nrCNF and DT (Spearman's test: $\rho = 0.519$, $p < 0.001$, N=13) which is evident looking at Figures 6.21 and 6.22. This correlation reinforces our interpretation of nrCNF as a measure of indecision.

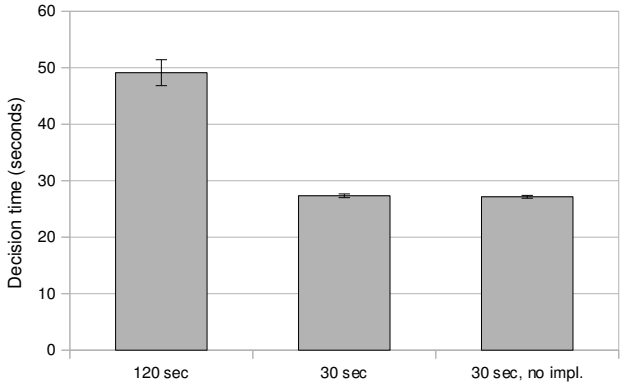


Figure 6.21: Decision time (in seconds).

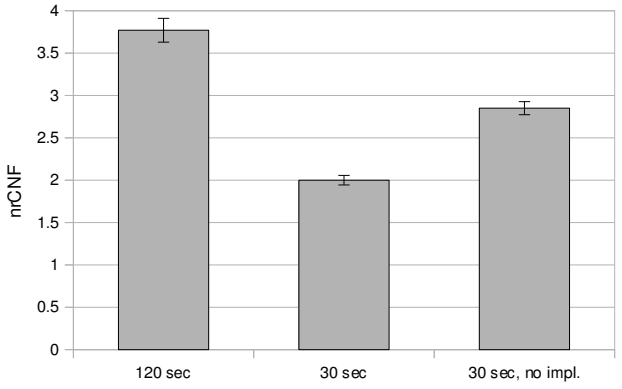


Figure 6.22: Number of configuration options explored.

E2: distribution of visual attention

Figure 6.23 and Table 6.20 describe the trend of pilots' visual attention on five relevant AOIs in the cockpit. Note that EHSI_2 contains the implications of the configuration alternatives, and EADI_1 contains the graphics, which are both under analysis in this experiment.

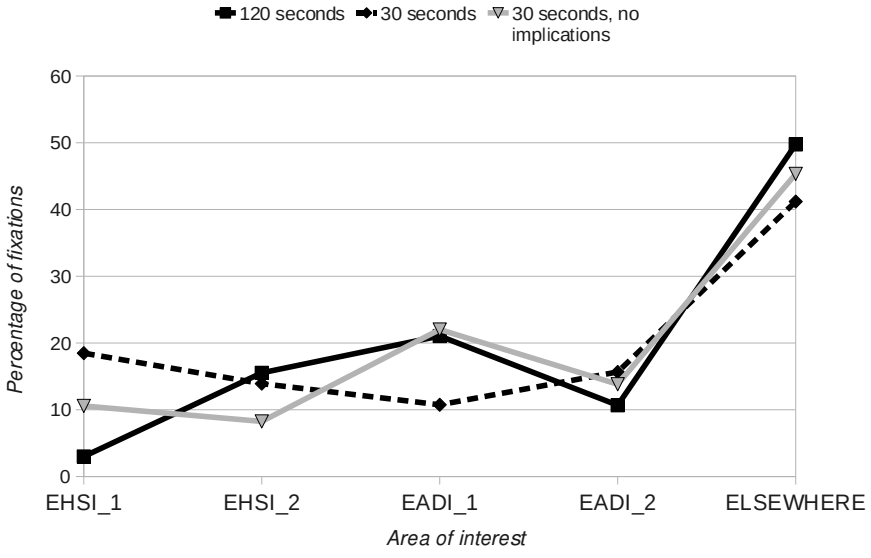


Figure 6.23: Percentage of on-target fixations relative to each AOI.

We ran five different Friedman's tests on the data related to each AOI. The results, reported in Table 6.21, show a statistically significant effect on each AOI.

Discussion

The first encouraging result is that in the specific conditions tested, pilot decision accuracy seems not to be statistically influenced by time pressure or by lack of implications, as predicted by E1. However, decision performance and physiological reactions are both significant and follow the predictions of claims 2, 3 and 4. This general result leads us to leave open the possibility that

	120 seconds	30 seconds	30 seconds, no implications
EHSI_1	3.0 (s.d. 0.16)	18.48 (s.d. 0.83)	10.57 (s.d. 0.86)
EHSI_2	15.53 (s.d. 1.49)	13.92 (s.d. 0.83)	8.25 (s.d. 0.73)
EADI_1	21.04 (s.d. 1.63)	10.75 (s.d. 0.74)	22.02 (s.d. 0.94)
EADI_2	10.67 (s.d. 0.84)	15.67 (s.d. 0.99)	13.82 (s.d. 0.62)
ELSEWHERE	49.75 (s.d. 2.34)	41.19 (s.d. 1.73)	45.33 (s.d. 1.24)

Table 6.20: Percentage of on-target fixations relative to each AOI.

	Friedman's test (N=13)
EHSI_1	$\chi^2(2) = 24.154$, $p < 0.01$
EHSI_2	$\chi^2(2) = 14.308$, $p < 0.01$
EADI_1	$\chi^2(2) = 16.615$, $p < 0.01$
EADI_2	$\chi^2(2) = 6$, $p < 0.05$
ELSEWHERE	$\chi^2(2) = 7.538$, $p < 0.023$

Table 6.21: Results from five different Friendman's tests related to each AOI under analysis.

in more complex and realistic situations, considering a wider number of pilots, time pressure and lack of implications could also impact on DA.

Concerning E2, the results of the FD, DT, VA and nrCNF analyses confirm claims about acceleration, selection and search pattern. The results from VA confirm that pilots value implications during the decision process, and nrCNF confirms that a lack of implications increases pilots confusion, making them jump between roughly 3 alternatives instead of 2 (note that in this experiment all configuration alternatives provided by SaIRA are correct and equally applicable).

With reference to Figure 6.23, the first clear result is the inversion of the convexity of the VA curves of the three test conditions from EHSI_1 up to EADI_1. When pilots have more time (120 seconds) they concentrate more on explanations, implications and graphical information (EHSI_2 and EADI_1) whilst under time pressure most of the VA on the graphical information (EADI_1) is distributed elsewhere.

EHSI_2 (implications and explanations) is steadily used in all three test conditions; there are no significant drops in attention in this area. Obviously, the VA is slightly lower during Test 3 because implications are not available, hence part of the attention is driven away from this AOI; however, explanations keep attracting the participants. This result suggests that pilots actually value explanations and implications during ADR decisions.

VA is higher on the EHSI_1 during the two tests characterised by time pressure (i.e. Test 1 and Test 3). The most probable explanation is that this AOI contains the timer and evidently, pilots make more intensive use of it when the time budget is very limited. A similar effect was noticed in a study on decision making under time pressure by Zakay [1993], in which also a decrease in accuracy was also found.

The facts that (a) the VA on the ELSEWHERE AOI is statistically different under the three test conditions and (b) the mean of VA on ELSEWHERE is lower under time pressure, both seem to suggest that, in proportion, pilots use the decision support information more intensively when the time budget is limited. More time to make a decision allows them to concentrate on other

information elsewhere in the cockpit.

In conclusion, the effect of time pressure is confirmed and it follows the behaviour predicted in the claims. Explanations and implications are found to be intensively used by pilots during ADR under all conditions, reducing attention to graphical information during the occurrence of severe time pressure. Implications are found to decrease decision uncertainties, confirming their importance and effectiveness. More pilots and a more realistic scenario are required to draw robust conclusions concerning the effects on decision accuracy.

6.11 Experiment E – Information Correctness and Reliability

Description and aim

Section 4.1.4 (page 84) discussed the framing effect, ambiguity aversion and loss aversion phenomena. In the light of these decision biases, we previously argue that the *degree of reliability* associated with decision support information would influence pilot decision behaviour. More specifically, as part of the definition of the user profile for SaIRA, we speculated that reconfiguration suggestions characterised by medium reliability would be more difficult to process than those presented as either strongly or poorly reliable.

As discussed in Appendix C, SaIRA associates four levels of reliability with the decision support it generates, on the basis of the reliability of the data sources: LOW, MEDIUM, HIGH, FULL. In all the experiments except this one, SaIRA was manipulated to either not show reliability information or, when shown, it was always set to FULL. Here this simplification no longer holds.

In this experiment LOW or MEDIUM reliability is associated with the fault management information presented to pilots in order to prompt them to use their judgement; this scenario simulates the case in which the system recognised that, given the characteristics of the input data available, the inferences made are not robust, therefore it is mandatory for pilots to consider the suggestions more carefully.

The aim was to collect information about the potential effect of the different degrees of reliability (degREL) associated with decision support information of dubious authenticity on pilot decision behaviour. In particular, we argue that LOW reliability would ease the process of identifying and discarding wrong decision suggestions, whilst MEDIUM reliability would increase the *complexity* of the decision even beyond the baseline (i.e. right information, fully reliable), with the potential by-product effect of impairing the decision performance.

The claim of reference is Claim 10:

> **Claim 10 - Reliability figures**: Providing *reliability* figures would influence pilot decision-making performance in the following ways:
>
> - the framing effect would emerge when providing pilots with reliable or uncertain information; more specifically, pilots would feel more comfortable applying a configuration associated with high reliability than with low uncertainty;

- evidently wrong ADR suggestions, when associated with low reliability, would be more easily spotted;
- low and high reliability options would both be easier to process than medium reliability options, i.e. the decision time would increase with medium reliability options.

The first point is elaborated in more detail by Claim 10; the analysis of this claim is described later in this chapter in Experiment G.

Procedure

X-Plane was configured to run 3 scenarios. Similar flight conditions were set for all cases.

In the other experiments, the pilot was told, before undertaking the tests, that in case of fault he/she would be supported by SaIRA, in order to efficiently solve the problem. The pilot was also informed that the system could potentially generate wrong decision support information, as a result of technological limitations.

The pilot was asked to complete any potential real-time fault management procedure correctly and in the shortest time possible.

A safety-critical fault was simulated between 30 and 120 seconds after the start. The system was configured to generate only one configuration option; the pilot could either accept it or switch to safe mode.

Pilots were divided into two groups: Group A and Group B. All pilots performed Test 1; then Group A performed Test 2a and Group B performed Test 2b, as follows:

- **Test 1 - both Group A and Group B**: SaIRA generated the *correct* decision support information characterised by *FULL reliability*. This was the baseline test, aimed at building pilot confidence in the system before providing them with the wrong information;
- **Test 2a - Group A only**: SaIRA generated the *wrong* decision support information characterised by *LOW reliability*;
- **Test 2b - Group B only**: SaIRA generated the *wrong* decision support information characterised by *MEDIUM reliability*;

The selection of pilots for Group A or B was randomised.

Expectations

The following results were expected:

- E1: *workload (WL)* would be higher with MEDIUM reliability than with LOW or FULL reliability;
- E2: *fixation duration (FD)* would be higher with MEDIUM reliability than with LOW or FULL reliability;

- E3: *decision time (DT)* would be higher with MEDIUM reliability than with LOW or FULL reliability.

E1 and E2 have the common aim of characterising the workload from both a subjective and objective (physiological) perspective.

The three expectations put forward cover all the points of Claim 10 apart from *decision accuracy (DA)* (DA is supposed to decrease with MEDIUM reliability). The rationale for excluding DA from this analysis is that the number of pilots and simulations in question, combined with the level of measurement of this variable (DA is nominal, i.e. a decision can either be right or wrong in this experiment) make the statistical power insufficient to provide significant results (each group is made of 6 pilots, each of them has 50% probability of avoiding the wrong suggestion by chance, instead of as a consequence of the system presenting it with LOW reliability).

Results

This experiment has a mixed factorial design. The two independent variables are the *correctness* of decision support information (which can be either 'correct' or 'incorrect', with the former being the baseline condition) and its *reliability* (either 'LOW', 'MEDIUM' or 'FULL', with FULL being the baseline condition). Correctness is the within-subject independent variable (i.e. all pilots test both its conditions) and reliability is the between-subjects independent variable (so that Group A is tested with the 'LOW' reliability condition and Group B is tested with the 'MEDIUM' condition).

For FD and DT, the main effect of both correctness (C) and reliability (R) is assessed. When ANOVA is used (for WL), the interaction between the correctness and reliability factors is also assessed (CR).

It must be noted that the main objective of this experiment, as previously stated, is to investigate the effect of the 'reliability' factor. However, because of the nature of the decision support information, it was not possible to design this experiment without using both correct and incorrect information. Hence, although the focus is on the main effect of reliability, the impact of correctness can be used to enrich the conclusions about complacency obtained from Experiment B.

E1: workload (WL)

WL was measured by means of the NASA-TLX technique. The results of the analysis are shown in Table 6.22. The factor 'Group' tests for the difference in reliability (degREL) whilst 'Test' examines the effect of the correctness of the information provided. Physical demand is not reported because it was rated null by all pilots.

A two-way split-plot ANOVA was performed on each parameter of the NASA-TLX test except PD. The results for the main effect of correctness (C) and reliability (R), and for their interaction (CR) are reported in Tables 6.23, 6.24 and 6.25.

In line with E1, WL with MEDIUM reliability is higher than in the other two cases. It must be noted that WL is also higher than the baseline with LOW reliability.

Interestingly, a peak of temporal demand (TD) is recorded with MEDIUM reliability. This is an unexpected result because no time limits for decisions were set for this experiment. We

Workload (NASA-TLX) - Test results				
	Test 1	**Test 2**	**Group A**	**Group B**
MD	66.67 (2.07)	74.14 (3.83)	64.58 (2.71)	76.25 (2.83)
TD	35.42 (4.15)	45.42 (4.01)	34.17 (3.53)	46.67 (4.28)
PE	83.33 (2.56)	59.17 (2.88)	71.25 (3.8)	71.25 (5.19)
EF	54.25 (2.85)	67.5 (5.06)	57.17 (4.39)	64.58 (4.46)
FR	25.83 (2.74)	61.67 (7.24)	34.58 (2.85)	52.92 (9.74)
OWL	*43.26 (2.23)*	*59.25 (3.56)*	*47.2 (1.51)*	*55.31 (4.9)*

Table 6.22: NASA-TLX data under the effect of different degREL.

Workload (NASA-TLX)	
Workload parameter	**Effect of 'correctness' (C)**
MD	$F(1, 10) = 5.031$, $p < 0.049$
TD	$F(1, 10) = 5.294$, $p < 0.044$
PE	$F(1, 10) = 40.239$, $p < 0.001$
EF	$F(1, 10) = 7.28$, $p < 0.022$
FR	$F(1, 10) = 80.742$, $p < 0.001$
OWL	$F(1, 10) = 66.87$, $p < 0.001$

Table 6.23: Main effect of 'correctness' of the decision support information (two-way split-plot ANOVA).

speculate that the increased perception of TD is a by-product of the increased frustration and cognitive demand. NASA-TLX data was not processed in real-time (as was eye movement data), therefore it was not possible to question pilots about this result in the post-experiment interviews.

Another interesting outcome is the negative effect of LOW reliability on pilot perception of their performance. In practice, the decision accuracy (DA) results show that contrary to participant perception, performance was not statistically worse than in the baseline case. The DA results are reported in Figure 6.24 in the form of percentages.

E2: fixation duration (FD)

Table 6.26 reports the descriptive statistics concerning FD for Experiment E.

The Wilcoxon Signed-Rank test reveals no statistical effect for 'correctness' of decision support information on pilot FD ($Z = 0.706$, n.s., N=13). Either the pilots did not notice the incorrect

Workload (NASA-TLX)	
Workload parameter	**Effect of 'reliability' (R)**
MD	$F(1, 10) = 13.517$, $p < 0.004$
TD	$F(1, 10) = 4.556$, n.s.
PE	$F(1, 10) = 1.722$, n.s.
EF	$F(1, 10) = 1.686$, n.s.
FR	$F(1, 10) = 30.062$, $p < 0.001$
OWL	$F(1, 10) = 7.026$, $p < 0.024$

Table 6.24: Main effect of 'reliability' of the decision support information (two-way split-plot ANOVA).

Workload (NASA-TLX)	
Workload parameter	**Correctness/reliability interaction (CR)**
MD	$F(1, 10) = 6.211$, $p < 0.032$
TD	$F(1, 10) = 2.353$, n.s.
PE	$F(1, 10) = 1.722$, n.s.
EF	$F(1, 10) = 4.941$, $p < 0.05$
FR	$F(1, 10) = 44.716$, $p < 0.001$
OWL	$F(1, 10) = 51.563$, $p < 0.001$

Table 6.25: Interaction between 'correctness' and 'reliability' of the decision support information (two-way split-plot ANOVA).

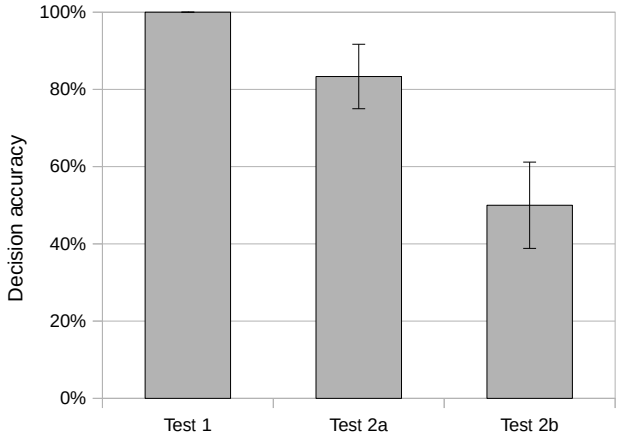


Figure 6.24: Decision accuracy (percentage of pilots who made the right decision)

information (which supports the results on complacency discussed in relation to Experiment B) or they did not have any observable physiological reactions in terms of FD.

On the other hand, the Mann-Whitney U test reveals the strong effect of the 'reliability' factor ($Z = 2.882$, $p < 0.004$, N=13); this test compares Group A and Group B *within* Test 2. The FD analysis confirms the increased complexity of processing MEDIUM reliability information.

E3: decision time (DT)

Similar results to FD were found for DT. The Wilcoxon Signed-Rank test shows no statistical effect for 'correctness' of decision support information on pilot DT ($Z = 1.883$, n.s., N=13). However, the Mann-Whitney U test reveals the statistically significant effect of the 'reliability'

	Fixation duration
Test 1	384.83 (s.d. 10.61)
Test 2	409.53 (s.d. 20.93)
Group A	371.52 (s.d. 8.56)
Group B	421.84 (s.d. 19.86)

Table 6.26: Fixation duration (in milliseconds) under the effect of 'correctness of information' (Test 1 vs Test 2) and 'reliability of information' (Group A vs Group B).

factor ($Z = 2.722$, $p < 0.006$, N=13). Table 6.27 reports the statistics.

	Decision time
Test 1	31.65 (s.d. 2.32)
Test 2	42.85 (s.d. 4.47)
Group A	32.57 (s.d. 1.8)
Group B	41.93 (s.d. 4.89)

Table 6.27: Decision time (in seconds) under the effect of 'correctness of information' (Test 1 vs Test 2) and 'reliability of information' (Group A vs Group B).

A correlation is found between FD and DT (Spearman's test: $\rho = 0.509$, $p < 0.011$, N=24), which is evident in Figure 6.26.

The results for Experiment E are summarised in Table 6.28 and Figures 6.25 and 6.26.

Experiment E			
	WL	**DT (sec)**	**FD (msec)**
Test 1	43.25 (s.d. 2.23)	31.68 (s.d. 2.32)	384.83 (s.d. 10.62)
Test 2a	48.28 (s.d. 1.23)	30.76 (s.d. 1.98)	352.78 (s.d. 7.18)
Test 2b	70.22 (s.d. 2.44)	50.95 (s.d. 5.03)	466.28 (s.d. 24.24)

Table 6.28: Overall results for Experiment E.

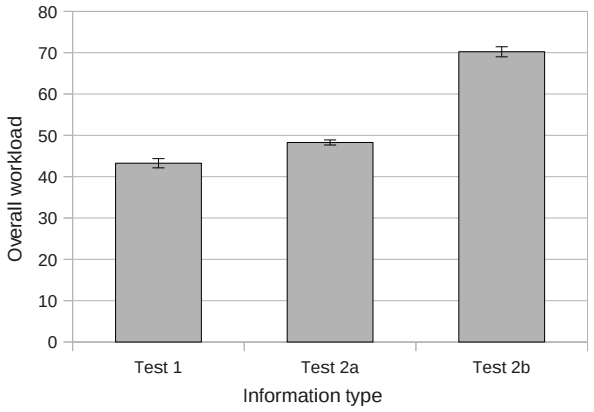


Figure 6.25: Workload ('Overall workload' parameter of the NASA-TLX test).

Discussion

All three claims are confirmed by the experimental results. The main conclusions are that (a) MEDIUM reliability worsens ADR decision performance and (b) LOW reliability improves pilot performance in discarding erroneous information. In both cases, reliability information has shown to allow pilots to make a more informed decision, which is a determining element in the design of a safety-critical system.

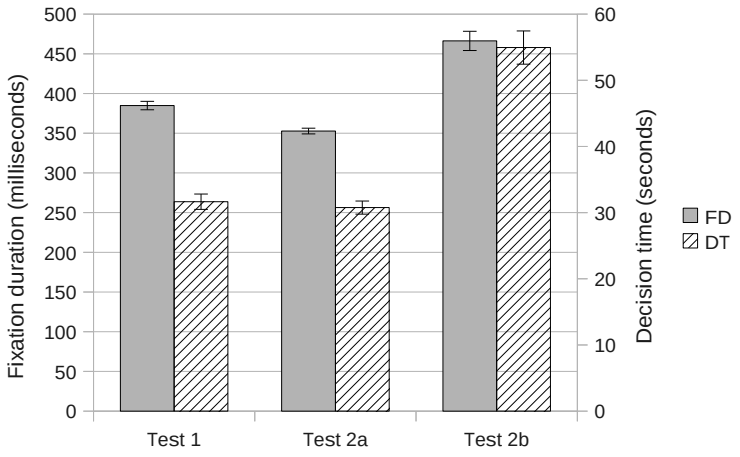


Figure 6.26: Decision time (DT) and fixation duration (FD).

Experiment B reveals that, in certain circumstances, pilots can be complacent about erroneous ADR decision support information. This experiment shows that reliability information has an effect on pilot decision performance, but no conclusions can be drawn about accuracy. Taken altogether, these conclusions give rise to a new question: can 'reliability' become a bias during ADR decisions? For instance, if SaIRA generates a correct configuration alternative, but erroneously associates it with LOW reliability, what is the reaction of the pilot? Will the pilot spend time investigating why a seemingly good suggestion is not viewed as reliable or would they be persuaded to apply another, possibly sub-optimal configuration? This is probably material for future investigation and these questions would require a new set of experiments that build on the results obtained here, so we leave them for future work.

6.12 Experiment F – Perception of Risk

Description and aim

Experiment F focuses on the effect of pilot perception of *risk* on ADR decisions, encompassing Claim 1 which is reported here for the sake of clarity:

> **Claim 1 - Risk perception**: If prompted by the system, pilots would accept a reconfiguration of the avionics in situations which are not particularly risky (e.g. whilst cruising). They would refrain from doing so in situations of pressing risk (e.g. before landing, when the system is more unstable and a change to the current state could become a catalyst for catastrophic consequences).

This is a within-subjects experiment.

Procedure

X-Plane was configured to run four simulations with similar flight conditions. The pilot was asked to complete any potential real-time fault management procedures correctly and in the shortest time possible.

The participant was also told that for technical reasons, the aircraft would have started the simulations with a sub-optimal avionics configuration, characterised by no bus redundancy and no application redundancy. As a result, both buses and LRUs constituted a single point of failure in all tests. The pilots were also told that they could improve the current state of the avionics in-flight through ADR, but *only* if they considered that action appropriate for current operating conditions.

The four simulations in question are associated with the following tests:

- **Test 1:** the pilot was required to take-off from Innsbruck International Airport, reach an altitude of 20,000 feet and hold it for 5 minutes of cruise. Between 2 and 3 minutes after reaching the target altitude, the participant was prompted with an ADR aimed at optimising the current avionics configuration.
- **Test 2:** the simulation started in the cruise phase, at an altitude of 35,000 feet, heading to San Francisco International Airport (SFO). The pilot's objective was to land at SFO. Roughly 2 minutes after starting the descent, the pilot was prompted with an ADR aimed at optimising the current avionics configuration.
- **Test 3:** identical conditions to Test 1 but taking off from London Stansted International Airport.
- **Test 4:** identical conditions to Test 2 but landing at Perscara International Airport.

Tests 1 and 3 represent the no-risk condition; Tests 2 and 4 represent the risky condition.

In all cases, SaIRA was configured to provide only one ADR alternative. During each test a similar fault was simulated (i.e. loss of a LRU) and a similar target configuration was suggested (i.e. switching off certain non-critical applications and changing the allocation of the critical software on the LRUs available). The pilot could either apply the suggested configuration or refuse to apply it, maintaining the current state of the system.

Expectations

The risks associated with the landing phase are intrinsically higher than those associated with the cruise phase. As a result, the number of pilots that would accept the application of a new configuration during Tests 2 and 4 should have been lower than during Tests 1 and 3, which take place during more relaxed phases of flight.

Results

The results, reported in Table 6.29, follow the expectations. The main effect is statistically significant (McNemar's test: $\chi^2(1, N = 26) = 8.643$, $p < 0.002$, N=26).

	Accepted	Refused	% of Accepted
Test 1	9	4	69.2%
Test 3	10	3	
Test 2	2	11	23.8%
Test 4	4	9	

Table 6.29: Descriptive statistics concerning the number of pilots who accepted/refused to proceed with the ADR. Test 1 and 3 are high-risk scenarios; Test 2 and 4 are low-risk scenarios.

Discussion

The evidence from this experiment is very important in relation to the design of the ADR process and related DSS technology. The main result is that when the perception of risk is high, pilots are not in a position to make a timely informed choice; they are over-cautious and, if a reconfiguration is really mandatory in those circumstances, then their attitude could have serious consequences.

On a civil aircraft, typical situations of this type are take-off and landing. The problem becomes more complicated in the military domain, which entails much more risky situations that are difficult to define and detect in real-time.

The aim of this work is not to define all possible situations that would impair pilot ability to supervise the ADR process; rather, the objective is to understand the nature of the effects generated by heightened levels of risk. As already discussed, the reflection effect (Section 4.1.1), which is the basis of several studies on decision making under risk and uncertainties, does not apply in this specific decisional context.

A potential way to address the issue for ADR can be found in **adaptive automation (AA)**, a form of automation that allows for dynamic changes in control function allocations between a machine and an operator, based on states of the collective human-machine system [Hilburn et al. 1997; Kaber and Endsley 2004].

According to AA theory, the degree of automation of the system should vary in proportion to the cognitive load of the operator. The theory has been developed to moderate operator workload or maintain it within predetermined acceptable limits and to preserve good SA. A very basic form of AA is currently available on modern aircraft in the form of a switch from manual to automatic pilot.

6.13 Experiment G – Reliability Framing

Description and aim

The objective of Experiment G was to investigate the potential consequences of *framing the reliability* portion of the decision support information generated by SaIRA in different ways. More specifically, Claim 6 is assessed:

> **Claim 6 - Framing effect**: Pilot decision behaviour during ADR would be subject to the framing effect; more specifically, presenting ADR information in terms of its *reli-*

ability instead of its *uncertainty* would make pilots more comfortable with accepting the application of the proposed configuration.

This is a within-subjects experiment.

Procedure

X-Plane was configured to run two tests with similar flight conditions. The pilot was asked to complete any potential real-time fault management procedures correctly and in the shortest time possible.

SaIRA was configured to show decision support information in INFO_3 format and to provide pilots with only one configuration option; the pilot could either apply the suggested configuration or refuse to apply it, switching to safe mode.

The two tests had the following characteristics:

- **Test 1:** a fault was simulated between 2 and 3 minutes after the start of the simulation; the pilot was prompted with an ADR request characterised by *70% reliability*;
- **Test 2:** a fault was simulated between 2 and 3 minutes after the start of the simulation; the pilot was prompted with an ADR request characterised by *30% uncertainty*;

The usual nominal scale used by SaIRA to present reliability information (i.e. ranging from 'LOW' to 'FULL' reliability) was replaced by percentage values for this experiment in order to make sure that exactly the same value was expressed both in terms of reliability and its opposite (see Section 4.1.4, page 84).

After Test 2 the pilot was asked to answer the question on the form shown in Figure 6.27.

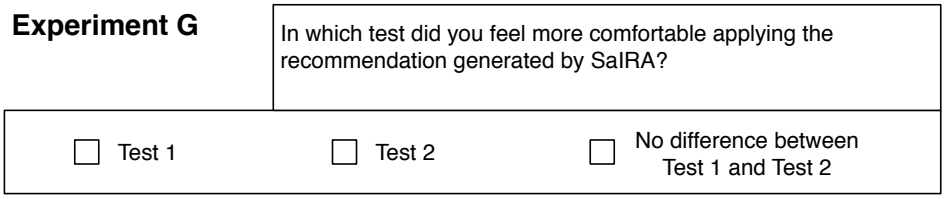


Figure 6.27: Format of the question submitted to pilots after Test 2.

Expectations

Due to the framing effect and loss aversion phenomena, the number of pilots that would accept the application of the reconfiguration should have been higher in Test 1 than in Test 2.

Results

In agreement with the expectations, 11 pilots accepted the reconfiguration in Test 1 and 5 pilots did so in Test 2 (see Table 6.30). McNemar's test reveals the statistical significance of this result ($\chi^2(1, N = 13) = 4.167$, $p < 0.031$).

	Accepted	Refused
Test 1 (with *reliability*)	11	2
Test 2 (with *uncertainty*)	5	8

Table 6.30: Number of pilots who accepted or refused to reconfigure during Test 1 and Test 2.

Interestingly, the Wilcoxon Signed-Rank test also reveals an effect on the on-target *fixation duration (FD)* relative to the EHSI_1 AOI, which contains the reliability/unreliability information ($Z = 2.83$, $p < 0.005$, N=13); see Figure 6.28.

The descriptive statistics concerning fixation duration are reported in Table 6.31.

	FD
Test 1 (with *reliability*)	362.31 (s.d. 25.23)
Test 2 (with *unreliability*)	403.97 (s.d. 34.11)

Table 6.31: Descriptive statistics for Experiment G.

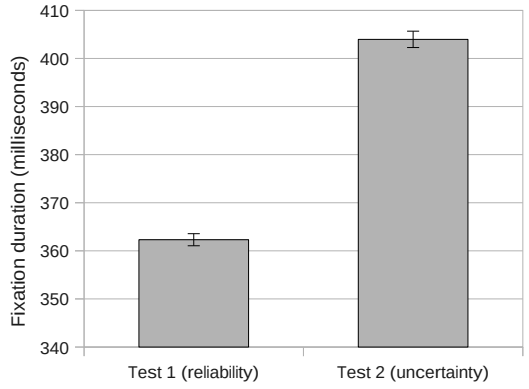


Figure 6.28: Fixation duration (in milliseconds).

The results from the question put forward after Test 2 are reported in Table 6.32. The framing effect is strikingly evident from the pilots' answers.

	Score
Test 1	8
Test 2	0
No difference	5

Table 6.32: Results of the question: *In which test did you feel more comfortable applying the recommendation generated by SaIRA?*

Discussion

The main result is the statistical confirmation of the emergence of the framing effect in relation to reliability figures associated with ADR decision support information; pilots are more likely to apply a reconfiguration when it is associated with high reliability than with low uncertainty.

This result is backed by the increased FD with 'unreliable' information, which is symptomatic of increased difficulty in processing the information in question.

The results suggest that particular attention should be paid when shaping the format of decision support information for safety-critical decisions like those used in the ADR process.

Furthermore, the results of the question submitted after Test 2 provide further support to the claim related to the framing effect of the reliability figures. Pilots should be trained not to be susceptible to the framing effect, each portion of the information provided by the system should be unequivocally explained and the scale used by the system to represent measures should be made clear.

6.14 Chapter Summary

This chapter describes seven experiments conducted with civil pilots which address different aspects of two main topics:

1. How pilots respond to ADR decisions, regardless of the design of the decision support information (Experiments A, B, D and F);

2. How the design of SaIRA decision support information influences the response of the pilots (Experiments C, E and G).

With regard to the first topic, the following material was covered:

- **Trust.** The results of Experiment A show that explaining how and why the system reaches certain conclusions enhances human trust in a DSS and system-human cooperation improves. More specifically, trust decreases the first time the system provides 'apparently wrong' suggestions, but increases again when explanations are provided. Younger pilots seem to have a higher *a priori* trust in the automated system.

- **Complacency.** Pilots were found to be complacent about basic reconfiguration decision support information (which does not include explanations, implications and uncertainty figures). In Experiment B, when provided with wrong decision suggestions, *circa* 77% of decisions taken by the pilots were wrong (i.e. erroneously complacent).

- **Time pressure.** Experiment D reveals that pilots use the decision support information more intensely when the time budget is narrow; having more time to make a decision allows them to concentrate on other information elsewhere in the cockpit. Explanations and implications are found to be intensively used by pilots during ADR under all conditions, reducing attention to graphical information during the occurrence of severe time pressure. Implications were found to decrease decision uncertainties, confirming their importance and effectiveness.

- **Perception of risk.** Experiment F shows that when the perception of risk is high (such as during a landing manoeuvre), pilots become over-cautious with respect to reconfigurations.

This attitude could have serious consequences when a reconfiguration is mandatory in risky circumstances.

Concerning the second main topic, the following conclusions were reached:

- **Availability of explanations and implications.** The SaIRA decision support information design, which includes explanations and implications of the decision alternatives, was found to improve decision accuracy and situation awareness, decrease decision time and reduce workload and frustration. The overall improvement in the decision making activity can be interpreted as a result of the improved cognitive readiness of the pilots.
- **Information reliability**. The availability of an assessment of the reliability of the fault information leads to a more informed choice and reduces complacency. However, despite the benefits, when the figures are in the middle of the scale, decision performance is decreased as a result of increased decision complexity.
- **Framing effect.** The emergence of the framing effect in relation to reliability figures associated with ADR decision support information was shown statistically. Pilots are more prone to apply a reconfiguration when it is associated with high reliability than with low uncertainty.

Table 6.33 provides a summary of the findings from all the experiments described in this chapter.

SUMMARY OF FINDINGS			
Experiment Identifier	**Independent Variable(s)**	**Dependent Variable(s)**	**Findings**
Experiment A	Information Correctness, Explanations	Trust self-assessment	Wrong decision support information decreases pilots' trust in the system. Subsequent provision of explanations for each recommendation option partially restores the trust.
Experiment B	Information Correctness	Decision accuracy	Pilots show a complacent behaviour to reconfiguration suggestions when no decision support information is provided (i.e. they accept to apply poorly described, wrong configurations)
Experiment C	Information Type	Decision accuracy, decision time, nr of clicks, fixation duration, workload and frustration self-assessment, situation awareness self-assessment	SaIRA decision support information (i.e. graphics, explanations, implications, reliability) has the following effects: complacency decrease (i.e. improved decision accuracy), decision time decrease, decision complexity decrease (i.e. decreased fixation duration), workload and frustration decrease. The improvement provided by graphics alone is not as strong as expected.
Experiment D	Time pressure, Implications	Decision accuracy, fixation duration, decision time, nr of alternatives explored, visual attention distribution	Decision accuracy is not impaired by time pressure nor lack of implications. Implications are extensively used under time pressure and their unavailability increases pilots' hesitation. With more time available, pilots make more use of explanations.
Experiment E	Information Correctness, Reliability	Workload self-assessment, fixation duration, decision time	When a medium value of reliability is associated to the recommendations, ADR decision performance decreases (i.e. the task becomes more complex). Low reliability values improve pilot performance in discarding erroneous information.
Experiment F	Perception of Risk	Pilot accept/refuse to reconfigure	Pilots are more reluctant to accept a reconfiguration in risky scenarios (e.g. just before landing) even if they would benefit from doing so.
Experiment G	Reliability Framing	Pilot accept/refuse to reconfigure, fixation duration, comfort self-assessment	Pilots are more likely to apply a reconfiguration when it is associated with high reliability than with low uncertainty

Table 6.33: Summary of the findings of the experiments.

Chapter 7

Conclusions

"Man is too quick at forming conclusions"
—Edward Emerson Barnard, Astronomer (1857-1923)

7.1 Overview and Main Contributions

This thesis investigates the problem of human involvement in the dynamic reconfiguration process of Safety-Critical Manned Systems (SCMS). In light of the complexity of the process, the current focus of mainstream research on dynamic reconfiguration of next-generation SCMS is on full autonomy and full authority solutions, which are capable of making the process as transparent to the operator as possible. This approach has inherent drawbacks which are explored in the thesis, drawing on ideas from Cognitive Psychology, Cognitive Engineering and Human Factors.

After questioning the viability and safety of fully automated dynamic reconfiguration solutions for next-generation SCMS, this thesis proposes a pragmatic but effective human-centred alternative: in the proposed framework, the operator is involved in the dynamic reconfiguration process by making critical decisions, but is assisted by a novel Decision Support System (DSS), specifically designed to parallel human cognitive strategies, as a means of addressing the complexity of the decision-making problem.

It must be noted that this thesis does not support the case for a reduction in the degree of automation on-board modern SCMS; instead it argues for the design of highly-automated systems in which the operator's role is properly taken into account early in the design phase, especially for invasive, safety-critical processes, which change the functionality of the system at run-time (e.g. dynamic reconfiguration).

Four main contributions stem from the research as a whole, summarised as follows:

1. **Literature review.** The design, development and validation of the technology proposed in this thesis required a nexus between ideas from diverse domains, including Cognitive Psychology and Engineering, Decision Support Systems Engineering, Aerospace Engineering, Human Factors, Computer Science, Human-Computer Interaction.

   The exercise of organic information synthesis presented in the first part of the thesis brings to light (a) issues with modern safety-critical technology and engineering praxis (e.g. the

drawbacks of full autonomy/authority SCMS fault management solutions) which had gone almost unnoticed in the literature up to the present day, and (b) dimensions of improvement which benefit from the coalescence of ideas from diverse domains. The review of the current state of the research and the cross-fertilisation of ideas from different disciplines paved the way for the development of SaIRA, a novel and effective cognitive engineering product.

2. **SaIRA.** All the novel algorithms and heuristics proposed in this thesis have been implemented in, and validated through, a real technology demonstrator called SaIRA. To the best of our knowledge, SaIRA is the first DSS prototype for SCMS dynamic reconfiguration problems. The framework has two main objectives: (a) generating applicable configurations for modern SCMS (such as Integrated Modular Avionics) whilst the system is operational and (b) generating effective decision support information for the operator. SaIRA represents proof of the viability and practicability of the ideas proposed in this thesis.

   Regardless of the application domain, the DSS technology is novel inasmuch as it is designed around the verified hypothesis that the accuracy and performance of the decision-makers can be improved by providing them with specific support information which parallels human cognitive strategies, such as favouring mental simulation.

3. **Experimental results.** The claims advanced in the first part of the thesis about both human behaviour during SCMS dynamic reconfiguration decisions and about the effectiveness of SaIRA were verified through a series of human-computer interaction experiments. The results reveal that SaIRA improves pilot decision accuracy, decision performance, situation awareness and more generally their cognitive readiness, whilst reducing cognitive workload and frustration under heavy time pressure.

   Regarding what concerns the novel algorithms proposed, the empirical evaluation performed against state-of-the-art methods reveals the superiority of `wsm decision-repair` *in the specific context* of combinatorial problems with similar structure and complexity to SCMS dynamic reconfiguration.

4. **Experimental methodology.** A significant contribution of this thesis is the development of a methodology for the assessment of the effectiveness of a decision support system that goes beyond classic measures (such as F-measure or ROC-measure) and merges sophisticated subjective and objective techniques. This approach allows for robust conclusions to be made about the effectiveness of the DSS, which not only take the decision results into consideration, but also human behaviour during the decision making process.

   The experimental methodology developed in this thesis was designed in the light of recent criticism against the validity of a number of mental constructs and metrics (e.g. situation awareness) which have been used in Human Factors and Human-Computer Interaction studies over the last two decades (Section 2.4). The novel approach proposed here allows for inferences that rest on coherence amongst several specifically-selected metrics of a heterogeneous nature. None of the major claims relies on a single source of data (e.g. situation awareness is assessed using a merging of eye-movement analysis data with the results of

the SA-SWORD test). The successful application of this approach, which is general enough to be extended to other human-computer interaction studies (not necessarily from the aviation domain), makes it a robust option for future HCI studies, as it is more resilient to the criticism mentioned above than mainstream methods.

In addition to the four main contributions described here, a number of additional achievements are made by this research, which are discussed in the next section.

7.2 Additional Contributions

The following additional contributions are made by this thesis:

1. **User profiling.** The effectiveness of a DSS is highly dependent on the quality of the user profile. A common *modus operandi* for the generation of user profiles for modern DSSs is to capture the behavioural aspect of the decision making whilst giving little attention to the cognitive aspect of decision support [Chen and Lee 2003], thereby neglecting the "executive mind-support systems" interpretation of DSSs, introduced in the early stages of this technology by Young [1983]. This thesis proposes a pilot profile—limited to the Avionics Dynamic Reconfiguration (ADR) scenario—which draws on ideas from the Naturalistic Decision Making (NDM) domain. The empirical assessment confirms the effectiveness of this approach.

2. **Situation awareness (SA).** This mental construct has acquired significant importance in the last two decades in several domains related to Human Factors/Ergonomics, *inter alia* military tactics, business decisions, NDM and aviation psychology. SA is specifically targeted in this thesis by the claims advanced and by the design of the HCI experiments. One major contribution is the evidence of the effectiveness, in terms of SA improvement, of the type of decision support information generated by the framework proposed. A rigorous approach to the assessment of SA is adopted, which correlates several subjective and objective metrics.

3. **Automation-Induced Complacency (AIC).** AIC is a relatively new and debated area of research in the aerospace human factors domain, which requires new contributions. The potential emergence of this phenomenon is specifically addressed in the HCI experiments; the contribution of this thesis is to reveal that the availability of explanations to justify each decision alternative has a constraining effect on the emergence of AIC. This approach stems from the original hypothesis of the effectiveness of decision support information that parallels human cognitive strategies. It would be interesting to extend the investigation performed in this research to domains other than aviation; there seems to be no reason why the results obtained in this study could not be generalised to other decision making contexts, such as control of nuclear power plants.

4. **Constraint-based decision support generation technology.** In order to prove the practicability of the approach to automated decision support information generation proposed in

this thesis, new technology aimed at producing the information in question in a programmatic manner was investigated. In order to tackle the complexity of the problem addressed, we propose a method in which an ontology for the ADR problem is developed first; the ontology is then used to support the modelling of the overall ADR problem as a constraint satisfaction problem. This approach is revealed to be effective and it is general enough to be applied to similar problems from different domains. Two novel algorithms are proposed:

- `wsm decision-repair` is an algorithm designed to handle major reconfigurations (resulting from over-constrained problems which require switching to a degraded operating mode), which exploits domain-dependent knowledge encoded at design time. The performance and scalability of the algorithm were empirically assessed and compared to standard *de facto* algorithms designed for the same purpose. The results show the effectiveness of the algorithm for the ADR problem. This algorithm is generic and can be applied to different problems.
- `SaIRA-XPlain` is an algorithm for the automated generation of explanations and implications of each reconfiguration alternative that integrates the technology discussed in the thesis. The algorithm has the key role of showing the practicability of the ideas developed in this thesis.

5. **Information masking and data reliability.** The trend towards increasing levels of autonomy and authority of automated control systems leads to the development of control interfaces that mask the current state of the system, filtering the raw information and providing the operator with only the 'most salient' portion of the data. A definition of what is 'salient' for an operator is both hard to find and controversial. This thesis maintains that whenever appropriate *the operator should be informed about the degree of uncertainty* hidden in the input data used by the system to perform its inferences. A major contribution of this work is to provide empirical evidence that unmasking this type of information enables the human to spot wrong conclusions of the system under control and, as a result, reduces the emergence of complacent behaviour, thereby improving safety.

7.3 Limits and Further Work

The problem addressed by this research is vast and has facets of a diverse nature; despite its effectiveness, the approach proposed to handle the SCMS dynamic reconfiguration process and to generate decision support information has also revealed a number of limitations and opportunities for further work which are indicated hereinafter. Some limitations of the adopted research method are also discussed.

1. **Experiments.** A number of decision biases which are relevant for the ADR problem are discussed in Chapter 4. The chapter sets up a research agenda but the list of decision biases is not intended to be comprehensive and, despite the statistical significance of the experimental results, some of the biases have only just begun to be examined; this is the case of the issues with pilot trust in the on-board automation.

There is plenty of room for more experiments which cover a larger number of aspects of the decision-making activity and provide a more in-depth examination of any facet of the human-computer interaction process being studied.

2. **SaIRA technology.** SaIRA is a novel technology. It should be regarded as a first prototype of a new class of constraint-based DSSs designed to parallel human cognitive strategies. There is room for improvement in almost every aspect of the system, including performance, content and framing of the information generated and type of interaction established with the human.

3. **Simulated environment.** A review of the literature shows that the use of simulators is becoming more and more common in studies of aviation psychology and Human Factors/Ergonomics. In spite of the documented benefits brought to the research community by this technology (such as cost reduction and accessibility), it seems reasonable to assume that certain aspects of the flight experience cannot be reproduced in a simulated environment, regardless of the quality of the simulation. The emotional responses generated by real flight experience cannot be reproduced in a laboratory for several reasons, the absence of a life threatening risk being one of the more influential. A comprehensive study of the effects experienced by participants in different types of virtual reality systems and viewing different virtual environments is presented by Sharples et al. [2008].

   The absence of life threatening risks poses limits for the examination of three factors, amongst those addressed by Chapter 4, primarily:

   - **Risk perception.** In Experiment F, risk is simulated by making pilots perform those manoeuvres that in reality are characterised by the highest degree of risk during the overall flight. The simulation should induce the feeling of a risky situation; thus, the pilots are asked to empathise with the role and with the flight scenario; however, a real scenario could alter the results. The analysis performed in this experiment should be considered as a precursor to a new experiment characterised by a more ecologically valid setting.
   - **Frustration.** By definition, frustration arises from the perceived resistance to the fulfilment of individual will; in a controlled experiment, this can be achieved in an infinite number of ways, each of them leading to a different set of secondary emotions. The well-established NASA-TLX technique is used to assess frustration during Experiment C and a statistically significant effect is recorded. However, the relevance of the results obtained is related to the simulated scenario; there is no guarantee that a real setting would not generate a set of secondary emotional responses that could potentially lead to different results.
   - **Trust.** Trust is affected by the same issues; in a real scenario pilots could show more cautious behaviour and be less trusting of the automated system.

4. **Multiple operators.** SaIRA is designed to interact with one operator. A natural continuation of the research path is to extend the framework to support multiple operators, possibly

located in different places and with different degrees and areas of accountability. This could be the case in the control of a nuclear power plant or the control of a satellite from a ground station.

The problem is not new and has acquired increasing interest in the last decade especially in the domain of situation awareness; Endsley [1997*a*] introduces the concept of **team situation awareness**—"the degree to which each team member has the information needed for his/her job"—whilst Kaber et al. [2001] propose the concept of **shared situation awareness**—"the degree to which team members have the same awareness of information requirements for team performance". Cox et al. [2007] study **distributed cognition**—"a theoretical and methodological framework that moves away from looking at the individual and instead focuses on larger socio-technical systems"—in the context of flightdeck collaboration and air-traffic control. Saikayasit and Sharples [2009] study the influence and effects of shared-representation facilities on collaboration and shared mental models development. The above research provides a good starting point for extending the functionality of SaIRA.

5. **Real-time performance.** Bespoke technology for the interactive management of the ADR process has been developed in the context of this thesis. Whilst the performance of `wsm decision-repair` is empirically examined, no claims are made concerning the applicability of the approach proposed, 'as is', for a hard real-time computing environment.

   In SaIRA a single ADR CSP is defined to handle all aspects of the reconfiguration problem, including the problem of task scheduling over the distributed network of computing modules. Whilst the efficiency of CP-based algorithms to solve this type of combinatorial problems has been demonstrated already [Cambazard et al. 2004; Hladik et al. 2008], bespoke algorithms, heuristics and schedulability analysis tools (which feature an impressive performance) have been specifically developed for modern Integrated Modular Avionics architectures, drawing on years of experience in the on-board software engineering domain [Lee et al. 2000]. An interesting opportunity for further research would be to exploit Bender's decomposition, introduced in Section 5.3.2, to integrate state-of-the-art algorithms for task scheduling within SaIRA.

6. **Information framing.** The way the decision support information is conveyed to the operator has a strong influence on the decision performance and accuracy, as demonstrated in Experiment G. This thesis focuses mainly on textual information. The framing chosen to display the information and the schema used for the text are basic; improving the current information design is a problem that could provide enough workload for a Ph.D. project on its own. In this regard, it would be interesting to investigate the effect of different textual information framing, different schemas and more sophisticated solutions which, for instance, integrate more powerful graphics. An interesting starting point could be providing pilots with a hierarchical representation of the explanations and implications, capable of capturing prospective causal relationships between them.

7. **Decision context.** All the experiments performed assume a safety-critical scenario with

a limited decision time budget. The experiments show that explanations are particularly effective at improving pilot situation awareness when the decision time budget is slightly increased. It would be interesting to study the effect of the decision support information generated by SaIRA in different decision contexts, characterised by longer time budgets. This is a promising direction of investigation, as intuitively, more time available to establish causal relationships should improve pilot situation awareness and decision accuracy.

8. **Adaptive Automation Technology.** Adaptive automation (AA) is a form of automation that allows for dynamic changes in control function allocations between a machine and an operator based on the states of the collective human-machine system [Hilburn et al. 1997; Kaber and Endsley 2004]. This concept was briefly mentioned during the discussion of Experiment F (Section 6.12) in relation to the perception of risk. It would be interesting to investigate—and to look at the effects of—dynamic changes to the type and amount of decision support information delivered to the operator during SCMS dynamic reconfiguration on the basis of the current operating conditions.

9. **Predicting pilot behaviour.** Lawson et al. [2009] investigate the problem of predicting human responses to an emergency situation. The authors use a talk-through method in which participants are asked to describe their actions in response to a scenario. An interesting extension of this research would be to adapt the method developed by Lawson et al. to SaIRA and use the predictions to improve the content of the decision support information directed to the pilots. The method used by Lawson et al. is interesting, as it would provide an additional framework to empirically verify the assumptions on mental simulation made at the beginning of this thesis.

10. **Experimental methodology.** The development of a robust experimental methodology for the type of study presented in this thesis is amongst the four main objectives achieved. Whilst the experimental results obtained are sound and show the effectiveness of the approach, several aspects of the method should be investigated further and possibly improved. It would be interesting to add new HCI techniques and metrics into the framework and examine how they correlate with the other constructs. Additionally, it would also be interesting to act on the power of the experiments, for example by increasing the number of participants, an option which was not available in the context of this research.

7.4 Concluding Remarks

The research performed for this thesis reveals, on one hand, that the exclusion of the operator of a safety-critical manned system from the system dynamic reconfiguration process represents a threat to the safety of the humans interacting with it; on the other hand, the risks of leaving too much control in the hands of the operator, without appropriate automated support, are also theoretically and empirically acknowledged. The problem of human involvement during dynamic reconfiguration, which at least in the aviation literature has gone mainly unnoticed so far, cannot be

neglected and on-board technology should be designed and developed accordingly. This problem represents the motivation for this research.

In Chapter 3 the following hypothesis was stated:

> During the process of avionics dynamic reconfiguration, decision support information that parallels cognitive strategies by including *explanations*, *implications* and an *assessment of the uncertainty* associated with the reconfiguration advice provided by the system should have a positive effect on pilot *situation awareness*, *workload*, *decision accuracy* and *performance*, thus it should improve the overall decision making effectiveness of the pilot and, as a result, the safety of the process.

The hypothesis has been demonstrated by developing, and empirically assessing the effects of, a novel decision support system framework for avionics dynamic reconfiguration named SaIRA (Safe and Interactive Reconfiguration Architecture) which has been specifically designed in accordance with the hypothesis. In fact, SaIRA generates decision support information that parallels human cognitive strategies and includes (a) explanations that justify each reconfiguration alternative, (b) implications for each alternative and (c) an assessment of the uncertainty embedded in the information processed by the system. The HCI experiments performed provide empirical evidence of the benefits brought by SaIRA to human decision making activities during SCMS dynamic reconfiguration. Several HCI metrics are used to assess the effectiveness of the framework; these metrics include, and are not limited to, all the metrics mentioned in the hypothesis. As a result, all the parts of the hypothesis are demonstrated.

The effectiveness of SaIRA is investigated in the aviation domain only. The improvement of several additional aspects of the operator's decision making activity (including improvement of situation awareness, decrease of frustration and workload and, more generally, improvement of cognitive readiness) suggests that similar benefits could be gained by adapting the decision support strategy proposed in this thesis to other safety-critical decision making problems with comparable characteristics, possibly in different domains.

"The noblest pleasure is the joy of understanding"
Leonardo Da Vinci (April 15, 1452 - May 2, 1519)

Appendices

Appendix A

Experimental Tools

Several algorithms and software applications have been developed in the context of this Ph.D. programme in order to validate the ideas proposed. Figure A.1 describes the human-computer interaction framework specifically devised to perform the experiments presented in Chapter 6 and produce the data required by the experimental design; the figure shows how the tools developed (i.e. ADR-Plugin, SETS and SETS-Analyser) have been integrated with third-party software to perform a complex series of HCI experiments.

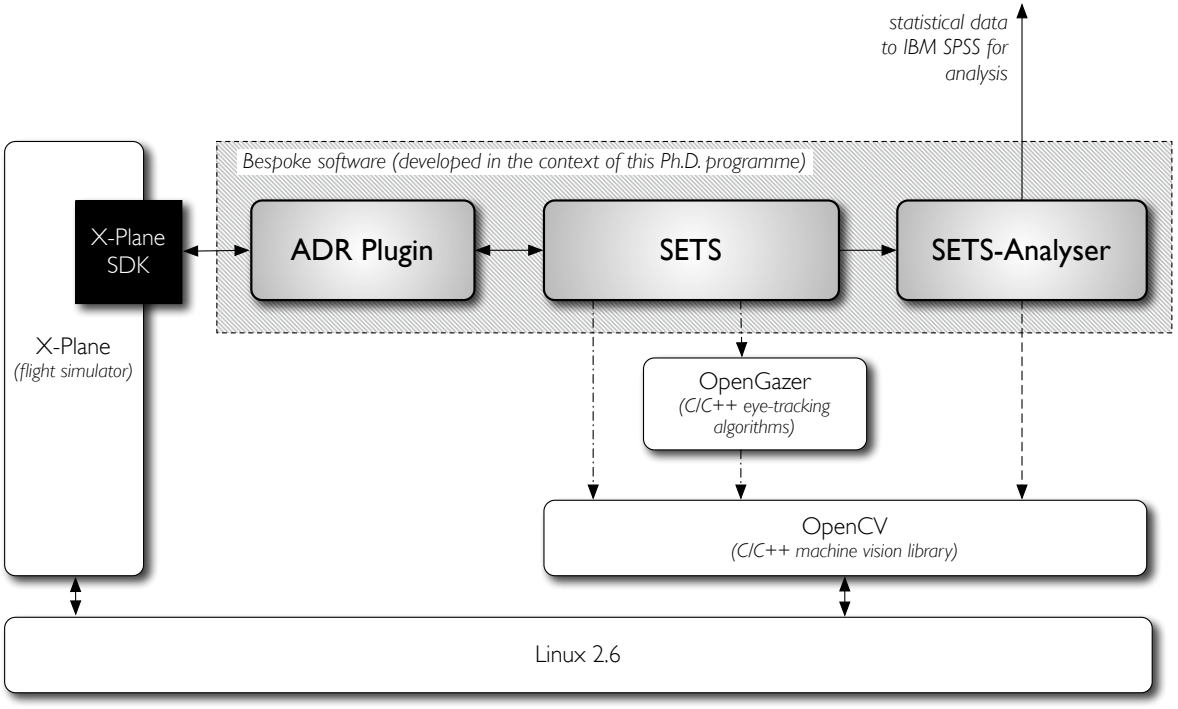


Figure A.1: Architecture of the framework developed for the human-computer interaction experiments described in Chapter 6.

Chapter 5 described the algorithms developed for the generation of decision support information together with an assessment of their performance; these algorithms are implemented in the ADR-Plugin. Chapter 6 provided an overview of the tools developed to perform the experiments

with the pilots but the exposition was purposely kept generic to favour the clarity of description of the experiments design and of the results obtained. In order to support the reproducibility of the experiments, this appendix expands the description of both SETS and SETS-Analyser systems started in Chapter 6.

A.1 SaIRA Eye-Tracking System (SETS)

SETS is designed to produce a streamlined set of data in real-time, integrating eye-movement coordinates localised on the cockpit display with simulation events (e.g. fault firing, change of information displayed on the cockpit, gaze out-of-screen) coming from the flight simulator. The output of SETS is processed by SETS-Analyser, which generates the data in the form required by our analysis. The functioning of SETS is better understood through Figure A.2, which captures the data flow in and out of it.

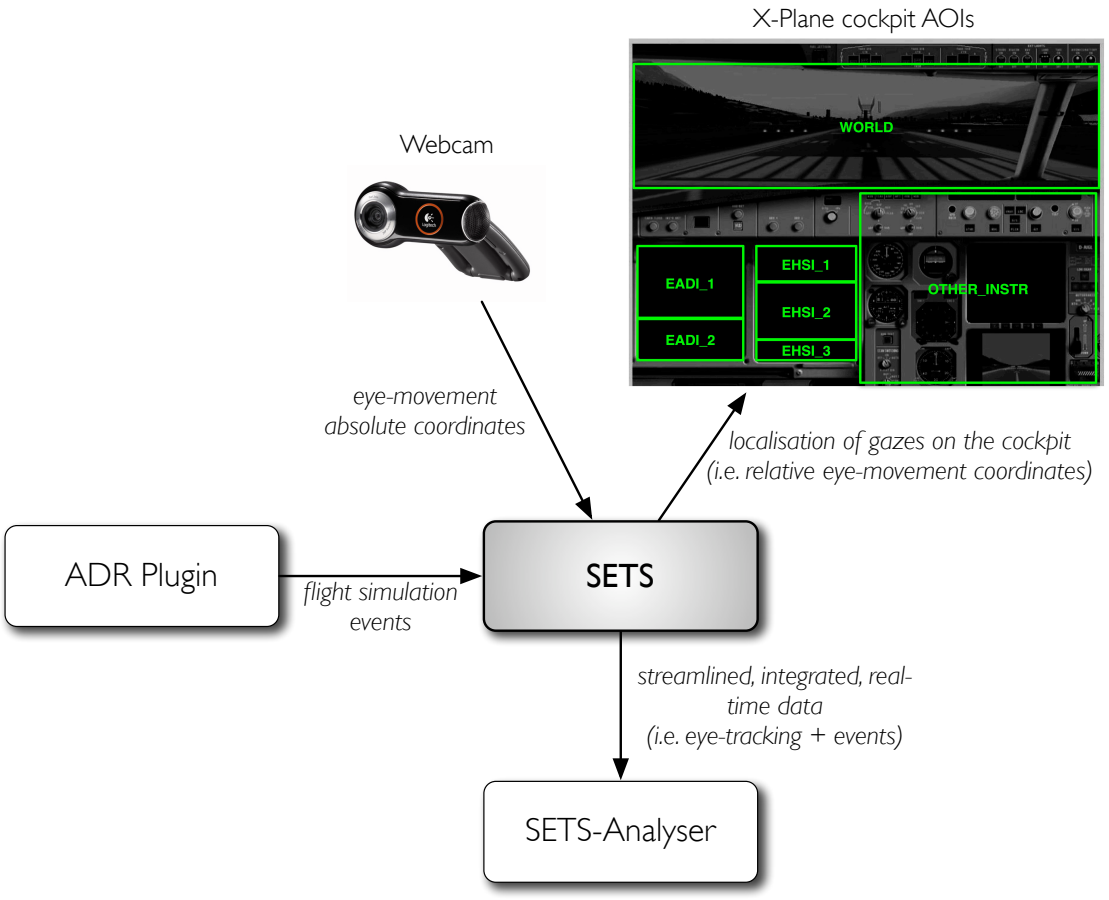


Figure A.2: Data flow in and out of SETS.

The gaze coordinates coming from the camera placed in front of the pilot (see section 6.2.2 for more details on the experiments design) are absolute with respect to the 40-inches display; SETS localises the coordinates on the cockpit, allowing precise characterisation of pilots' visual attention.

The flow of relative gaze coordinates is merged with flight simulation events generated by the ADR-Plugin at a frequency set by Simulation Time Units (STU). All events unfold and are logged in STU; one STU corresponds to 1/60 second.

Before presenting some of the algorithms of SETS, it is worth noting that the software interfaces itself with the X-Plane flight simulator through the ADR-Plugin; the communication logic was written using the API[1] provided by the X-Plane Source Development Kit [Laminar Research Inc. 2009]. Furthermore, some of the functions of SETS are written with the support of the OpenCV [Intel 1999] machine vision library (e.g. image colour adjustment, saturation manipulation, coordinates calculation).

Both SETS and SETS-Analyser are written in C and C++ and are designed to run on Linux 2.6. SETS uses OpenGazer [Zielinski 2009] as eye-tracking engine and extends its functionalities to meet the needs of the SaIRA project. In fact, OpenGazer allows capturing a streamlined series of gazes on the screen but it has no concept of fixation, saccade, and of all the related eye movement features which are required by our experiments design (e.g. mean fixation duration). The need to have access to these eye movement features motivated the development of SETS and SETS-Analyser.

Listing A.1 contains the basic interface provided by the X-Plane SDK which was used to develop the ADR-Plugin. The `XPluginStart()` function, which is called once when the plugin is loaded (it is a sort of "constructor") is the most interesting portion of code in relation to the eye-tracking system because it configures SETS for execution amongst other things. Listing A.2 contains a relevant portion of the `XPluginStart()` implemented for the ADR-Plugin.

Listing A.1: X-Plane SDK 'Hello World' code.

```
#include <stdio.h>

PLUGIN_API int XPluginStart(
                char *    outName,
                char *    outSig,
                char *    outDesc) { }

PLUGIN_API void XPluginStop(void) { }

PLUGIN_API void XPluginDisable(void) { }

PLUGIN_API int XPluginEnable(void) { }

PLUGIN_API void XPluginReceiveMessage(
             XPLMPluginID inFromWho,
             long      inMessage,
             void *       inParam) { }
```

In function `XPluginStart`, after having registered the plugin with the flight simulator engine, the `FindWindowCoordinates()` function localises the position of the cockpit displays on the LCD screen. The AOIs mask is then adjusted to the correct location of the cockpit, as previously

[1] Application Programming Interface

shown in Figure 6.2 (page 150). The fine adjustment of the AOI grid on the cockpit is necessary for SETS to translate absolute into relative gaze coordinates.

Afterwards, the graphics and textures used by SaIRA are loaded in memory (they are kept ready in memory for the instant in which a reconfiguration is triggered in order to avoid execution glitches) and the POSIX thread `commandListenerThread` is launched. The threaded function `commandListenerServlet()` implements an UDP socket which accepts commands from the Control Computer; a typical command triggers a fault at the hand of the operator. As mentioned in Chapter 6, the Control Computer is connected to the ADR-Plugin through Ethernet connection (see Figure 6.1 on page 147).

Listing A.2: A portion of the `XPluginStart` function of the ADR plugin.

```
PLUGIN_API int XPluginStart(char * outName, char * outSig, char * outDesc) {
   strcpy(outName,"IMSDR");
   strcpy(outSig, "xpsdk.experimental.IMSDR");
   strcpy(outDesc, "SaIRA ADR plug-in (author: Giuseppe Montano, Department of Computer
       Science, The University of York).");

   //Find X-Plane window coordinates
   if (FindWindowCoordinates()) {
      printf("IMS-DR: error: impossible to find X-Plane window coordinates.\n");
      return 0;
   }

   //Set panel displays relative coordinates
   dsplPosRel.PFDx = 13;
   dsplPosRel.PFDy = 60;
   dsplPosRel.PFDheight = 245;
   dsplPosRel.PFDwidth = 230;
   dsplPosRel.NDx = 270;
   dsplPosRel.NDy = 65;
   dsplPosRel.NDheight = 250;
   dsplPosRel.NDwidth = 217;
   dsplPosRel.EWDx = 735;
   dsplPosRel.EWDy = 145;
   dsplPosRel.EWDheight = 170;
   dsplPosRel.EWDwidth = 210;

   scroll_pos = XPLMFindDataRef("sim/graphics/misc/current_scroll_pos");

   //Setup fault scheme (images) data
   char *pFileName = "Resources/plugins/IMSDR"/";
   XPLMGetSystemPath(gPluginDataFile);
   strcat(gPluginDataFile, pFileName);
   gExampleGaugePanelDisplayWindow = XPLMCreateWindow(768, 256, 1024, 0, 1,
       ExampleGaugePanelWindowCallback, NULL, NULL, NULL);
   RED = XPLMFindDataRef("sim/graphics/misc/cockpit_light_level_r");
   GREEN = XPLMFindDataRef("sim/graphics/misc/cockpit_light_level_g");
   BLUE = XPLMFindDataRef("sim/graphics/misc/cockpit_light_level_b");
```

```
  // Load the textures and bind them etc.
  LoadTextures();

  //Listen for remote commands
  pthread_create(&commandListenerThread, 0, commandListenerServlet, NULL);

  printf("ADR plugin started\n");
}
```

Before running any experiment, a special command is sent from the Control Computer to the ADR-Plugin to enable SETS. The command initialises the necessary data structures and then enables the SETS eye-tracking service, which is threaded by the `gazeTrackingThreadServlet` reported in Listing A.3.

SETS is interfaced with the OpenGazer eye-tracking engine through another UDP socket. The thread running the `gazeTrackingThreadServlet` function executes a loop that continuously reads from the socket; x and y gaze coordinates are extracted from the data stream reaching the socket.

As shown in Chapter 6, significance analysis is performed on the statistical data generated during the HCI experiments. For significance tests, missing data in time-related series is particularly important and must be handled carefully in order to obtain genuine results. In order to allow correct handling of missing eye-tracking data, after having extracted x and y coordinates, the `gazeTrackingThreadServlet` checks whether the gaze is detected inside the flight simulation area or outside the LCD screen; in the latter case, both coordinates values are set to -1, representing a missing data and enabling IBM SPSS [IBM 1968] to correctly handle it at post-processing time.

Listing A.3: `gazeTrackingThreadServlet`

```
while ((bytes = read(sock, data, 1024)) > 0) {
  data[bytes] = '\0';
  // x-coordinate
  char *nlIndex = strchr(data, '\n');
  xLenght = (nlIndex - data) / sizeof(char) - 2;
  char xString[xLenght];
  for (int i = 0; i < xLenght; i++) {
    xString[i] = data[i + 2];
  }
  xString[xLenght] = '\0';
  //y-coordinate
  char *endIndex = strchr(data, '\0');
  char *yIndex = strchr(data, 'y');
  yLenght = (endIndex - yIndex) / sizeof(char) - 3;
  char yString[yLenght];
  for (int i = 0; i < yLenght; i++) {
    yString[i] = data[xLenght + 5 + i];
  }
  yString[yLenght] = '\0';
  //convert the coordinates to int
  x = atoi(xString);
```

```
  y = atoi(yString);
  /*
   * log the coordinates if the gaze is within the X-Plane window,
   * otherwise set them to -1
   */
  if (x >= windowX && x <= windowX + windowWidth && y >= windowY && y <= windowY +
      windowHeight) {
    x -= windowX;
    y -= windowY;
  } else {
    x = -1;
    y = -1;
  }

  //Check if the user is looking at...
  //... the PFD?
  if (x >= dsplPosAbs.PFDx && x <= dsplPosAbs.PFDx + dsplPosAbs.PFDwidth && y >=
      dsplPosAbs.PFDy && y <= dsplPosAbs.PFDy + dsplPosAbs.PFDheight) {
    if (currentEyesTarget != EYES_ON_PFD) {
       gettimeofday(&time2, NULL);
       delta_sec = time2.tv_sec - time1.tv_sec;
       delta_usec = time2.tv_usec - time1.tv_usec;
       timeDelta = ((delta_sec) * 1000 + delta_usec / 1000.0) + 0.5;
       time1 = time2;
       currentEyesTarget = EYES_ON_PFD;
    }
    ...
  }
  //... the ND?
   } else if ....

   /* check for other locations on the screen */

  //... the world outside
   } else {
   ...
   }

}
```

Whilst the execution loops inside the `gazeTrackingThreadServlet` function, the following actions are performed for each gaze inside the cockpit area:

1. the AOI in which the current gaze is into is identified;

2. if the previous gaze was in the same AOI, no further actions are taken;

3. if the previous gaze was in a different AOI, a ‘change of AOI’ event is recorded: the current time—at millisecond resolution—is logged, allowing calculation of the amount of time spent in the AOI just left by the pilot’s visual attention;

The 'raw' data stream generated by SETS is post-processed by SETS-Analyser, as described in the next section.

A.2 SETS-Analyser

Before discussing the type of processing performed by SETS-Analyser, it is worth clarifying how the eye movement is modelled in this software.

Eye movement is a combination of two main behaviours: first, *fixations*, where the eye is relatively still (for 150-200 ms); second, *saccades*, where the eye moves rapidly between fixations.

SETS has a sampling rate of 60Hz. In light of this constraint, we modelled a fixation as a series of gazes in a tiny squared area of the screen. More specifically, SETS-Analyser registers a fixation when at least 5 consecutive gazes detected by SETS fall within a squared area of 40 pixels. Figure A.3 provides a graphical representation of a SETS-Analyser fixation, defined as a cluster of raw gazes falling within the area in question.

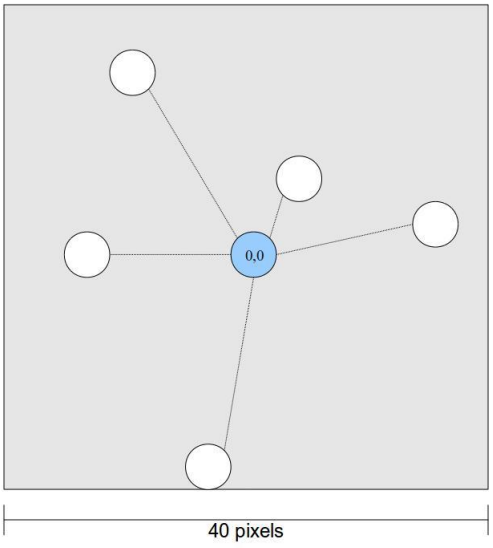


Figure A.3: Fixation cluster as defined in SETS-Analyser.

SETS-Analyser generates the following data from the raw gazes coordinates recorded in real-time during the simulation: number of fixations, number of saccades, raw gazes in each AOI, gazes transition matrix, fixations transition matrix, sum of fixations time for each AOI, mean fixation duration, mean fixation duration for each AOI, rate of fixation, on-target fixations, mean saccades length, mean raw gazes per AOI, saccades rate, number of backtracks, number of backtracks per AOI, fixations/saccades rate, scanpath map (image), backtracks map (image), hotspot map (image). For reasons of brevity, only some of the functions that produce this data are reported hereinafter; the algorithms covered, however, allow exhibiting the data processing approach adopted in SETS-Analyser.

The first step for further processing is the calculation of fixations and saccades from the raw streamlined data set. The class `FixSacAnalyser` of SETS-Analyser provides all the functionality to handle fixations and saccades. Listing A.4 shows one of its functions, `CalculateFixAndSac`, which calculates the two basic eye movement features in question. This function is called before proceeding to any other type of processing.

In brief, an initial (x, y) coordinates pair of raw gaze coordinates (from the `rawGazesVect` vector) is selected; then, the following pairs of coordinates in the vector are serially scanned and a set of raw gazes are grouped into a single fixation—and pushed into the fixation vector `fixations`—if all the raw gazes in the set fall within a squared area of pre-defined side length `FIX_AREA_SIDE`. A very small fixed number of gazes outside the area is allowed within the set of gazes that make up a fixation; this number is given by the constant `MAX_JUMPS_PER_FIX`. The rationale for `MAX_JUMPS_PER_FIX` is to compensate for camera detection errors that we found in the preliminary validation of the system (i.e. before starting the series of experiments); we found out that, from time to time, a single 'jump' lasting less than 50 milliseconds was added to the streamlined data set. The jump was evidently inconsistent with the visual path followed by the pilot given its duration and length.

Besides the `FIX_AREA_SIDE`, another constraint that a set of consecutive raw gazes coordinates must satisfy to qualify for a fixation is that their count must be higher that the value of the constant `MIN_GAZES_PER_FIX`. This constraint allows for the cancellation of noise from the eye movement data.

If all the constraints are not satisfied by the next gaze coordinates pair in the series, the following actions are performed: (a) the coordinates pairs scanned so far, except the last one, are stored in a single fixation object; (b) a new saccade is defined and its data structure is created; (c) the new saccade is added to the vector `saccades`.

Listing A.4: Calculation of fixations and saccades

```
/*
 * Fixations and saccades calculation
 *
 * A single fixation contains gazes inside a square of FIX_AREA_SIDE side
 * lenght plus a maximum of MAX_JUMPS_PER_FIX gazes outside the relevant
 * area. This is to overcome cameras acquisition noise.
 */
int FixSacAnalyser::CalculateFixAndSac(DataStore& dS) {
  int x1, y1, x2, y2;
  unsigned index1 = 0;
  unsigned index2 = 0;
  int in = 0;
  int out = 0;
  Fixation fix;
  int totalGazesPerFix = 0;
  int prevIndex2;
  Saccade sac;
  while (index1 < dS.rawGazesVect.size()) {
    x1 = ((CvPoint) dS.rawGazesVect[index1]).x;
```

```
      y1 = ((CvPoint) dS.rawGazesVect[index1]).y;
      index2 = index1 + 1;
      while (index2 < dS.rawGazesVect.size()) {
        x2 = ((CvPoint) dS.rawGazesVect[index2]).x;
        y2 = ((CvPoint) dS.rawGazesVect[index2]).y;
        if (abs(x2 - x1) <= (FIX_AREA_SIDE / 2) && abs(y2 - y1) <= (FIX_AREA_SIDE / 2)) {
          out = 0;
          in++;
        } else {
          out++;
        }
        totalGazesPerFix++;
        index2++;
        if (out > MAX_JUMPS_PER_FIX) {
          totalGazesPerFix -= out;
          index2 -= (out);
          x2 = ((CvPoint) dS.rawGazesVect[index2]).x;
          y2 = ((CvPoint) dS.rawGazesVect[index2]).y;
          break;
        }
      }
      if (in >= MIN_GAZES_PER_FIX) {
        fix.x = x1;
        fix.y = y1;
        fix.nrOfGazes = totalGazesPerFix;
        fixations.push_back(fix);
        if (fixations.size() > 1) { //register saccade
          sac.x = fix.x;
          sac.y = fix.y;
          sac.nrOfGazes = index1 - prevIndex2;
          saccades.push_back(sac);
        }
        prevIndex2 = index2;
      }
      totalGazesPerFix = 0;
      index1 = index2;
      in = 0;
      out = 0;
    }
    fixationsCalculated = 1;
    return EXIT_SUCCESS;
}
```

Listing A.5 processes the `fixations` vector populated in the way shown in Listing A.4 to calculate the total time spent by the pilot fixating each AOI. For its calculations, the functions assumes that the camera has a sampling rate defined by `SCAN_FREQ`. For reasons of brevity, only three AOI are defined in Listing A.5.

Listing A.5: Calculation of fixation time within an AOI

```
/*
```

```
 * This function makes the assumption that the cameras take snapshots
 * at the frequency specified by SCAN_FREQ. This value is used to calculate
 * total time of fixations (nrRawGazes*capturePeriod)
 */
int FixSacAnalyser::CalculateAOIFixationsTimes(DataStore& dS) {
  if (!fixationsCalculated) {
    return EXIT_FAILURE;
  }

  //load fixations of each AOI into the relative vector
  Fixation fix;
  for (unsigned i = 0; i < fixations.size(); i++) {
    fix = (Fixation) fixations[i];
    if (fix.x >= dS.pA_aoi1.x && fix.x <= dS.pB_aoi1.x && fix.y >= dS.pA_aoi1.y && fix.y
        <= dS.pB_aoi1.y) {
      fixationsAOI1.push_back(fix);
      all_nrOfFixationsAOI1++;
    } else if (fix.x >= dS.pA_aoi2.x && fix.x <= dS.pB_aoi2.x && fix.y >= dS.pA_aoi2.y &&
        fix.y <= dS.pB_aoi2.y) {
      fixationsAOI2.push_back(fix);
      all_nrOfFixationsAOI2++;
    } else if (fix.x >= dS.pA_aoi3.x && fix.x <= dS.pB_aoi3.x && fix.y >= dS.pA_aoi3.y &&
        fix.y <= dS.pB_aoi3.y) {
      fixationsAOI3.push_back(fix);
      all_nrOfFixationsAOI3++;
    }
  }

  float fixTimeAOI1 = 0;
  for (unsigned i = 0; i < fixationsAOI1.size(); i++) {
    fixTimeAOI1 = fixTimeAOI1 + fixationsAOI1[i].nrOfGazes;
  }
  float fixTimeAOI2 = 0;
  for (unsigned i = 0; i < fixationsAOI2.size(); i++) {
    fixTimeAOI2 = fixTimeAOI2 + fixationsAOI2[i].nrOfGazes;
  }
  float fixTimeAOI3 = 0;
  for (unsigned i = 0; i < fixationsAOI3.size(); i++) {
    fixTimeAOI3 = fixTimeAOI3 + fixationsAOI3[i].nrOfGazes;
  }

  fixTimesPerAOI.push_back(fixTimeAOI1 / SCAN_FREQ);
  fixTimesPerAOI.push_back(fixTimeAOI2 / SCAN_FREQ);
  fixTimesPerAOI.push_back(fixTimeAOI3 / SCAN_FREQ);

  return EXIT_SUCCESS;
}
```

The function in Listing A.6, as its name suggests, is a simple routine to calculate the means fixation duration of the data in vector `fixations`. The vector `fixations` contains the set of

fixations extracted from the raw gazes but they are not necessarily all the fixations of a flight simulation; in fact the content of the vector can be tuned by setting the portion of the simulation to be analysed, as explained later in this section.

Listing A.6: Calculation of mean fixation durations

```
int FixSacAnalyser::CalculateMeanFixDurations(DataStore& dS) {
  if (!fixationsCalculated) {
    return EXIT_FAILURE;
  }

  float totalGazes;
  for (unsigned i = 0; i < fixations.size(); i++) {
    totalGazes += fixations[i].nrOfGazes;
  }
  meanFixTime = totalGazes / SCAN_FREQ / fixations.size();
  return EXIT_SUCCESS;
}
```

The way the mean saccadic amplitude is calculated is shown in Listing A.7 whilst Listing A.8 contains the algorithm to calculate backtracks.

Listing A.7: Calculation of mean saccadic amplitude

```
int FixSacAnalyser::CalculateMeanSacAmplitude(void) {
  float x = 0;
  float y = 0;
  for (unsigned i = 0; i < fixations.size(); i++) {
    x += fixations[i].x;
    y += fixations[i].y;
  }
  x = x / fixations.size();
  y = y / fixations.size();
  meanSacAmplitude = sqrt(pow(x, 2) + pow(y, 2));
  return EXIT_SUCCESS;
}
```

Listing A.8: Calculation of backtracks

```
Backtrack b;
for (unsigned i = 0; i < fsa.fixations.size() - 2; i++) {
  if ((abs(fsa.fixations[i].x - fsa.fixations[i + 2].x) <= BACKTRACK_AREA || abs(fsa.
      fixations[i].y - fsa.fixations[i + 2].y) <= BACKTRACK_AREA) && (abs(fsa.fixations[i
      ].x - fsa.fixations[i + 1].x) > BACKTRACK_AREA || abs(fsa.fixations[i].y - fsa.
      fixations[i + 1].y) > BACKTRACK_AREA)) {
    b.p1 = cvPoint(fsa.fixations[i].x, fsa.fixations[i].y);
    b.p2 = cvPoint(fsa.fixations[i + 1].x, fsa.fixations[i + 1].y);
    backtracks.push_back(b);
  }
}
```

For each test, SETS generates a file named `gazes_xxx.txt`—where `xxx` is the identifier of each pilot—which contains the streamlined data set including eye-tracking coordinates and simulation events. The file `gazes_xxx.txt` is the input for SETS-Analyser.

From the input, SETS-Analyser generates the following output files (the definitions of the eye movement features listed hereinafter were given in section 6.2.2):

a. `results_xxx.txt`: (unique for all participants) It contains the basic descriptive statistics of the dependent variables; its purpose is providing the experiment controller with an overview of the results for a quick assessment of the validity of the simulation, e.g. percentage of 'out-of-screen' gaze detected.

b. `backtracks_between_AOIs_xxx.csv`: (unique for all participants) It contains a table listing the backtracks amongst the pre-defined AOIs.

c. `fix_rates.csv`: (unique for all participants) It contains the mean fixation rates.

d. `mean_fix_durations.csv`: (unique for all participants) It contains the mean fixation durations.

e. `mean_fix_per_AOI.csv`: (unique for all participants) It contains the mean number of fixations for each AOI.

f. `mean_on_target_fix_per_AOI.csv`: (unique for all participants) It contains the mean on-target fixations for each AOI.

g. `mean_raw_gazes_per_AOI.csv`: (unique for all participants) It contains the mean raw gazes for each AOI.

h. `nr_backtracks.csv`: (unique for all participants) It contains the number of backtracks of each participant.

i. `hotspotMap_xxx.jpg`: a typical hotspot map produced by SETS has already been shown in Figure 6.5 on page 153. The file is unique for all participants. It is a JPG image representing the areas of the cockpit that receive more visual attention. The visual attention is represented by means of a temperature scale (shifting from green to red with increasing levels of attention).

j. `scanpath_xxx.jpg`: (see Figure A.4(a)) the file is unique for all participants. It is a JPG image representing the pilot's scanpath on the cockpit.

k. `backtracks_xxx.jpg`: (see Figure A.4(b)) the file is unique for all participants. It is a JPG image representing the backtracks in the forms of green segments on the cockpit. The extremes of the segments are the two points between which the visual attention is divided.

The numerical eye-tracking data is stored in .csv files (Comma-Separated Values), which are compatible with Microsoft Excel and IBM SPSS and can be easily used for further analysis.

SETS-Analyser is launched from the Linux command line with the following mutually exclusive arguments sets:

(a) Scanpath image generated by SETS-Analyser.

(b) Backtracks image generated by SETS-Analyser.

Figure A.4: Scanpath and backtracks images generated with SETS-Analyser on a small portion of the raw gazes data collected during a flight simulation.

- No arguments: all the raw gazes contained in the `gazes_xxx.txt` file of each participant are processed;
- `-timeRange <startTime> <stopTime>`: the user specifies a range of time starting at `<startTime>` and finishing at `<stopTime>`, included in the observation time. Only the raw gazes recorded during the specified range of time are processed.
- `-betweenTags <startTAG> <stopTAG>`: ADR-Plugin inserts tags for events that happen during the simulation in the list of raw gazes recorded in real-time, e.g. `FAULT_1` to specify that 'fault-1' has been triggered. With the `-betweenTags` option the user can instruct SETS-Analyser to process only the raw gazes between the specified tags.

The `-timeRange` and `-betweenTags` arguments can be used to tune the content of the `fixations` and `saccades` vectors, as mentioned earlier in this appendix.

Appendix B

Generating Recommendations in SaIRA

B.1 Generating Recommendations Algorithmically

Section 5.4.3 stated that, in SaIRA, the objective of a decision support message is to help the pilot answering the question "why should I apply `Configuration A` instead of `Configuration B`?" Accepting one configuration instead of another corresponds to accepting one repair action instead of another (e.g. sacrifice the Elevator Feel System instead of the Waypoint Generator System).

The constraint-based reconfiguration system must go through the following three steps in order to provide support to the pilot:

1. Generate a number of *configurations* using the current active constraints;
2. Generate a set of *repair actions* for the inconsistencies;
3. Generate *recommendations* for the repair actions.

The first step has already been discussed in detail through an example in Section 5.4. The second and third steps are dealt with in this appendix. The problem being examined here can be described as follows:

> Let $\Delta = (X, D, Z, H)$ be the dynamic constraint network associated with the ADR problem. Initially, the partial assignment ρ of the variables in X is empty: $\rho = \emptyset$. At each iteration of the enumeration process, for each unassigned variable $x_i \in X \setminus \text{dom}(\rho)$, a *valid domain* $D_i^\rho \subseteq D_i$ is computed. Then, a value $v_i \in D_i^\rho$ is selected for the variable x_i.
>
> After a number of iterations, the problem becomes over-constrained and no solution is available given the current set of constraints.
>
> Through repair actions, SaIRA must be able to generate one or two recommendations for the solution of Δ which enable the pilot to make an informed decision.

Current research in the recommendation technology domain focuses mainly on product and configuration recommendation in the retail industry field, e.g. book selling, camera configuration.

To the best of our knowledge there are no studies on the application of constraint-based recommendation technology to safety-critical decisions in the aerospace arena. Indeed, as discussed in more detail in the following sections, this domain has specific requirements in terms of recommendation features that limit the applicability of standard methods and lead to the development of a new, bespoke, constraint-based explanation algorithm, which is introduced and empirically evaluated afterwards.

First, state-of-the-art constraint-based recommendation technology is reviewed. Then two novel algorithms for automated generation of explanations and implications are presented and the performance of the core algorithm is empirically assessed. Subsequently, a method to translate the computer-encoded recommendations in natural language messages for the pilots is discussed. The technology expounded in this appendix was used to produce the information displayed in the SaIRA user interface described in Section 5.7.2.

B.1.1 Recommendations with Explanation-based Constraint Programming

The solution of CSP with classic algorithms, based on chronological backtracking, has been shown to have several disadvantages, including the well known *trashing* [Gaschnig 1987] (the search always fails for the same "reasons" because of the impracticability of remembering all the past failure conditions).

One of the solutions proposed in the literature is **Explanation-based Constraint Programming (ECP)** [Jussien 2001]; the theory behind ECP is briefly discussed here because the algorithm proposed later in this chapter is inspired by this technology. In brief, explanations contain information that justifies the decisions taken by the solver during the search for a solution (e.g. retracting a constraint). The information contained in ECP explanations is made up of constraints and choices made by the solver. In ECP, this information is used both to guide the search for solutions and as a baseline for decision support information for the user.

Consider a CSP defined by Jussien [2003] as follows:

Definition B.1.1 *A Constraint Satisfaction Problem (CSP) is defined by:*

- *a finite set V of variables;*
- *a finite set C of constraints;*
- *a function* $var : C \rightarrow \mathcal{P}(V)$
- *a family* $(D_x)_{x \in V}$ *(the domains)*
- *a family* $(T_c)_{c \in C}$ *(the constraints semantic)*

Note that, even if framed differently, the previous definition is equivalent to Definition 5.3.1. It follows that:

Definition B.1.2 *A* ***solution*** *for a CSP* $(V, C, var, (D_x)_{x \in V}, (T_c)_{c \in C})$ *is a tuple s on V such that* $\forall c \in C, s|_{var(c)} \in T_c$.

Now consider the case of a CSP whose current state is contradictory (i.e. over-constrained). In this situations it is possible to isolate a **conflict set** (also known as a nogood or contradiction explanation):

Definition B.1.3 *A **conflict set** or **nogood** is a subset of the current constraint system of the problem that left alone leads to a contradiction (no feasible solution can contain a conflict set).*

A conflict set can be divided in two parts: a subset of the original set of constraints ($C' \subset C$ in Equation B.1) and a subset of the decision constraints introduced so far ($dc_1, \ldots, dc_k$).

$$\neg(C' \wedge d_1 \wedge \ldots \wedge d_k) \tag{B.1}$$

Equation B.1 can be rewritten as follows:

$$C' \wedge \left(\bigwedge_{i \in [1..k] \setminus j} dc_i \right) \rightarrow dc_j \tag{B.2}$$

Having considered $dc_j : v = a$ in Equation B.2, Jussien refers to the left side of the implication as the **eliminating explanation** (or explanation for short) because it justifies the removal of a value a from the domain $d(v)$ of the variable v. The explanation is noted as $\texttt{expl}(v \neq a)$.

A solution to a CSP is given by the following:

$$\textbf{sol}(V, (C' \wedge d_1 \wedge \ldots \wedge d_n)) = \emptyset \tag{B.3}$$

Less formally, Rochart, Jussien and Laburthe [2003] describe an explanation as follows:

Definition B.1.4 *An **explanation** of an inference (X) consists of a subset of the original constraints ($C' \subset C$) and a set of instantiation constraints (i.e. choices made during the search: $d_1, d_2, \ldots d_k$) such that $C' \wedge d_1 \wedge \ldots \wedge d_n \Rightarrow X$. $C' \wedge d_1 \wedge \ldots \wedge d_n$ justifies the inference and is called explanation.*

Jussien and colleagues add that an explanation-set e_1 is said to be more precise than explanation-set e_2 if and only if $e_1 \subset e_2$. The more precise an explanation, the more useful it is.

In Jussien's terms, if (during the propagation of the constraints defined for a CSP the domain) the domain $d(v)$ of variable v is emptied, the problem is over-constrained, i.e. no value is available for v. The explanations generation mechanism of ECP can be used to understand the reason why $d(v)$ has been emptied, and this information can be used to implement a repair action (i.e. choose which of the constraints propagated so far should be retracted). For $d(v) = \emptyset$, a **contradiction explanation** can be computed:

$$\neg\left(\bigwedge_{a \in d(v)} \texttt{expl}(v \neq a) \right) \tag{B.4}$$

Usually several explanations exist for every over-constrained situation and more generally for any inference performed by the solver. Recording all the explanations during the enumeration process would enable making optimal repair decisions at any time during the search. However,

this would make the space complexity non tractable for most complex problems. In ECP *one* explanation for each inference made by the solver is retained during the enumeration process.

Jussien [2001] introduces PALM (Propagate and Learn with Move), a CSP solver that implements ECP ideas which is *capable of explaining its behaviour* and *handling dynamic constraints additions and removals*. PALM is developed on top of CHOCO [Laburthe 2000]. As already mentioned, CHOCO has been extensively used in this thesis to implement the majority of CP technology.

The ideas behind ECP are now put into the context of SaIRA. In the example of the simplified IMA reconfiguration problem described in Section 5.4.1, a set of six flight control applications must run in the system; the total bus bandwidth available in nominal conditions is 100 Mb/s but, as a result of a fault, the bandwidth is drastically reduced to 15 Mb/s. The bandwidth requirements of the six applications are given in Table B.1.

	Bandwidth Required
App 1	$b_1 = 3$ Mb/s
App 2	$b_2 = 1.5$ Mb/s
App 3	$b_3 = 2.5$ Mb/s
App 4	$b_4 = 2$ Mb/s
App 5	$b_5 = 3$ Mb/s
App 6	$b_6 = 14$ Mb/s

Table B.1: Bandwidth requirements for each application.

The total bandwidth consumption of the set of active applications is calculated as follows:

$$B_t = \sum_{i=1}^{6} x_i \cdot b_i \leq 15 \text{ Mb/s} \tag{B.5}$$

The process of enumerating the applications activation constraints is described in Table B.2. A generic activation constraint ρ_i, implemented as a boolean variable $x \in \{0, 1\}$, indicates whether the i-th application is requested to execute or not. When ρ_6 is propagated the problem becomes over-constrained (the total bandwidth required exceeds the maximum bandwidth available by 11 Mb/s).

♯	**Constraint**	**Bandwidth Consumed**	**Argument/Conflict**
0	$\rho_0 : B_t = \sum_{i=1}^{6} x_i \cdot b_i \leq 15$ Mb/s	—	$\{ \rho_0 \}$
1	$\rho_1 : x_1 = 1$	B_t=3 Mb	$\{ \rho_0, \rho_1 \}$
2	$\rho_2 : x_2 = 1$	B_t=4.5 Mb	$\{ \rho_0, \rho_1, \rho_2 \}$
3	$\rho_3 : x_3 = 1$	B_t=7 Mb	$\{ \rho_0, \rho_1, \rho_2, \rho_3 \}$
4	$\rho_4 : x_4 = 1$	B_t=9 Mb	$\{ \rho_0, \rho_1, \rho_2, \rho_3, \rho_4 \}$
5	$\rho_5 : x_5 = 1$	B_t=12 Mb	$\{ \rho_0, \rho_1, \rho_2, \rho_3, \rho_4, \rho_5 \}$
6	$\rho_6 : x_6 = 1$	B_t=26 Mb	$\{ \rho_0, \rho_1, \rho_2, \rho_3, \rho_4, \rho_5, \rho_6 \}$
		FAIL	$\{ \rho_0, \rho_1, \rho_2, \rho_3, \rho_4, \rho_5, \rho_6 \}$ $\downarrow$ $\{\rho_0, \rho_5, \rho_6\}$

Table B.2: Computing of a complete explanation during the enumeration process.

Several explanations for the conflict found at the bottom of Table B.2 could be provided to

the pilot. The full explanation $\{\rho_0, \rho_1, \rho_2, \rho_3, \rho_4, \rho_5, \rho_6\}$, containing all the decisions taken by the solver before the conflict, is not helpful as a decision making support. A minimal explanation, which explains in a capsule why a contradiction was found, is more interesting. As previously mentioned, by default PALM keeps only one explanation per inference made by the constraints involved in the contradiction. In this example, the explanation for the over-constrained situation (the domain of x_6 was emptied) is $\{\rho_0, \rho_5, \rho_6\}$, which could be translated in natural language as "not enough bandwidth for both Applications 5 and 6". Note that there are also other valid explanations (e.g. $\{\rho_0, \rho_1, \rho_6\}$, $\{\rho_0, \rho_2, \rho_6\}$) even though they are not recorded by the eCP solver.

The explanation generated implicitly suggests the actions to be taken in cases where the pilot prefers Application 6 to be active in the next configuration, (a) either the maximum bandwidth is increased (constraint $\{B_t\}$ is retracted) or (b) Application 1 is deactivated (constraint $\{\rho_1\}$ is retracted). Whilst the former option is physically impossible to implement (in fact, in a more realistic example, $\{B_t\}$ would be implemented as a static/hard constraint), the pilot can choose to have Application 6 running instead of Application 1.

A family of eCP-based algorithms have been developed to automatically repair over-constrained CSP: the `decision-repair` algorithms [Jussien and Lhomme 2002].

The approach adopted in PALM to generate explanations for over-constrained problems is particularly interesting in the content of ADR because this type of information explains *why* the system reaches one conclusion instead of another. However, apart from the problem of translating the computer-encoded explanations into natural language—which is discussed later in this Chapter—a few relevant drawbacks of the ECP explanations require attention.

The first limitation concerns the *minimality* of the explanations, which is not guaranteed. Jussien [2003] reports that "computed explanations therefore cannot be guaranteed to be minimal (although experiments show that they remain quite precise) nor exhaustive (other explanations may exist)". In the example given the explanation was minimal because of the simplicity of the problem; larger, over-constrained problems have several explanations for each inference, characterised by different sizes.

The concision of the decision support information is particularly important during ADR because of the pilot operating conditions (e.g. time pressure, stress) and because of the limits of space of the cockpit displays. A statistical investigation of the size of explanations generated with PALM with different problem configurations and sizes is beyond the scope of this research. This thesis assumes that explanations generated with PALM remain concise also in large CSP, as claimed and empirically verified by Jussien.

Another aspect to take into consideration is the *elicitation of preferences* for explanations. Given that for each inference that are several explanations, the question of how to select the explanation that is most helpful for the pilot in each situation becomes an important one. This topic is elaborated below, when three state-of-the-art explanation algorithms are compared and a novel algorithm, developed by the author, is proposed.

A further aspect of the type of explanations examined here is that they provide information about *why* the system came to certain conclusions but no information about the implications of each repair action are generated. ECP-based explanations reveal why the solver shaped two con-

figurations in the way it did, not why one configuration should be chosen instead of another with respect to an objective function. O Sullivan et al. [2005] comment on this pointing out that existing work on explanation generation for the solution of interactive CSP is focused on "blaming" the inconsistent set of constraints being propagated instead of explaining to the user what to do to repair the current situation.

This last aspect of ECP-based explanations is particularly important to enable the type of interaction that is required in SaIRA. Here we speculate (but verify empirically in Chapter 6) that standard PALM explanations are effective to help the pilot to spot any potential wrong inferences of the system, but they should be enriched with the implications of each decision alternative, in order to provide more complete/effective decision support to the pilot.

Some of the issues mentioned in this section, but not all, are addressed by two widely referenced algorithms for constraint-based explanation, QUICKXPLAIN and FASTXPLAIN, which are very briefly introduced in the next two sections.

B.1.2 Recommendations with QUICKXPLAIN

The QUICKXPLAIN algorithm [Junker 2001] is the industrial standard *de facto* in terms of constraint-based explanation systems. It is used as the explanation component of the Configurator tool by ILOG [ILOG 2010].

The problem of generating minimal and helpful explanations is addressed by a preference-controlled algorithm that successively adds the preferred constraints until a contradiction is found, in which case the algorithm backtracks and removes least preferred constraints if this avoids the contradiction.

The author uses the notation $\rho_i \prec \rho_j$ to indicate that explanation ρ_i is preferred to ρ_j. He divides the overall set of constraints of a CSP in two subsets: $\mathcal{B}$ is the *background*, containing constraints that cannot be relaxed (this corresponds to the static/hard constraints introduced earlier in this chapter) and C is the set of constraints that can be relaxed. A relaxation problem is then defined as follows:

Definition B.1.5 *A subset R of C is a* relaxation *of a problem $\mathcal{P} := (\mathcal{B}, C)$ $\iff$ $\mathcal{B} \cup \mathcal{R}$ has a solution.*

It follows that:

Definition B.1.6 *A subset C' of C is a* conflict *of a problem $\mathcal{P} := (\mathcal{B}, C)$ $\iff$ $\mathcal{B} \cup C'$ has no solution.*

Exploiting the concept of *lexicographic relaxation* [Brewka 1989], Junker is able to define **preferred relaxations** and **preferred conflicts** of an over-constrained CSP. Adding a constraint to a relaxation corresponds to the retraction of a constraint from an explanation.

Conflicts and relaxations are calculated by following the constructive definitions. The author proposes two versions of the QUICKXPLAIN algorithm, a basic version and an optimised version; the latter, which achieves better performance by adopting a divide-and-conquer approach, is reported

Algorithm B.1 QuickXPlain($\mathcal{B}, C, \prec$)

```
1: if isConsistent(B ∪ C) then
2:    return 'no conflict'
3: else
4:    if C = ∅ then
5:       return ∅
6:    else
7:       return QuickXPlain'(B, B, C, ≺)
8:    end if
9: end if
```

Algorithm B.2 QuickXPlain'($\mathcal{B}, \Delta, C, \prec$)

```
1: if Δ ≠ ∅ and not isConsistent(B) then
2:    return ∅
3: end if
4: if C = {α} then
5:    return {α}
6: end if
Require: α_1, ..., α_n be an enumeration of C that respects ≺
Require: k be split(n) where 1 ≤ k < n
7: C_1 := {α_1, ..., α_k} and C_2 := {α_{k+1}, ..., α_n}
8: Δ_2 := QuickXPlain'(B ∪ C_1, C_1, C_2, ≺)
9: Δ_1 := QuickXPlain'(B ∪ Δ_2, Δ_2, C_1, ≺)
10: return Δ_1 ∪ Δ_2
```

in Algorithms B.1 and B.2. The algorithm is not discussed in details here for the sake of brevity, but complete information is provided by Junker [2001].

An interesting aspect of QuickXPlain for the ADR recommendation problem is that it always terminates; it returns with no explanations if $\mathcal{B} \cup C$ has a solution, otherwise it provides a preferred explanation of $(\mathcal{B}, C, \prec)$.

One notorious limit of QuickXPlain is computational time. QuickXPlain needs $O(2k \cdot log(n/k + 2k)$ consistency checks (worst case) to compute a minimal conflict set of size k out of n constraints [Felfernig et al. 2009]. The computational requirements of QuickXPlain are the rationale for the development of FastXPlain.

B.1.3 Recommendations with FastXPlain

Schubert et al. [2010] introduce the FastXPlain algorithm for the calculation of minimal conflict sets. One feature that distinguishes FastXPlain from other constraint-based explanation methods is that it handles the recommendation task as a new CSP, using a table representation of constraints (which are 'requirements' in SaIRA) and items (which are 'configurations' in SaIRA).

Let the ADR CSP consist of a set of constraints $C = \{c_1, c_2, \ldots, c_n\}$ and a pool of applicable configurations $P = \{p_1, p_2, \ldots, p_n\}$. Each configuration $p_i \in P$ has a set of possible values D_i from the domain. When there is an applicable solution, all the constraints in C are satisfied. For over-constrained problems, the conflict set $CS \subset C$ is defined such that none of the configurations in P

is satisfied. It follows that CS is minimal if there exists no conflict set CS' such that $CS' \subset CS$.

The algorithm was originally designed for product configuration (e.g. selection of a mobile phone from a set on the basis of their characteristics). Hence, it assumes that a set of 'products' is provided as an input to the algorithm. This can be achieved in SaIRA by generating a set of optimal configurations which do not take into account the unexpected event that has affected the system making the CSP over-constrained. *Consequently*, repair actions are calculated for each optimal configuration and minimal conflict sets are generated by means of FastXPlain, which in turn are used to generate decision support information for the pilot.

The algorithm is reported in Algorithm B.3; a complete discussion is provided by Schubert et al. [2010].

Algorithm B.3 FastXPlain(*root*, *p*)

Require: p - table of constraints and applicable configurations
Require: root - the root node of the resulting tree
Require: MSC - set of all minimal conflicts sets
1: $d \leftarrow$ *getMinCardinalityDiagnosis(p)*
2: **for** all *constraints c from d* **do**
3: $p' \leftarrow reduce(c_i, p')$
4: $child \leftarrow ok$
5: **if** $p' = \emptyset$ **then**
6: $child \leftarrow ok$
7: **end if**
8: **if** $path(child) \notin MSC$ **then**
9: $MSC \leftarrow path(child)$
10: **return**
11: **end if**
12: **if** $\exists\, cs \in CS : CS \subseteq MCS \subseteq path(child)$ **then**
13: $child \leftarrow closed$
14: **end if**
15: **if** $child \neq closed$ **then**
16: FastXPlain(child, p')
17: **end if**
18: **end for**

The algorithm is designed to outperform QuickXPlain. However it does not provide any means to specify preferred explanations and implications.

B.1.4 Qualitative Comparison of Constraint-Based Recommendations Methods

Table B.3 summarises the features of the three state-of-the-art approaches which are more significant for the applicability of recommendation problems of the type of SaIRA. The three algorithms in question have been selected because they define three different approach to the same problem.

eCP does not guarantee minimal explanations, but tools have been developed to (a) translate computer-encoded explanations into user-friendly information, and (b) use heuristics to specify preferences on preferred explanations and preferred relaxations for over-constrained situations [Jussien 2003].

	Minimal Explanations	Preferred Explanations	User-Friendly Explanations
ECP	-	X	X
QUICKXPLAIN	X	X	-
FASTXPLAIN	X	-	-

Table B.3: Comparison of the three state-of-the-art approaches to constraint-based autonomous generation of explanations for over-constrained CSP.

QUICKXPLAIN generates minimal explanations and allows specification of the preferred explanations, by setting up a lexicographic ordering of the constraints. Although this approach to specify preferences amongst explanations is particularly neat, establishing a lexicographic ordering amongst the constraints becomes particularly challenging as the number of constraints grows, as we found out in our preliminary investigations. A typical ADR problem has a large number of inter-dependent constraints and variables; the complexity of ordering the constraints effectively should not be underestimated. To the best of our knowledge, no bespoke means are provided to translate computer-encoded information generated by this algorithm into natural language. These motivations make QUICKXPLAIN difficult to apply to SaIRA.

FASTXPLAIN generates minimal explanations and in an experimental study performed by Schubert et al. [2010], it proved to perform better than QUICKXPLAIN in different combination of (a) number of constraints, (b) number of variables, and (c) explanations to be generated. Its drawbacks are the impossibility of specifying preferred explanations and the unavailability of tools to produce high-level explanations. The possibility to specify preferred explanations is crucial for SaIRA, for the reasons discussed earlier in this chapter, a fact that makes FASTXPLAIN inapplicable.

Having discussed the necessity for it, we provide a contribution to current research in constraint-based explanation generation by proposing a novel ECP-based algorithm for the generation of explanations for over-constrained situations specifically designed for safety-critical contexts: `wsm decision-repair`. The algorithm is discussed in detail in the next section and the approach is validated through a comparison with state-of-the-art techniques.

B.1.5 Weighted-Sum Model Decision Repair

B.1.5.1 Related work

Jussien and Debruyne [2007] give a detailed discussion about explanation-based repair techniques for constraint programming and convey their observations on the `PLM` algorithm (Algorithm B.4), which is an archetype for explanation-based repair.

The CHOCO solver implements a version of `PLM` which combines arc-consistency maintenance and tabu search: `decision-repair` [Jussien and Lhomme 2002] (Algorithm B.5).

The algorithm starts by assigning C_D with an initial set of decisions that is determined by some initialization method. Then it loops applying the filtering to $C \cup C_D$ and generating new sets of constraints $C' = \phi(C \cup C_D)$. The function `obviousInference` interprets the output C' of the filtering operator and returns one of the following answers:

- *solution*: a solution is found; C' is returned;

Algorithm B.4 PLM(V, C, C_D)

```
1:  P ← {V, C, C_D}
2:  repeat
3:     P ← filter(P)
4:     if check(P) = no solution then
5:        P ← forget(repair(record(P)))
6:     else if check(P) = solution found then
7:        return P
8:     else if check(P) = not enough information then
9:        P ← extend(P)
10:    end if
11: until conditions of termination
```

Algorithm B.5 `decision-repair`(C)

```
1:  C_D ← any initial set of decisions
2:  repeat
3:     if condition of failure satisfied then
4:        return false
5:     else
6:        C' ← φ(C ∪ C_D)
7:        if obviousInference(C') = no solution then
8:           k ← conflict explaining the failure
9:           C_D ← neighbour(C_D, k, Γ)
10:       else if obviousInference(C') = solution then
11:          return C'
12:       else
13:          C_D ← extend(C_D, Γ)
14:       end if
15:    end if
16: until false
```

- *flounder*: the algorithm tries to extend the current set of decisions C_D by adding a decision constraint through the `extend`(C_D, Γ) function: the function selects a decision constraints and adds it to C_D (the parameter Γ is used for customised versions of the algorithm);
- *no solution*: in this case $C \cup C_D$ is inconsistent, and C_D cannot be extended. The algorithm tries to repair the conflict by selecting a new set of decision constraints through the function `neighbour`(C_D, k, Γ).

The authors consider `decision-repair` a family of algorithms because several parameters remain undefined, e.g. the way of handling the neighbourhood. Jussien and Lhomme provide the following description:

> `decision-repair` is a generic algorithm, instances are obtained by specialising several parameters:
>
> - the nature and behavior of Γ the storage structure;
> - the neighboring computation function (`neighbour`);
> - the extension computation function (`extend`);
> - the failure conditions that indicate when to halt the search;
> - the filtering techniques to be used (the ϕ function).

`tabu decision-repair` is a member of the `decision-repair` family which is of particular interest for this thesis for several reasons, including the fact that it has been found to perform particularly well with open-shop problems similar to ADR [Jussien and Lhomme 2002]. This is a version of `decision-repair` which combines tabu search and bespoke heuristics; the decision constraint to be evaluated for negation are kept in a tabu list Γ. Jussien and Lhomme provide the following concise description of the logic of the algorithm:

> [The `tabu decision-repair` algorithm] tries to find one decision in k such that negating this decision makes the decision set compatible with all the conflicts. When several decisions can be negated, we use the following heuristics, which we call weighting-conflict heuristics: a weight is associated with each decision; the weight characterizes the number of times that the decision has appeared in any conflict. A weighting-conflict heuristic that works well takes into account the arity of the conflicts. Each time a conflict is found, the weight of its decision constraints is increased by $1/r$ where r is the arity of the conflict. The `neighbour` function chooses to negate the decision with the greatest weight that, when negated, makes the new decision set compatible with all the conflicts in Γ. If such a decision does not exist, the neighborhood can be extended. For example, we may try to negate two decisions.

Jussien and Lhomme also give the following definitions which offer an introduction to the material presented later in this section:

Definition B.1.7 (Conflict) *A conflict k for a set of constraints C and a decision set C_D is a subset of C_D, $k \subseteq C_D$ such that $C_D \wedge k \Rightarrow$ false.*

Let $c \in k$ be a constraint in the conflict k to be removed from the decisions set C_D. A neighbour C'_D of a decision set C_D is obtained by removing c from C_D and adding its negation.

Definition B.1.8 (Neighbour w.r.t. one conflict) *Let k be a conflict for a decision set C_D, a neighbour of C_D w.r.t. k is a decision set C'_D such that $\exists c \in k, C'_D = C_D \backslash c \cup \{\neg c\}$*

The concept of neighbourhood can be extended to several conflicts.

Definition B.1.9 (Neighbour w.r.t. several conflicts) *Let Γ be a set of conflicts and let C_D be a decision set. A neighbour C'_D of C_D w.r.t. Γ satisfies $\exists c \in k, C'_D = C_D \backslash c \cup \{\neg c\}$ and C'_D is compatible with the conflicts in Γ.*

The `neighbour` function, which is the core of `tabu decision-repair`, is reported in Algorithm B.6.

Algorithm B.6 `neighbour`(C_D, k, Γ)

**From the `tabu decision-repair` algorithm [Jussien and Lhomme 2002]*

Require: $k \subset C_D$, C_D is compatible with Γ
1: `add` k to the list of conflicts Γ
2: **if** `sizeof`$(\Gamma) > s$ **then**
3: `remove` the oldest element of Γ
4: **end if**
5: $L \leftarrow$ ordered list (decr. weight) of decisions in k
6: **repeat**
7: `remove` the first decision c from L
8: $C'_D \leftarrow C_D \backslash \{c\} \cup \{\neg c\}$
9: **if** C'_D is compatible with all conflicts in Γ **then**
10: **return** C'_D
11: **end if**
12: **until** L is empty
13: **return** stop (or extend the neighbourhood)

`tabu decision-repair` is designed for *generic* combinatorial problems and aims at converging to a solution as quickly as possible. Because the algorithm is generic, no assumptions are made concerning the nature of the constraints being processed; hence, no preferences can be expressed. In other words, no information is available *a priori* concerning the importance of each constraint and the type of requirement it implements (e.g. safety requirement, functional requirement).

The situation is slightly different in the context of SaIRA. For instance, given the safety-critical nature of the problem, the designers know that certain safety-related constraints have higher priority over other constraints. In the majority of cases, they are the last constraints to be relaxed in cases of conflict. Safety is only one example of the metrics of importance of constraints that the designers of the system can have at their disposal. By exploiting the availability of domain-dependent knowledge, it is possible to design bespoke conflict repair logic (possibly) able to improve the overall performance of the reconfiguration algorithm.

Besides performance, there is another aspect of the reconfiguration problem that could benefit from a bespoke conflict repair logic: the capability of the algorithm to generate *helpful* and *meaningful* explanations for the pilot. With `tabu decision-repair`, each decision taken by the algorithm can be translated in natural language as follows: *'Constraint A' was chosen instead of 'Constraint B' because the former has appeared more time in other conflicts previously explored* (in terms of ADR, the two constraints in question could represent the availability of two different avionics functions, for example). This is not a helpful or convincing explanation for a pilot facing an ADR decision, even though it explains the 'generally efficient' logic followed by the solver. A more helpful explanation would be: *'Constraint D' was chosen instead of 'Constraint E'. Reason: safety improvement.* In order to generate the latter type of explanation, the conflict repair algorithm must have knowledge about (a) the nature of the constraints and (b) their relative importance.

The next section proposes a novel member of the `decision-repair` family of algorithms based on the Weighted-Sum Model.

B.1.5.2 The `wsm decision-repair` algorithm

Montano [2007] investigates the applicability of several multi-criteria decision making algorithms (also known as multi-criteria decision analysis algorithms) to the problem of eliciting preferences from the constraints of an ADR CSP. The Weighted-Sum Model (WSM) and the lexicographic approach were found to be particularly suitable for the characteristics of the ADR problem. The difficulties of applying a lexicographic ordering to the constraints of an ADR CSP were discussed above with the introduction of the QuickXPlain algorithm (Sections B.1.2 and B.1.4). Here a WSM-based method developed in this research is proposed.

Let $R = \{r_1, r_2, \ldots, r_n\}$ be a set of n criteria of evaluation for m alternatives. Let w_j be the weight of importance of the criterion r_j and let be a_{ij} the performance value of alternative A_i when it is evaluated in terms of criterion r_j. The importance of alternative A_i results from the combination of all the criteria, it is denoted as A_i^{score} and it is calculated as follows:

$$A_i^{\text{score}} = \sum_{j=1}^{n} w_j a_{ij} \tag{B.6}$$

All the criteria are assumed to be monotonically increasing (increasing weight always means increasing importance) and are expressed using the same unit.

The logic of `wsm decision-repair` can be informally described as follows:

> The designers define n criteria for the evaluation of the constraints. Typical criteria are 'safety', 'performance', 'pilot's comfort'; a typical safety constraint is one that enforces the triple bus redundancy, whilst an example of a constraint that is important both in terms of safety and pilot comfort is one that enforces the availability of the Elevator Feel System.
>
> A constraint is evaluated against each criterion. Let $\Delta = (X, D, Z, H)$ be the dynamic constraint network associated with the ADR CSP. In the design phase, weights are as-

sociated with the dynamic constraints (referred to as 'decision constraints' by Jussien et al.) in H.

A different set of weights is devised for each aircraft operating mode in order to reflect the different importance that the same constraint assumes in different modes (or phases of flight). For instance, the constraints that require the activation of the Way-Points Generator has high importance in terms of safety in the landing phase but it is negligible during reconnaissance. Obviously, the set of active weights changes with the system mode transitions.

In cases of conflict during the search for an applicable configuration, `wsm decision-repair` tries to find one constraint in H such that negating this constraints eliminates the contradiction. If only one solution is found, that constraint is relaxed and the information about its importance—in terms of the predefined criteria—is used to inform the pilot about how the aircraft functionality has been affected. For example, consider the case of a constraint c_i which enforces the availability of the Elevator Feel System and that is important both in terms of safety (with weight $w_{\text{safety}}(c_i)$) and comfort (with weight $w_{\text{comfort}}(c_i)$). When negated, a message for the pilot might be:

Elevator Feel System: DEACTIVATED
Impact: SAFETY (↓), COMFORT (↓)

If more than one constraint in H can be negated, the `neighbour` function chooses to negate the constraint that scores less with Equation B.6 and that, when negated, makes the new decision set compatible with all the conflicts in Γ. As with other `decision-repair` algorithms, if such a decision does not exist, the neighborhood is extended, i.e. an additional decision is negated.

Note that, in cases of conflict and multiple decisions to be negated, the dynamic weights for each criterion allow the selection of the constraint with the lowest importance *given the current operating conditions*.

The `neighbour` function for `wsm decision-repair` is now defined more formally. The following sets are defined:

- C, C_D and k are defined as in Definition B.1.7;
- $M = \{m_1, m_2, \ldots, m_k\}$ is the set of all system operating modes;
- $R = \{r_1, r_2, \ldots, r_m\}$ is the set of criteria used for the evaluation of the constraints.

For each operating mode $m_i \in M$, a function f_{weight} associates a weight $w \in W$ with each criteria $r \in R$:

$$f_{weight} : R(m_i) \rightarrow W \text{ , where } w_i \in [0, 1] \subset \mathbb{R}^+ \tag{B.7}$$

For each constraint $c \in C$, a function f_{rank} associates a ranking $q \in Q$ with each criteria:

$$f_{rank} : C \rightarrow Q \tag{B.8}$$

where $Q = \{q_1, q_2, \ldots, q_n\}$, $q_i \in [0, 1] \subset \mathbb{R}^+$ and $|Q| = |R|$.

It follows that each constraint $c \in C$ is evaluated on the basis of the score calculated by means of the following WSM function:

$$\Psi(c_i) = \sum_{i=1}^{n} f_{weight}(r_i) f_{rank}(c_i) \tag{B.9}$$

In (B.9) the weights w_i are used to provide a ranking of the constraints which adapts to the current operating mode. The weights have a specific purpose for the problem of user interaction. For example, suppose that during a reconfiguration the pilot wants to override the logic of the system and give more importance to preserving one criteria over the others (e.g. giving more importance to safety); the weights can be changed dynamically through the SaIRA interface. Through this mechanism, repair options that do not favour the safety of the configuration are implicitly filtered out when a conflict arises.

Note that both the definition of the weights given *a priori* by the system designers and the eventual modification of the weights done by the pilot are much easier problems than setting up a new lexicographic ordering of the constraints to elicit different preferences (which is the approach adopted by QuickXPlain). For instance, the interface can be designed to allow the pilot to click on a button named 'SAFETY' in order to dynamically increase the importance of the 'safety metric' at the expense of the other metrics in real-time; the proposed configuration would be automatically recalculated, using Equation B.9, to favour solutions in which constraints that affect the safety of the system are not retracted. This type of interaction would be impossible having defined the ordering of the constraints by means of a lexicographic ordering.

The `neighbour` function for the `wsm decision-repair` algorithm is reported in Algorithm B.7.

Algorithm B.7 `neighbour`(C_D, k)

**Bespoke `neighbour` function for the `wsm decision-repair` algorithm*

Require: $k \subset C_D$

1: $m \leftarrow$ `getCurrentMode`
2: $P \leftarrow$ `getPilotPreferences`
3: $W \leftarrow$ `adjustWeights`(P, W)
4: $\Omega \leftarrow$ `WSMrank`(k, W, m)
5: **repeat**
6: `remove` the first decision c from Ω
7: $C'_D \leftarrow C_D \backslash \{c\} \cup \{\neg c\}$
8: **if** C'_D is compatible with all conflicts in Ω **then**
9: **return** C'_D
10: **end if**
11: **until** Ω is empty
12: **return** stop (or extend the neighbourhood)

`WSMrank` orders the constraints in the conflict k in decreasing order of importance, implementing Equation B.9. Steps two and three are optional and are executed only in cases where the pilot wants to modify the pre-defined ranking of importance of the evaluation criteria. In these cases,

`getPilotPreferences` retrieves information about the pilot's preferences from the SaIRA interface and `adjustWeights` uses the information obtained to adjust the pre-defined vector of weights.

`WSMrank` implements function (B.9) and ranks the constraints returning the partial order Ω.

The constraint c which scores less in Ω is negated. Given that Ω is a partial order, in the case of constraints with equal score, precedence is given to the negation of constraints which are not related to safety (given the safety-critical operating scenario); this is a decision taken specifically for SaIRA, another heuristic could be adopted for a different system. If the contradiction is not resolved, more decision constraints sequentially selected from Ω are negated.

In order to facilitate the process of associating weights with constraints and cope with large CSP, an Analytic Hierarchy Process [Saaty 1980] can be used by the designer (Figure B.1). General criteria can be divided in sub-criteria and constraints can be grouped into sets which have the same weight for a number of criteria. It is always possible to establish a hierarchical organisation for the constraints of a CSP [Jussien and Ouis 2001]. This approach has been found to be particularly efficient in this research.

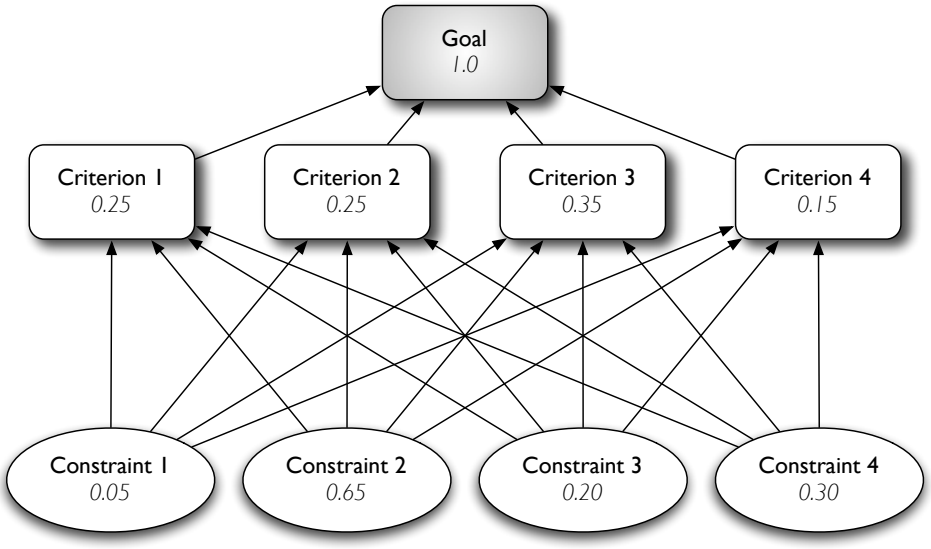


Figure B.1: Example of how to use an Analytic Hierarchy Process to associate weights to constraints.

B.1.6 Evaluation of WSM Decision-Repair

Jussien and Lhomme [2002] performed an in-depth evaluation of `tabu decision-repair` varying the three main tuning parameters of the algorithm:

- the maximum number of allowed movements without improvement;
- the length of the tabu list;
- the selection of the candidate neighbour for repair (handled through the definition of the weight of each constraint in the conflict).

The authors conducted several experiments using open-shop problems. Concerning the conflict repair strategy—which is the core of `wsm decision-repair`—Jussien and Lhomme make a direct comparison between `tabu decision-repair` and the well-established `mac-dbt` [Jussien et al. 2000], which combines arc-consistency with dynamic backtracking and, since it does not perform a local search because of the completeness requirements, it can be considered a peculiar member of the `decision-repair` family.

For the sake of consistency and traceability, a series of experiments is performed here to evaluate the performance of `wsm decision-repair` with respect to `tabu decision-repair` and `mac-dbt`, harking back to the approach of Jussien and Lhomme. Whilst these authors used generic benchmarking open-shop problems, here typical (programmatically created) ADR problems are used. Open-shop problems, although quite simple to enunciate, are hard to solve optimally: simple instances of size 6 x 6 (number of jobs by number of machines) leaves 36 unsolved variables [Guéret and Prins 1998]. Indeed, instead of focusing on performance alone, this study gives more importance to finding a trade-off between performance and ability to generate explanations that are helpful for ADR. As a result, the structure of the ADR CSP used here is the one described in Section 5.4.1 and used throughout this chapter as an example of a typical but simplified ADR problem. However, in the experiments presented in this section, the size of the problem is strongly increased and other parameters are manipulated, as discussed hereinafter, in order to reflect a more realistic case.

The `tabu decision-repair` has been implemented as described by Jussien and Lhomme [2002]. The same implementation was used as a baseline for the implementation of `wsm decision-repair` but Algorithm B.7 was used to implement the `neighbour` function. `wsm decision-repair` is configured as follows (see Jussien and Lhomme [2002] for the configuration of `tabu decision- repair`):

- *Filtering techniques*. Precedence constraints are handled with 2B-consistency filtering [Lhomme 1993];
- *Computation of conflicts*. Conflicts are computed using the facilities provided by Choco, as described by Jussien [2001];
- *Neighbourhood*. Algorithm B.7 is used;
- *Pilot preferences*. In order to verify the performance of the `wsm decision-repair` and compare it to the other two algorithms, pilot preference elicitation is disabled in these experiments. This means that steps 2 and 3 in Algorithm B.7 are not implemented and `WSMrank`, at step 4, uses the pre-defined weights;
- *Stopping criterion*. As with `tabu decision-repair`, the algorithm stops either when a stop is returned by the `neighbour` function or when the limit of maximum iterations without improvement since the last solution is reached (1,500 iterations in these experiments).

The experiments were performed on an Apple MacBook computer (Intel Dual Core2, 2.16Ghz, 2Gb RAM) running Mac OSX 10.5 Leopard, Java 1.6 with maximum Java memory heap size of 1.5 Gb. The test application was implemented in Java using Choco 2.1.1.

B.1.6.1 Satisfaction rate

Satisfaction rate was introduced by Schubert et al. [2010] as an index of how many constraints in a configuration CSP are satisfied by all the alternatives the user is supposed to decide from. This value is important because the number and size of the conflict sets (i.e. explanations) is highly dependent on it. A low satisfaction rate results in a high number of conflict sets with a low cardinality.

In the context of the ADR problem, the lower the satisfaction rate, the higher the impact of the unexpected event on the system. Schubert et al. created 7 different settings, increasing satisfaction rate r from 10% to 70% for a CSP characterised by 10 constraints. This setting is unrealistic for the ADR problem, because firstly, a typical ADR CSP has many more constraints; as an example, the simplified ADR problem described earlier in this chapter (Section 5.4.1) has 29 constraints and it is still an underestimation of a real problem. Secondly, in terms of avionics reconfiguration, a satisfaction rate of 10% means that the system is so affected by the unexpected event that only 10% of the constraints can be satisfied by a configuration. Whilst this is acceptable for a mobile phone configuration problem (like the one examined by Schubert et al.), in the ADR context, such a low satisfaction rate would mean that the aircraft would be destroyed. As a result, this experiment uses four CSP settings with an increasing satisfaction rate r (the independent variable) going from 80% to 95%.

The dependent variable is the *runtime*, which is the time from the instant at which the algorithm is started to the instant at which it returns with *two applicable configurations* and *annexed explanations*; the CSPs are over-constrained. At this stage, the explanations have not yet been converted into natural language.

Thirty CSPs were automatically generated, with 56 constraints acting on 37 variables; 22 out of the 56 constraints defined were static, hence, they could not be negated during the search and were propagated at the beginning of the enumeration process. For each problem, the constraints set was manipulated in order to produce four degrees of satisfaction rate: 80%, 85%, 90% and 95%. As previously discussed, the solution given by the eCP solver is dependent on the enumeration process; therefore, the first two solutions found mark the end of the search process.

For `wsm decision-repair`, five different criteria of evaluation of the decision constraints were defined: safety, communications, redundancy, resilience, automation.

Either because it was not possible to verify the parametric behaviour of the dependent variables or because there was simply not enough knowledge in this respect, non-parametric significance tests are used in the experiments presented in this chapter.

The descriptive statistics of the results are reported in Table B.4 and Figure B.2.

The results for each satisfaction rate were compared by means of Friedman's test; the outcome is reported in Table B.5.

The results of a post-hoc test using Wilcoxon Signed Rank tests with Bonferroni correction are reported in Table B.6. Whilst both `tabu-dr` and `wsm-dr` performed statistically better than `mac-dbt` under all conditions, the post-hoc test reveals that `wsm-dr` performs better with higher satisfaction rate whilst `tabu-dr` is more scalable, overtaking `wsm-dr` when r is lower. This phe-

Runtime*: effect of *satisfaction rate			
	`mac-dbt`	`tabu decision-repair`	`wsm decision-repair`
$r = 80\%$	0.757 (0.300)	0.391 (0.041)	0.502 (0.054)
$r = 85\%$	0.364 (0.091)	0.225 (0.047)	0.236 (0.031)
$r = 90\%$	0.211 (0.031)	0.137 (0.027)	0.119 (0.017)
$r = 95\%$	0.175 (0.023)	0.079 (0.010)	0.051 (0.003)

Table B.4: Effect of different percentages of satisfaction rate r on the *runtime* (measured in seconds) of `mac-dbt`, `tabu decision-repair` and `wsm decision-repair`. The statistical values in the table are obtained from 30 ADR problems (56 constraints, 37 variables) for each satisfaction rate.

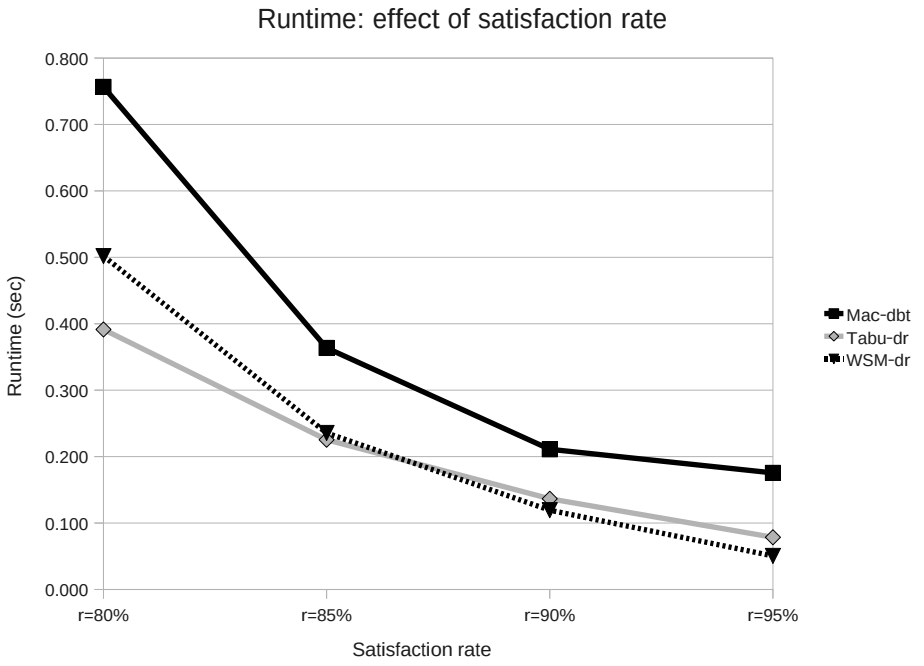


Figure B.2: Effect of different satisfaction rates (X axis) on the runtime (Y axis, measured in seconds) for `mac-dbt`, `tabu decision-repair` and `wsm decision-repair` algorithms. Whilst `wsm-dr` performes better at higher satisfaction rates, the cost of better explanations it provides is paid in terms of scalability with respect to `tabu-dr`, which performs better with lower satisfaction rates.

nomenon is clear in Figure B.2.

In conclusion, `wsm-dr` shows better performance at a high satisfaction rate; the cost of the better explanations that this algorithm provides (using domain-dependent knowledge within the `neighbour` function) is paid for with less scalability with respect to satisfaction rate than `tabu-dr` (e.g. worse performance when satisfaction rate is below approximately 80%). This is not really an issue in SaIRA because ADR problems with very low satisfaction rates represent situations in which the aircraft is almost destroyed, which is an unrealistic scenario of application.

B.1.6.2 Problem complexity

In the previous experiment the CSP was characterised by 56 constraints (22 static, 34 dynamic) and 37 variables.

Satisfaction rate: significance test	
Rate	**Friedman's test**
r **= 80%**	$\chi^2(2) = 29.267$, $p < 0.001$
r **= 85%**	$\chi^2(2) = 28.467$, $p < 0.001$
r **= 90%**	$\chi^2(2) = 43.4$, $p < 0.001$
r **= 95%**	$\chi^2(2) = 60.0$, $p < 0.001$

Table B.5: Results of Friedman's tests for the effect of the satisfaction rate.

Satisfaction rate: post-hoc test		
Rate	**Pairwise comparison**	**Significance**
r **= 80%**	tabu-dr ↔ wsm-dr	$Z = -3.615$, $p < 0.001$
	tabu-dr ↔ mac-dbt	$Z = 5.293$, $p < 0.001$
	wsm-dr ↔ mac-dbt	$Z = 3.678$, $p < 0.001$
r **= 85%**	tabu-dr ↔ wsm-dr	$Z = -1.42$, n.s.
	tabu-dr ↔ mac-dbt	$Z = 5.164$, $p < 0.001$
	wsm-dr ↔ mac-dbt	$Z = 3.744$, $p < 0.001$
r **= 90%**	tabu-dr ↔ wsm-dr	$Z = 1.162$, n.s.
	tabu-dr ↔ mac-dbt	$Z = 6.197$, $p < 0.001$
	wsm-dr ↔ mac-dbt	$Z = 5.035$, $p < 0.001$
r **= 95%**	tabu-dr ↔ wsm-dr	$Z = 3.873$, $p < 0.001$
	tabu-dr ↔ mac-dbt	$Z = 7.746$, $p < 0.001$
	wsm-dr ↔ mac-dbt	$Z = 3.873$, $p < 0.001$

Table B.6: Post-hoc test (Wilcoxon's test with Bonferroni correction) for the Friedman's test on the effect of the satisfaction rate (Table B.5).

Here the scalability of `wsm decision-repair` is assessed by increasing the complexity of the CSP as shown in Table B.7. The number of static constraints is not modified, because they cannot be negated during the search, therefore, they do not influence the complexity of the problem.

Benchmark configurations			
Benchmark name	**Nr. of variables**	**Nr. of static constraints**	**Nr. of dynamic constraints**
Baseline	37	22	34
Large	58	22	71
Complex 1	74	22	103
Complex 2	96	22	153

Table B.7: Characteristics of the four benchmark problems used for the assessment of the effect of problem complexity on the algorithm runtime. Static constraints cannot be negated at runtime, hence their number is not manipulated in this experiment.

Thirty over-constrained CSPs were generated for each configuration; the satisfaction rate was set to 90% for all problems. The descriptive statistics for the results are shown in Table B.8.

The statistical significance of the differences was verified using Friedman's test; the results are reported in Table B.9.

A series of Wilcoxon Signed Rank tests with Bonferroni correction was used for post-hoc analysis; the results are shown in Table B.10.

Runtime: effect of *problem complexity*			
	`mac-dbt`	`tabu decision-repair`	`wsm decision-repair`
Baseline	0.21 (0.03)	0.14 (0.02)	0.12 (0.02)
Large	0.72 (0.06)	0.49 (0.07)	0.39 (0.05)
Complex 1	1.7 (0.08)	1.4 (0.19)	1.1 (0.07)
Complex 2	8.11 (0.18)	5.28 (0.79)	3.9 (0.09)

Table B.8: Effect of problems complexity (intended as number of variables an constraints) on the runtime (seconds) of the `mac-dbt`, `tabu-dr` and `wsm-dr` algorithms. The statistical values in the table are obtained from 30 problems for each benchmark.

Problem complexity: significance test	
Benchmark name	**Friedman's test**
Baseline	$\chi^2(2) = 44.838$, $p < 0.001$
Large	$\chi^2(2) = 54.2$, $p < 0.001$
Complex 1	$\chi^2(2) = 52.267$, $p < 0.001$
Complex 2	$\chi^2(2) = 58.067$, $p < 0.001$

Table B.9: Results of the Friedman's tests for the effect of problem complexity.

Figure B.3 shows clearly that `wsm-dr` scores better than both `tabu-dr` and `mac-dbt`. As confirmed by the post-hoc test, `wsm-dr` and `tabu-dr` have a similar performance on simple problems (see the statistical non-significance between `wsm-dr` and `tabu-dr` in relation to the 'Baseline' benchmark, first row in Table B.10), but when the complexity increases, `wsm-dr` scales much better. This result is both positive and expected: the domain-dependent knowledge used by `wsm-dr` provides a considerable boost in the search for a solution, especially with growing numbers of variables and constraints.

Satisfaction rate: post-hoc test		
Benchmark	**Pairwise comparison**	**Significance**
Baseline	wsm-dr ↔ tabu-dr	$Z = 1.291$, n.s.
	wsm-dr ↔ mac-dbt	$Z = 6.261$, $p < 0.001$
	tabu-dr ↔ mac-dbt	$Z = 4.97$, $p < 0.001$
Large	wsm-dr ↔ tabu-dr	$Z = 3.486$, $p < 0.001$
	wsm-dr ↔ mac-dbt	$Z = 7.359$, $p < 0.001$
	tabu-dr ↔ mac-dbt	$Z = 3.873$, $p < 0.001$
Complex 1	wsm-dr ↔ tabu-dr	$Z = 3.615$, $p < 0.001$
	wsm-dr ↔ mac-dbt	$Z = 7.23$, $p < 0.001$
	tabu-dr ↔ mac-dbt	$Z = 3.615$, $p < 0.001$
Complex 2	wsm-dr ↔ tabu-dr	$Z = 3.615$, $p < 0.001$
	wsm-dr ↔ mac-dbt	$Z = 7.617$, $p < 0.001$
	tabu-dr ↔ mac-dbt	$Z = 4.002$, $p < 0.001$

Table B.10: Post-hoc test (Wilcoxon's test with Bonferroni correction) for Friedman's test on the effect of problem complexity (Table B.9).

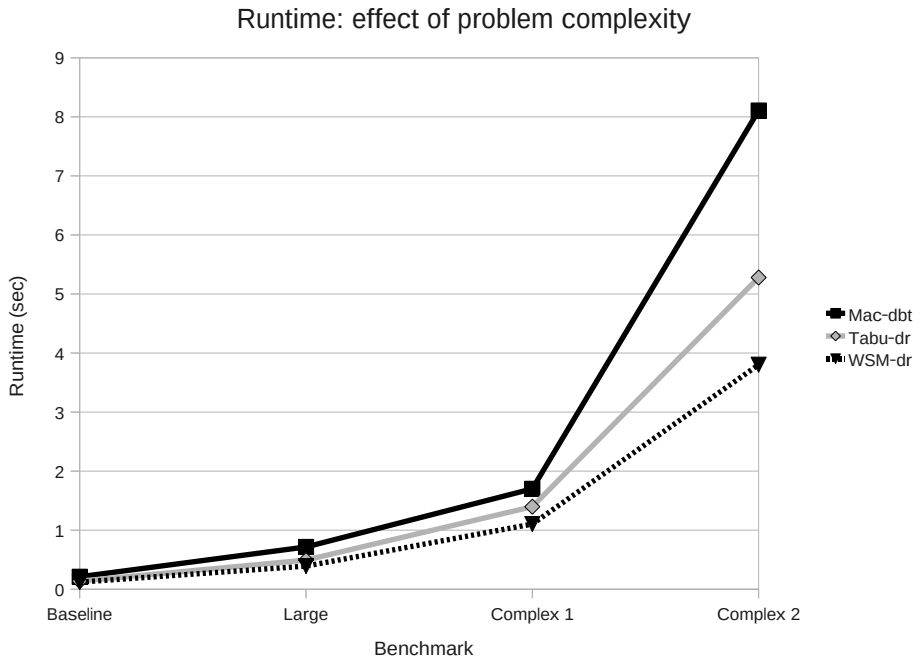


Figure B.3: Effect of problem complexity on the runtime (in seconds) of `wsm decision-repair`, `tabu decision-repair` and `mac-dbt`. `wsm decision-repair` provides the best results both in terms of average runtime and scalability.

B.1.6.3 Number of criteria of evaluation

Any number of criteria for the evaluation of the constraints during the conflict repair process could be defined in `wsm decision-repair`. Intuitively, a large number of criteria leads to higher granularity in the structure of the explanations and more informed decisions during conflict repair; on the other hand, a small number of criteria leads to easier problem design, better performance resulting from lower computation for the `WSMrank` at the cost of less precise explanations.

This experiment only concerns `wsm decision-repair` and aims at characterising its performance under the effect of different numbers of criteria considered by the conflict repair function.

Each of the four benchmarks defined for the previous experiment (Baseline, Large, Complex 1 and Complex 2) were performed again, with 5, 15, 30, 50 and 100 criteria. For each benchmark, 25 CSPs were randomly generated, with the satisfaction rate set at 90%. There were 5 observations for each of the 5 criteria, leading to a total of 100 observations (5 CSP x 5 criteria levels x 4 benchmarks). The descriptive statistics are presented in Table B.11.

As with the previous experiments, Friedman's test is used to verify the statistical significance of the results. The results of the test are reported in Table B.12.

Whilst Friedman's test reveals a statistical difference at runtime, resulting from different numbers of criteria, a post-hoc analysis performed by means of a series of Wilcoxon Signed Rank tests with Bonferroni correction (not reported here for the sake of brevity given the large number of permutations) shows that the statistical differences only come from the comparison of the extrema

Runtime: effect of *number of criteria*				
	Baseline	**Large**	**Complex 1**	**Complex 2**
5 criteria	0.120 (0.002)	0.412 (0.075)	1.095 (0.056)	3.790 (0.101)
15 criteria	0.121 (0.003)	0.385 (0.065)	1.143 (0.052)	3.852 (0.082)
30 criteria	0.122 (0.002)	0.396 (0.052)	1.156 (0.061)	3.856 (0.067)
50 criteria	0.131 (0.002)	0.431 (0.042)	1.173 (0.062)	3.896 (0.085)
100 criteria	0.151 (0.006)	0.432 (0.052)	1.205 (0.045)	4.090 (0.097)

Table B.11: Effect of the number of criteria (used to rank the constraints during conflict repair) on the runtime (measured in seconds) of `wsm decision-repair`. The statistical values in the table are obtained from 25 problems for each level of number of criteria.

Number of criteria: significance test	
Benchmark	**Friedman's test**
Baseline	$\chi^2(4) =, p < 0.001$
Large	$\chi^2(4) = 11.157, p < 0.025$
Complex 1	$\chi^2(4) = 28.85, p < 0.001$
Complex 2	$\chi^2(4) = 52.0, p < 0.001$

Table B.12: Results of Friedman's tests for the effect of the number of criteria used used by the Weight Sum Function to rank the constraints.

in the scales (e.g. 5 criteria *versus* 100 criteria). This phenomenon is clear in Figure B.4. In other words, the number of criteria has a statistical effect on the runtime, but the size of this effect is negligible compared to the effects of other factors (e.g. problem complexity). Figure B.4 also provides a clear idea of how the number of criteria does not affect the scalability of the algorithm, at least in the context of the type of problems that are typical for ADR.

B.1.6.4 Conclusions

The main conclusions stemming from the experimental evaluation of `wsm decision-repair` presented in the sections above, can be summarised as follows:

- `wsm-dr` has performance comparable to state-of-the-art conflict repair algorithms; it provides even better performance than `tabu-dr` in problems characterised by a high satisfaction rate, but the cost of best explanations is paid for in terms of less scalability (but only in relation to the satisfaction rate);
- `wsm-dr` provides better performance than `mac-dbt` and `tabu-dr` at any degree of problem complexity (i.e. better scalability);
- The number of criteria defined for the evaluation of the constraints during conflict repair allows the practitioners to define the type and granularity of the decision support information generated by `wsm-dr`; the experiments reveal that this factor has a negligible effect on the runtime of `wsm-dr` and does not scale with the increase of problem complexity.

On the basis of the analyses performed, `wsm decision-repair` appears highly effective for performing conflict repair for ADR problems: the algorithm has the main benefits of both pro-

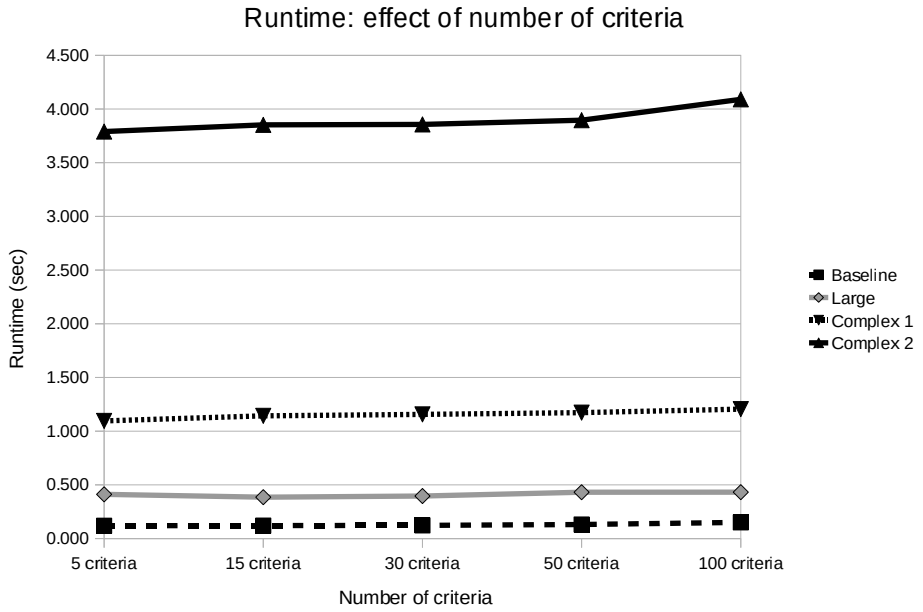


Figure B.4: Effect of the number of criteria (used to rank the constraints during conflict repair) on the runtime (measured in seconds) of `wsm decision-repair`. Whilst there is a statistically significant effect of the independent variable, the impact is negligible compared to the computational demands of other phases of the algorithm.

ducing the type of decision support information required in SaIRA and providing performance comparable to state-of-the-art conflict repair algorithms.

B.2 Generating Readily-Understandable Information

The `wsm decision-repair` algorithm generates computer-encoded explanations of conflict-repair actions. Depending on the number of conflicts repaired during the search for a configuration, the explanation generated can possibly become too large to be delivered to the pilot in a way that would efficiently support the human decision process. Furthermore, the information produced will not be readily understandable.

Two main problems can be identified: (a) the generation of readily understandable *explanations* using the information produced by `wsm decision-repair`, and (b) the generation of readily-understood *implications* for each decision alternative. Both problems are addressed in the following sections, following a brief discussion of relevant research in the literature.

B.2.1 Related work

Using logic puzzles as a case study, Sqalli and Freuder [1996]; Freuder et al. [2001*b*] introduce the use of inference-based constraint satisfaction to generate explanations for the decisions of the CSP solver during combinatorial problems. The type of explanations proposed are intended to improve the effectiveness of user involvement in the search for a solution to the problem.

The explanations devised by Freuder et al. are strongly user-centric. In the design phase, sys-

tem developers associate user-relevant information with each constraint; the information attached to each constraint is directed to the user in cases where the solver finds a contradiction which entails that constraint. The user decides how to proceed on the basis of that information, e.g. to relax the constraint.

Jussien [2003] has a different approach to explaining the decisions of the CP solver. His research interest is in automatically generating explanations for a given set of constraints without determining in advance the nature of the explanations which will be provided—the solver generates them dynamically. Jussien's approach is known as **explanation-based constraint programming (ECP)**, introduced earlier in the chapter.

In ECP, the solver records specific information whenever a value is deleted from the domain of a variable, in order to handle situations in which constraints are dynamically added or retracted from the constraint network. ECP distinguishes between two types of explanations: **contradiction explanations** and **eliminating explanations**. The former explains the conflict between a specific constraint and the assignments made previously; the latter specifies why values were deleted from a variable domain. The latter notion of explanation is similar to that of *justification* developed by Bessiere [1991] in the context of Dynamic CSPs.

The information recorded by the ECP solver during execution was originally seen as improving the performance of the solver, not for user support. In a paper titled 'Explanations for Whom?', Wallace and Freuder [2001] question the effectiveness of formal, 'solver-level' explanations for user support. They identify a gap between the formal representation of an explanation, as produced by the CSP solver, and the way it should be represented in order to be helpful to the user. The authors distinguish between *explanation* and *explanatory sign*, the latter being the way that the formal explanation, or part of it, can be conveyed to the user.

Jussien and Ouis [2001] overcome the lack of 'user-friendliness' of the explanations generated with ECP by proposing a set of tools to organise the constraints of a CSP hierarchically and produce more 'user-friendly' information, which can effectively support the user during interactions with the solver.

Having compared Freuder's and Jussien's approaches, Van Der Linden [2002] introduces the concept of **meta-constraint**. A meta-constraint has a minimum and maximum value and a set of constraints it ranges over. The CSP is organised into 'clusters' of meta-constraints which 'contain' sets of constraints. User-friendly explanations are associated with top-level constraints (those constraints which are at a high level in the hierarchy of clusters) and are directed to the user in cases when one or more constraints in the cluster reach a contradiction. The principle of organising the set of constraints in a hierarchical manner and associating high-level explanations with constraints high in the hierarchy suggested in Van Der Linden is very similar, in many aspects, to the method proposed earlier by Jussien and Ouis [2001], who also provide a rigorous, formal description of the explanation generation method.

In summary, the approaches to explaining the decisions of the CSP solver, developed by Freuder et al. on one side and Jussien et al. on the other, represent the two main trends for automated explanation generation for interactive CSPs. Freuder focuses on the perspicuity of the explanation, looking at small and homogeneous problems. Jussien tackles the problem of dy-

namically explaining the decisions of the solver in complex, big and unstructured combinatorial problems.

ADR is a complex combinatorial problem with a large search space, a feature of the problem addressed in this thesis that favours Jussien's approach to generating readily-understood explanations for pilot decision support.

Jussien and Ouis [2001] propose structuring the CSP in a hierarchical manner (as a tree). Low-level constraints are projected into higher level nodes. User-friendly information is attached to sets of constraints by the programmer: when one or more constraints in the set are relaxed, the attached information is directed to the user, providing a user-friendly information that *explains* to the user the function of the constraints in the context of the problem. This information should enable the user to make an informed choice.

In SaIRA, explanations for reconfiguration decisions are not required for all reconfiguration decisions, but only in case of over-constrained situations, in which one or more decision constraints must be negated. The previous section has already proposed the `wsm decision-repair` algorithm for this purpose. The mechanism of attaching user-friendly information to sets of constraints can be integrated with the algorithm in order to translate the output of `wsm-dr` into readily understandable information. Figure B.5 shows an example of how the ideas from Jussien and Ouis can be employed in the context of the ADR problem.

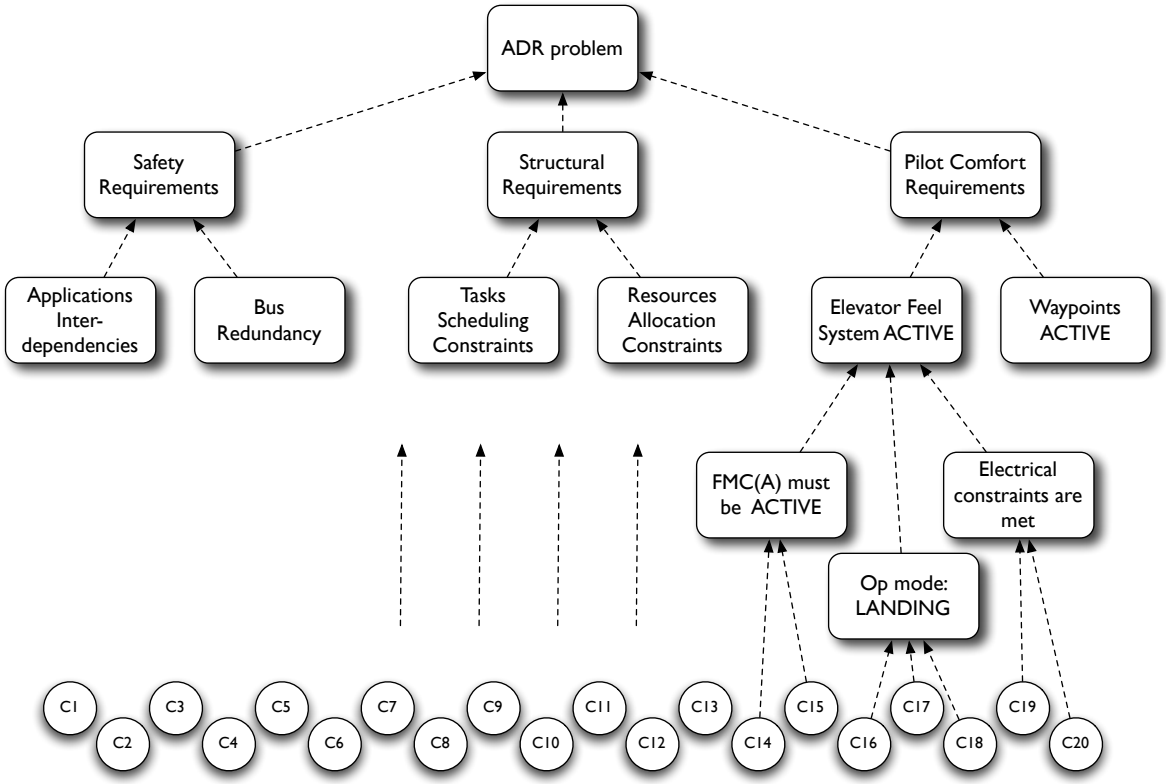


Figure B.5: An example of the application of the hierarchical organisation of user-friendly explanations of constraints [Jussien and Ouis 2001] to a small portion of a typical ADR problem.

At the bottom of the tree, the user-friendly information is less concise, but better reflects the

information carried by the attached constraints. Higher levels of explanations provide more concise information, but the information becomes more general and less precise. For instance, if one or more constraints from C14 to C20 are retracted, the user is provide with the message 'Elevator Feel System ACTIVE' (note that the wording is explanatory, a better framing can be used, but this is not important at this stage) the pilot would understand that the constraints relaxation is necessary to guarantee the functioning of the EFS function which, as a result, is going to be lost in the next configuration.

Navigating the tree towards the bottom allows the user to have more precise information about the specific aspect of the configuration proposed by the system which leads to the deactivation of the EFS; navigating towards the top of the tree provides greater abstraction and, in the example in question, would enable the pilot to realise that the deactivation of the EFS would lead to a decrease in comfort, just like the deactivation of the Waypoints function would do.

Two problems are quickly identified with this baseline approach:

1. *Filtering problem.* The pilot should only be provided with a limited amount of information for the decision support process to be effective; this information must be concise and helpful at the same time, e.g. representative of the current state of the system. When there are a lot of constraints that must be retracted, only a portion of the user-friendly information attached to the constraints can be delivered to the pilot. The problem of filtering the information is not addressed by Jussien and Ouis;

2. *Abstraction problem.* The method proposed by Jussien and Ouis is static, the hierarchy is defined at design time and it is used "as is" during the interaction. Whilst this is acceptable in the type of problems considered by Jussien and Ouis, in the case of ADR it is not possible to provide the pilot with a full explanation tree, because this would result in unhelpful decision support. At the same time, the problem of selecting the right degree of abstraction is not obvious, e.g. how should the system balance between precision of the information provided (at the expense of brevity) or abstraction (at the risk of not being able to inform the pilot of the real risks hidden in a configuration)? These questions are addressed hereinafter.

B.2.2 Natural Language Generation in SaIRA

Natural Language Generation is a subfield of Computational Linguistics and language-oriented Artificial Intelligence research devoted to the production of high-quality text from computer-internal representation of information.

Dale and Reiter [1995] characterise the input to a single invocation of a NLG system as a four-tuple $\langle k, c, u, d \rangle$, where k is the *knowledge source* to be used, c is the *communicative goal* to be achieved, u is the *user model*, and d is a *discourse history*.

The knowledge source k is usually encoded in a database that is accessed by the NLG system. In SaIRA, k is represented by the low-level explanations generated by the CSP solver (which include the information generated by `wsm decision-repair`).

The communicative goal c—the purpose of the message being generated—can be summarised as follows:

> For each reconfiguration alternative, provide the pilot with readily understandable *explanations* of the reconfiguration alternative proposed, *implications* on the functionality of the aircraft, and an *assessment of the reliability* of the information provided.

Relating to the concerns of the user model, Chapter 2 anticipated that in this research the user model (or profile) u would be developed using basic ideas from the cognitive psychology domain (Chapters 3 and 4); the conclusions reached were used to define the overall framework for automated generation of decision support information proposed in this chapter.

The discourse history d represents the text generated by the system before the current message. At present, SaIRA is designed as a 'single-interaction' system, hence no history is taken into consideration.

The literature contains several architectures for NLG systems. Figure B.6 is an adaptation from Dale and Reiter [1995]. The left side of the picture highlights the three main building blocks, document planner, microplanner and surface realiser.

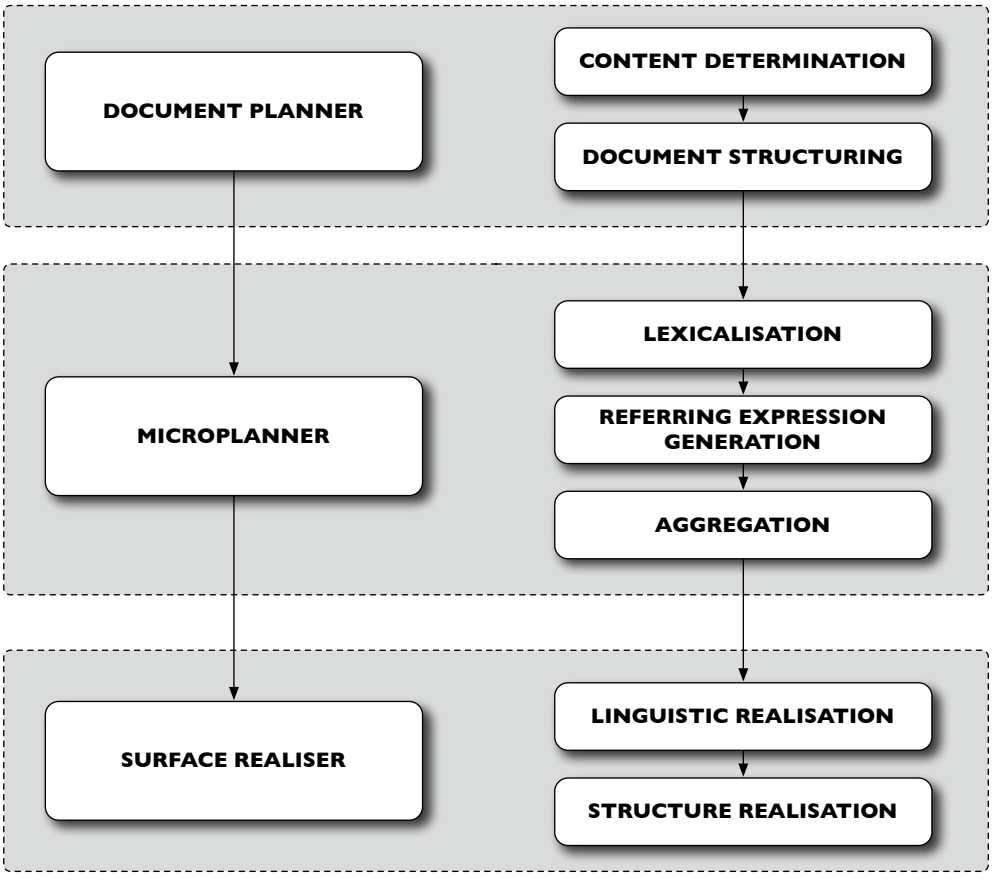


Figure B.6: TBD

Dale and Reiter give the following general description. The purpose of the **document planner** is to take the four-tuple $\langle k, c, u, d \rangle$ and determine content and structure of the document to generate. More specifically, **content determination** is the task of deciding what chunks of information are going to be included in the output text. **Document structuring** is the task of building a structure

that contains the portions of information (or 'messages') selected during the content determination phase.

The purpose of the **microplanner** is making fine-grained decisions about the form of the text being produced, e.g. deciding which syntactic structures to use. **Lexicalisation** is concerned with selecting the specific words or constructions. The **referring expression generation** component then decides how references should be made in the text. Finally, the **aggregation** step takes care of ordering the information and translating the structures created by the planner into sentences and paragraphs.

The **surface realiser** maps the abstract representation used by the microplanner into actual text. In particular, **linguistic** and **structure realisation** convert meta-content and meta-structures, as generated by the microplanner, into real text content and structures.

The decision support information required for SaIRA needs to be expressed in English, but it should have a fixed structure that the pilot can easily recognise and process in a timely manner. This requirement is motivated by the peculiar operating environment. As a result, the majority of the components characteristic of state-of-the-art, sophisticated NLG systems are not required. However, particular attention is paid to the content determination module, which requires some sophistication.

In SaIRA each message delivered by the system is always made up of the following four components: (1) fault description, (2) explanations, (3) implications and (4) reliability assessment. This thesis does not focus on the fault description (which is considered the baseline information generated by the standard cockpit instrumentation); it is assumed that a descriptive string of text is available for any fault which can be shown to the pilot when it happens, e.g. "Failure to the Left-Engine Power Generator". The attention of this thesis is on the decision support SaIRA associates with the fault description (components 2, 3 and 4 in the previous list).

The four components of the decision support message delivered by SaIRA are organised according to the *schema* given in Figure 5.10 which was already discussed in Section 5.7.2, when the design of the SaIRA user interface was described. The schema was designed in order to provide a fixed basis which is tailored with the data relative to each configuration alternative, so as to allow for the rapid processing of the information by the pilots (e.g. the pilots always know which information is where). `SaIRA-XPlain` is the algorithm proposed to dynamically fill the *schema* with information; `SaIRA-XPlain` makes use of `wsm decision-repair` to generate explanations and implications. The algorithm is discussed in the following section.

B.2.3 The SaIRA-XPlain algorithm

With reference to the schema presented in the Chapter 5 (Section 5.7.2), three portions of the message need to be generated by `SaIRA-XPlain`: the uncertainty figures, the explanations and the implications. The computation of the uncertainty figures is dealt with in the next section; here the focus is on explanations and implications.

Conflict-repair is a critical step in the reconfiguration process for the generation of explanations and implications. During a minor reconfiguration (i.e. the CSP is not over-constrained) no explanations and implications are necessary because the functionality of the system is unchanged.

Intuitively, the information that constitutes the explanations and implications is generated *during* the process of restoring an over-constrained problem (i.e. major reconfiguration), which involves retracting one or more decision constraints and switching to degraded functionality as a result.

In Section B.1.5.2 the `wsm decision-repair` algorithm for conflict-repair of ADR problems was proposed. This section shows how the information generated by `wsm decision-repair` during the conflict repair is used by a new algorithm, `SaIRA-XPlain`, which filters the information about explanations and implications in order to produce decision support to fill the schema shown in Figure 5.10.

Before presenting the algorithm, the logic is explained with the support of Figure B.7. As previously mentioned, information about fault diagnosis is assumed to be available before the execution of `SaIRA-XPlain`. SaIRA is designed to support the pilot during the process of fault recovery; fault detection and identification are beyond the scope of this research.

A hierarchical structure is preventively enforced on the decision constraints of the ADR problem as proposed by Jussien and Ouis (Figure B.5). However, instead of associating a single explanatory sign with each set of constraints, two pieces of readily-understood information are attached: (a) an explanatory sign ϵ and (b) an implication sign ι. More formally:

$$\forall c_i \in C \rightarrow (\epsilon_i, \iota_i) \tag{B.10}$$

where C is the set of constraints of the ADR problem, as in Definition 5.3.5.

If constraint c_i is negated in order to repair a conflict, the implication sign ι_i is considered for inclusion in the message shown to the pilot; the implication sign, pre-defined by the system designer, tells the pilot in English the implications of negating constraint c_i. For instance, let $c_i \in \{0, 1\}$ be the constraint that regulates the activation of the Elevator Feel System (0=inactive, 1=active); if $c_i = 1$ must be retracted, a typical implication sign could be:

Elevator Feel System → DEACTIVATED

Similarly, being $c_j = 3$ the constraint representing the degree of redundancy of a data bus, an example of implications sign to be shown when the constraint is negated could be:

Data Bus Redundancy → TRIPLE REDUNDANCY LOST

In the schema shown in Figure 5.10 (previous section), under the implication sign, there is an 'impact' field; SaIRA shows one (or two, if that is the case) metrics which are most affected by the repair action performed. This information is obtained directly from the results of the `WSMrank` function, as described in Section B.1.5.2.

The explanatory sign (formally defined in Section B.2.1) associated with one or more constraints explains why the constraints in question have been kept active at the expense of the negated ones. Typical explanatory signs associated with the same couple of constraints used for the previous example on implications would read as follows:

Elevator Feel System → REQUIRED in current mode

Data Bus Redundancy → TRIPLE (SAFETY requirement)

In conclusion, in SaIRA *an implication sign tells the pilot what happens when constraint c_i is negated; an explanatory sign tells the pilot why the constraint should not be relaxed, providing a quick insight into the rationale for the reconfiguration suggestion given by the system.*

As discussed in the sections above, there could be a need to negate several constraints from the conflict set in order to repair a contradiction. Similarly, there could be more than one implication for each decision constraint to be retracted (constraints are linked by inter-dependencies). Many negated constraints and many implications lead to many explanatory and implications signs, which cannot be all shown to the pilot; due to the dimensions of the cockpit and human cognitive limits, only the most important and significant portion of the information should be displayed.

The ranking of constraints generated by the `WSMrank` function, previously introduced in the context of the `wsm decision-repair` function, is used again here to filter the explanatory and implication signs and select only the most relevant. The ranking produced by `WSMrank` is based on the evaluation criteria defined by the system designers (domain-based knowledge which includes safety and/or performance requirements). As shown above in this chapter, during the conflict repair process, the logic of `wsm decision-repair` is such that the decision constraint that scores less in the ranking produced by `WSMrank` is negated first (see Figure B.7). If a single-constraint retraction does not suffice to repair the conflict, another constraint from the conflict set is negated, the second-less-important in the WSM-raking, and so on.

The negation of these constraints represents the starting point for the construction of all the implications of the reconfiguration; in fact, the negated constraints represent a set of implications for the reconfiguration, e.g. negating a constraint that requires the activation of an application leads to the deactivation of the application in question. There is no guarantee, however, that such constraints are the most important for decision support. There could be situations in which constraints that are dependent from those negated are more important than them in terms of decision support, because, for instance, they have a greater impact on the safety of the system. To this purpose, once a conflict is repaired, all its implications are calculated (as discussed later in this section) and the `WSMrank` function is once again used to order the implication constraints, on the basis of domain-dependent knowledge codified by the system designers in the algorithm weights. For the sake of clarity, it is worth noting that for explanatory signs the `WSMrank` function is run on the constraint conflict set, whilst for the implication signs the same function is used to order to set of implications of decisions taken during the generation of the new configuration (see Figure B.7).

Unlike with the conflict set, the constraints that score *higher* in the ranking of the implications are used for decision support (i.e. the set of implication signs attached to them is shown to the pilot). Given that `WSMrank` uses domain-dependent knowledge regarding the importance of the constraints, the constraints that score higher in the ranking are the most relevant ones and, intuitively, they represent the most relevant information for the pilot.

`SaIRA-XPlain` is reported in Algorithm B.8.

The `wsm decision-repair` function returns a solution C' for the ADR problem which contains a set of negated constraints C_{neg} (note that $C_{neg} \neq \emptyset$ in any case otherwise there would be no need to call `SaIRA-XPlain`) and a partial order Ω_{C_S} over the conflict set C_S.

The `calcImplication` function calculates all the implications of the negated constraints $c_i \in$

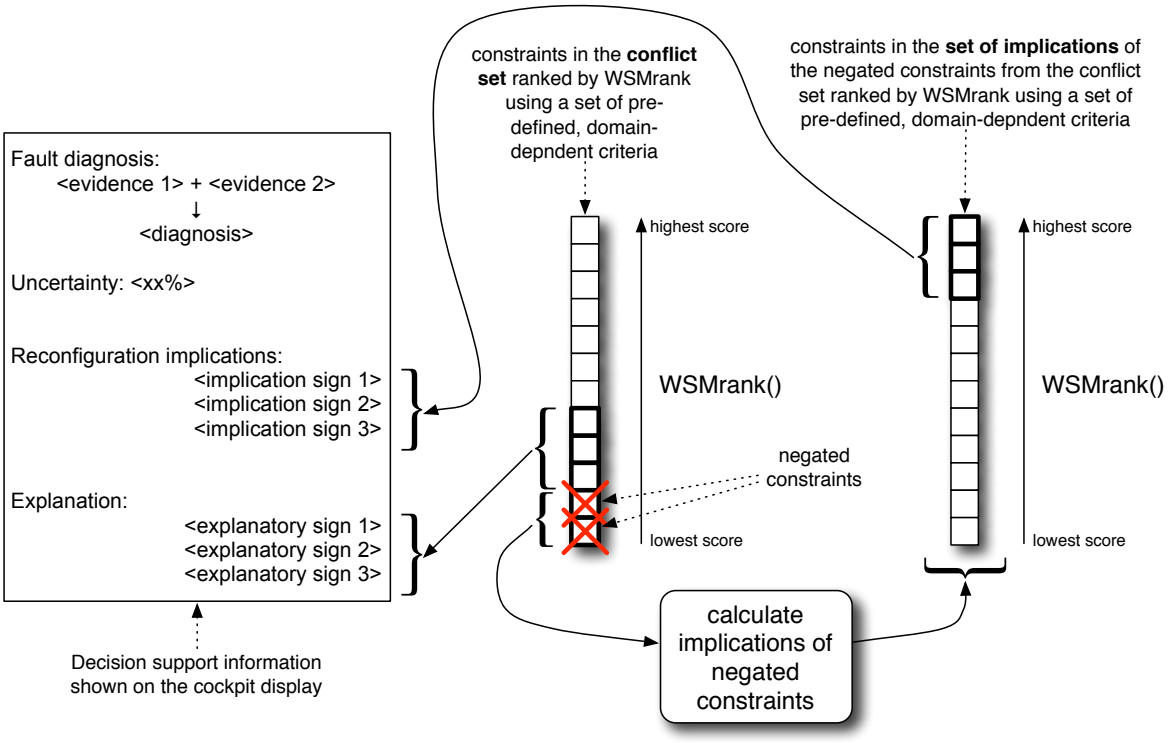


Figure B.7: `SaIRA-XPlain` uses the ranking of the decision constraints performed by the `WSMrank` function (Section B.1.5.2) during conflict repair.

Algorithm B.8 `SaIRA-XPlain()`

Require: m: current operating mode
Require: W: constraints weights (used by `WSMrank`)
Require: γ_I, γ_E: limits of number of implication/explanatory (respectively) signs that can be included in the decision support message

1: $(C', \Omega_{C_S}, C_{neg}) \leftarrow$ `wsm decision-repair`
2: $I \leftarrow$ `calcImplications`(C_{neg})
3: $\Omega_I \leftarrow$ `WSMrank`(I, W, m)
4: $I_S \leftarrow$ `implicationSigns`(Ω_I, γ_I)
5: $E_S \leftarrow$ `explanatorySigns`$(\Omega_{C_S}, C_{neg}, \gamma_E)$
6: **return** I_S, E_S

C_{neg}; it returns the set of implications I. The problem of dynamic calculation of the implications of soft constraints—including issues related to performance—have been already addressed in the literature [Freuder et al. 2001*a*; 2003]. `SaIRA-XPlain` uses the Generalised Arc Consistency `GAC3m` algorithm [Lecoutre and Szymanek 2006] which is implemented by the `infeasibleTupleAC` function of the Choco library.

The set of implications calculated is ordered by the `WSMrank` function using the pre-defined weights $w_i \in W$ and modified on the basis of the current operating mode m; the partial order Ω_I is returned.

Implication and explanatory signs for pilot support must be generated from I and C_S respectively. The number of signs shown on the display depends on the size of the display and on usability

issues which are further investigated in Chapter 6. Here, γ_I and γ_E represent the number of maximum implications and explanatory signs respectively that can be shown on the display.

The function `implicationSigns` generates a set I_S of γ_I implication signs from the partially ordered set of implication constraints Ω_I.

Slightly more complicated is the logic of the `explanatorySigns` function which produces a set E_S of γ_E explanatory signs using the partially ordered set of explanation constraints Ω_{C_S} and the set of negated decision constraints C_{neg}. The logic is as follows:

- the decision constraints in $c_i \in C_{neg} \subset C_S$ that score less when ranked by the `WSMrank` function are negated; therefore, they become part of the set of implications of the reconfiguration proposed;
- the decision constraints remaining in Ω_{C_S} 'justify' the retraction of the constraints in C_{neg}, because the former are more important (according to the domain-dependent knowledge codified in the WSM ranking);
- amongst all the constraints in Ω_{C_S}/C_{neg}, γ_E constraints must be selected to serve as basis for the construction of the explanatory signs; these constraints are those that scored 'just more' than the negated constraints when evaluated by `WSMrank`. These constraints are 'slightly more important' than those in C_{neg}, hence they are meaningful for the construction of explanations.

The overall purpose of `SaIRA-XPlain` is to perform abstraction and filtering of the basic explanations and implications, in order to provide a meaningful and helpful set of decision support information for the pilot, getting past problems with the basic approach to user-friendly explanations described in Section B.2.1.

So far, the mechanism for producing readily understandable explanations and implications of reconfiguration decisions have been described. The next section discusses the need to provide pilots with uncertainty figures (expanding the argument introduced in Chapter 4) and addresses the problem of calculating the uncertainty embedded in the fault assessment message included in the decision support information.

B.3 Limitations

Three main limitations of the approach to automated generation of ADR decision support information adopted in SaIRA have been identified.

Global constraints. This limitation pertains to a more general issue with the design of any kind of CSP, in which explanations of the decisions of the solver are to be given to the user. It has been found that the way the CSP is designed and the type of constraints used to implement the designers' knowledge can influence the number, size and effectiveness of the low-level explanations generated during the search for a solution.

More precisely, *global constraints* represent a major problem. A constraint is said to be global if "there exists no decomposition scheme for which the consistency notion removes as many local

inconsistencies as on the original constraint" [Gaudin et al. 2008]. A typical example of global constraint is the `allDifferent` constraints which are used to associate different values from a specified set with a set of variables. Intuitively, the definition of an effective, user-friendly explanation for the retraction of constraints of this type is particularly challenging (for example, how could the retraction of an `allDifferent` constraint be explained to the pilot? Such a constraint could affect dozens of variables). The use of global constraints could impair the quality of explanations generated by SaIRA during ADR.

The problem of explaining global constraints has been recently addressed by a number of researchers who devise different methodologies to mechanise the explanation process and several guidelines for the design of CSP that can be easily explained [Rochart et al. 2003; Rochart and Jussien 2007; Gaudin et al. 2008]. In the code used in this thesis, global constraints have only been used for the static part of the CSP, which does not require explanations.

Algorithm generality. Another aspect of SaIRA that requires attention is the `wsm decision-repair` algorithm. This algorithm is designed to use domain-dependent knowledge about the ADR problem, in order to improve the performance and quality of the explanations with respect to a standard version of the `decision-repair` algorithm archetype. The improvement recorded in the experiments presented in Section B.1.6 must be taken into consideration only in the context of the ADR problem and problems with a similar structure (i.e. dynamic reconfiguration of interdependent functions over a distributed network). The applicability of `wsm decision-repair` to the explanation of combinatorial problems of a different nature and structure has not been investigated in this thesis, therefore, no inferences are made in this regard.

Natural language generation. The quality of the natural language messages delivered by SaIRA relies on the ability of the designers to associate effective messages with computer-encoded constraints. Whilst a single natural language message can be associated with groups of constraints, and the messages can be organised hierarchically, the problem of associating messages with constraints can become difficult with very large CSPs, especially when a hierarchical structure cannot be easily defined (even though it was proved that it is always possible to do so [Jussien 2003]). A potential area for further research on SaIRA is the possibility of using more sophisticated natural language generation techniques to enable the system to automatically produce the explanatory and implication signs directly from the computer-encoded explanations and implications. The generation of natural language to explain problems based on large, numerical data-sets has been already addressed in the literature (e.g. [Goldberg et al. 1994]); there seem to be no reason why similar technology could not be applicable to the ADR problem.

Appendix C

Evidential Reasoning

C.1 Against Bayesian Reasoning in SaIRA

Suppose D is the diagnosis of an unexpected event (e.g. a fault), E_1 and E_2 are two pieces of evidence from sensors s_1 and s_2. The classic Bayesian Rule for information fusion is:

$$P(H|E_1, E_2) = \frac{P(E_1, E_2|H)\ P(H)}{P(E_2, E_1)} \tag{C.1}$$

Assuming that E_1 and E_2 are independent, we have:

$$\begin{aligned} &P(E_1, E_2) = P(E_1)\ P(E_2) \\ &P(E_2|H, E_1) = P(E_2|H) \\ &P(E_1, E_2|H) = P(E_1|H)\ P(E_2|H, E_1) \end{aligned} \tag{C.2}$$

We can rewrite $P(H|E_1, E_2)$ as:

$$P(H|E_1, E_2) = \frac{P(E_1|H)\ P(E_2|H)\ P(H)}{P(E_1)\ P(E_2)} \tag{C.3}$$

where $P(H|E_1, E_2)$ is the *a posteriori* probability after considering two pieces of evidence E_1 and E_2; $P(H)$, $P(E_1)$ and $P(E_2)$ are *a priori* probabilities; $P(E_1|H)$ and $P(E_2|H)$ are *conditional* probabilities of the belief in E_1 and E_2 when H is considered, and $P(E_2)P(E_1)$ is a *normalization factor*.

The Bayesian approach makes some strong assumptions about the operating scenario that cannot be satisfied in ADR. Since the operating scenario of SCMS considered in this thesis (e.g. aircraft, spacecraft) is unstructured, it is not possible to reproduce it in a laboratory. As a consequence, $P(H)$ is usually very difficult to estimate, e.g. the probability of occurrence of any fault. Furthermore, when more than one sensor are used to make inferences on a specific unexpected event, the conditional probabilities (e.g. $P(E_1|H)$ and $P(E_2|H)$) are usually unavailable. These probabilities are strongly affected by the contingent operating conditions.

Whilst allowing updating a priori probabilities at run-time, Dempster-Shafer (DS) theory relaxes the Bayesian restriction on mutually exclusive hypothesis so that it is possible to assign evid-

ence to propositions (i.e. union of hypotheses). This mechanism allows specification of partial knowledge. Pieces of knowledge and uncertainty are processed. At the end of the inference process, the pilot can be provided with information about how uncertain are the conclusions achieved by the system. The pilot has the freedom of either trusting or not trusting the system; in both cases the choice will be more informed than one made accepting conclusions that hide degrees of uncertainty.

C.2 General Background about Evidential Reasoning

Evidential Reasoning (ER) is a reasoning framework developed by SRI International and funded by DARPA. It is based on the *theory of belief functions* conceived by Dempster [1967] and further developed by Shafer [1976]. It is a generalisation of probability theory that allows specification of *degrees of precision* as well as *degrees of uncertainty*.

The goal of ER is to assess the effect of all available pieces of evidence upon a hypothesis by making use of domain-specific knowledge. Bodies of evidence are expressed as probabilistic opinions about the partial truth or falsity of statements composed of subsets of propositions from a space of distinct and exhaustive possibilities (called the *frame of discernment*).

The distribution of beliefs over a frame of discernment is called a *body of evidence* (BOE). Several formal methods are provided to fuse (i.e. pooling) two BOE (see Sentz et al. [2002] for an exhaustive and critical review). The result is a new BOE representing the consensus of the two original BOE.

The theory allows belief to be assigned to individual propositions in the space or to disjunctions of propositions or both. Belief assigned to a disjunction explicitly represents a lack of sufficient information to enable more precise distribution. This allows belief to be attributed to statements whose granularity is appropriate to the available evidence.

Independent beliefs are expressed by multiple (independent) BOE; dependent beliefs (in which belief in one proposition depends on that of another) can either be expressed by a single BOE or by a network that describes the inter-relationships among several BOE.

Given its singular capacity to define uncertainty/ignorance embedded in the input data and use it to make inferences, ER is proposed for the handling the of uncertain sensor information in SaIRA and for the generation of uncertainty figures in support of pilots' decision making activities.

There are three important functions in Dempster-Shafer theory which must be put into context in SaIRA: the *basic probability assignment function* (bpa or m), the *support* (or *belief*) function (Spt), and the *plausibility* function (Pls). Each of them is introduced and discussed in the following sections.

C.2.1 Basic probability assignment function

A BOE is represented by means of a *basic probability function* (bpa). A bpa (m), also known as *mass*, is a set mapping from subsets of a frame of discernment, Θ, into the unit interval:

$$m_\Theta = 2^\Theta \rightarrow [0, 1] \tag{C.4}$$

such that:

$$m_\theta(\phi) = 0 \quad \text{and} \quad \sum_{\substack{X \subseteq \Theta \\ X,Y \subseteq \Theta}} m_\Theta(X) = 1 \tag{C.5}$$

Generally speaking, the term "basic probability assignment" does not refer to probability in the classical sense. The bpa defines a mapping of the power set to the interval between 0 and 1, where the bpa of the null set is 0 and the summation of the bpa of all the subsets of the power set is 1. The value of the bpa for a given set X (represented as $m(X)$), expresses the proportion of all relevant and available evidence that supports the claim that a particular element of Θ (the universal set) belongs to the set X but to no particular subset of X [Klir 1999].

The value of $m(X)$ pertains only to the set X and makes no additional claims about any subsets of X. Any further evidence on the subsets of X would be represented by another bpa, i.e. $Y \subset X$, $m(Y)$ would the bpa for the subset Y.

C.2.2 Support and Plausibility functions

Any proposition that has been attributed nonzero mass is called a *focal element*. One of the ramifications of this representation of belief is that the probability of a hypothesis X is constrained to lie within an interval $[Spt(X), Pls(X)]$ where:

$$Spt(X) = \sum_{Y \subseteq X} m_\Theta(Y) \tag{C.6a}$$

$$Pls(X) = 1 - Spt(\overline{Y}) \tag{C.6b}$$

$$[Spt(X), Pls(X)] \subseteq [0, 1] \tag{C.6c}$$

$Spt(X)$ is referred to as *support* (or *belief*) and indicates the degree to which the evidence supports the proposition. $Pls(X)$ is referred to as *plausibility* and indicates the degree to which the evidence fails to refute the proposition (or, in other words, the degree to which it remains plausible).

We refer to the frame of discernment as Θ. As a consequence, (C.6b) can be rewritten as:

$$Pls(X) = 1 - Spt(\theta - Y) \tag{C.7}$$

Complete ignorance is represented as an interval of $[0.0, 0.1]$. Other degrees of ignorance (or knowledge) are captured by intervals like $[0.5, 0.7]$, $[0.8, 0.4]$, $[0.9, 1.0]$

As mentioned before, several formal methods are provided to fuse two BOE. The Dempster rule [Shafer 1976] is the most commonly used. However, SaIRA uses a modification of this rule developed by Yager [1987]. They are both briefly introduced in the next two sub-sections with the support of an intuitive explanatory ADR scenario. The rationale for using the Yager's rule instead of Dempster's is also explained.

C.2.3 Dempster's rule

Dempster's rule is the most common belief combination rule. It combines multiple belief functions b_n through their basic probability assignments (m_{ik}).

These belief functions are defined on the same frame of discernment, but are based on *independent* arguments or bodies of evidence. Argument independence is a critical factor when combining evidence within the ADR problem which makes this rule inappropriate. In fact, as will be shown later, BOEs can be interdependent in the ADR problem. Since Yager's rule is a modification of the Dempster's that relaxes the constraint of interdependence, we briefly describe Dempster's rule here and then introduce the Yager's.

Dempster's rule is purely conjunctive (AND). The combination m_{12} is calculated from aggregation of basic probability assignments m_1 and m_2 in the following manner:

$$m_{12}(X) = \frac{\sum_{Y \cap Z = X} m_1(Y) m_2(Z)}{1 - K} \tag{C.8}$$

when:

$$X \neq \emptyset$$
$$m_{12}(\emptyset) = 0$$

and where:

$$K = \sum_{Y \cap Z = \emptyset} m_1(Y) m_2(Z) \tag{C.9}$$

K represents the basic probability mass associated with the *conflict* between the BOEs Y and Z. This is determined by summing the products of the bpas of all sets where the intersection is null. This rule is commutative, associative, but not idempotent or continuous.

The denominator in Dempster's rule, 1-K, is a normalization factor. This has the effect of completely ignoring conflict and attributing any probability mass associated with conflict to the null set. This is not exactly what is required in SaIRA. Consider the following example:

> A sudden and severe lack of power is detected by the Health Monitoring Unit (HMU). There are three possible causes: *a*) a fault to the left engine power generator; *b*) a fault to the right engine power generator; *c*) a fault to the APU. A set of sensors attributes this event mainly to a with support $Spt_1 = (m(\{a\}) = 0.80, m(\{c\}) = 0.20)$. Another set of sensors attributes this event mainly to b with support $Spt_2 = (m(\{b\}) = 0.80, m(\{c\}) = 0.20)$.

As highlighted by Zadeh [1979], *strongly contradictory evidence* like the example above, when combined with the Dempster combination rule, can produce incoherent results. In this specific example, applying (C.8), the resulting diagnosis is c, a fault to the APU (which is wrong!). The reason for this is that contradictory evidence cancels out one another and the cause c "wins" even if it was given a lower support with respect to a and b.

This should never happen in SaIRA because, with reference to this specific example, contradictory evidence means that the HMU doesn't know enough about the fault in question so its diagnosis is not 100% reliable; this eventuality, as just seen, can potentially cause more harm than the fault itself.

C.2.4 Yager's rule

Ronald R. Yager makes an important distinction between the basic probability mass assignment (m) and what he refers to as the *ground probability mass* assignment (designated by q). The major differences between the basic probability assignment and the ground probability assignment are in the normalization factor and the mass attributed to the universal set. The combined ground probability assignment is defined as follows:

$$q(X) = \sum_{Y \cap Z = X} m_1(Y) m_2(Z) \tag{C.10}$$

where X is the intersection of subsets Y and Z (both in the power set $\mathbb{P}(\Theta)$), and $q(X)$ denotes the ground probability assignment associated with X. This rule is known as Yager's combination rule or sometimes as the Modified Dempster's Rule.

Note that there is no normalization factor $(1 - K)$. In Yager's formulation, he circumvents normalization by allowing the ground probability mass assignment of the null set to be greater than 0:

$$q(\emptyset) \geq 0 \tag{C.11}$$

$q(\emptyset)$ is calculated in exactly the same manner as Dempster's K (conflict) in Equation C.8. Then Yager adds the value of the conflict represented by $q(\emptyset)$ to the ground probability assignment of the universal set, $q(X)$, to yield the conversion of the ground probabilities to the basic probability assignment of the universal set $m^Y(X)$:

$$m^Y(X) = q(X) + q(\emptyset) \tag{C.12}$$

Consequently, instead of normalizing out the conflict, as we find in the case of Dempster's rule (C.8), Yager ultimately attributes conflict to the universal set X.

The interpretation of the mass of the universal set (X) is the degree of *ignorance*. Dempster's rule has the effect of changing the evidence through the normalization and the allocation of conflicting mass to the null set. Yager's rule can be considered as an epistemologically honest interpretation of the evidence as it does not change the evidence by normalizing out the conflict.

In other words, in Yager's rule, the mass associated with conflict is attributed to the universal set and thus *enlarges the degree of ignorance*. This solves the problems of applying Dempster's rule in SaIRA and provides an elegant way to extract information about the ignorance that characterizes the reasoning process that we were looking for in the previous section.

C.3 Problem Modeling

Let $E = \{E_1, E_2, ..., E_n\}$ be the set of possible events that require ADR, and E_i mean that the type of a given event is E_i.

A frame of discernment $\Theta = \{E_i, \neg E_i\}$ is the set of problem diagnoses under evaluation. For instance, with reference to Accident 5.1 previously introduced in section 5.5, given the fault `fuel-leak` (it is referred to as `FL` hereinafter), the following bpa is defined:

$$m(\{\texttt{FL}\}) + m(\{\neg\texttt{FL}\}) + m(\{\texttt{FL}, \neg\texttt{FL}\}) = 1 \tag{C.13}$$

where $\{\texttt{FL}, \neg\texttt{FL}\}$ is the ignorance set.

As explained in section C.2, the sum of the bpa of the singleton subsets of Θ may be less than 1, representing the fact that there is ignorance/uncertainty in the inference process. For instance, given $m(\{\texttt{FL}\}) = 0.7$, $m(\{\neg\texttt{FL}\}) = 0$, it follows that $m(\{\texttt{FL}, \neg\texttt{FL}\} = 0.3$ which is less than 1 (30% ignorance). This could be that case in which two sensors diagnose `FL`, no sensors excludes the fuel leak ($\neg$`FL`) and other sensors diagnose some other fault.

For a subset K of Θ, the support function *Spt* is defined as the sum of the beliefs committed to the possibilities in K. For example,

$$Spt(\{\texttt{FL}, \neg\texttt{FL}\}) = m(\{\texttt{FL}\}) + m(\{\neg\texttt{FL}\}) + m(\{\texttt{FL}, \neg\texttt{FL}\}) = 1 \tag{C.14}$$

For individual members of Θ (in this case `FL` and $\neg$`FL`), *Spt* and *m* are equal. As a consequence,

$$Spt(\{\texttt{FL}\}) = m(\{\texttt{FL}\}) = 0.7 \tag{C.15}$$

$$Spt(\neg\{\texttt{FL}\}) = m(\neg\{\texttt{FL}\}) = 0 \tag{C.16}$$

As anticipated in section C.2, in SaIRA pieces of evidence are combined by means of Yager's combination rule. It is worth recalling that this rule is both associative and commutative, which means that the process of combining evidence from multiple sensors is independent of the order in which the sensors outputs are combined.

C.3.1 Confusion Sets

Whilst investigating the problem of sensors fusion for forces aggregation and classification in a battlefield, Yu et al. [2004] introduced the notion of *confusion sets* with the aim of efficiently aggregating the information produced by different sensors.

Putting Yu's notion of confusion set into the context of SaIRA, *a confusion set is a set of possible events, where the sensor S_i may confuse one with another when we use the same sensor to diagnose the event in question.*

For example, with reference to the example used in section 5.5, at the occurrence of a fuel leak (the evidence are oil temperature increase and oil pressure decrease), oil pressure and temperature sensors could diagnose a `FL`; fuel valves sensors could diagnose a `FL` or a fault to the Fuel Cross-

feed Valve[1] (`FCV-fault`); however, they will usually not confuse these kind of unexpected events with a fault to the Fuel Vent System (`FV-fault`), which is of a completely different nature.

Tables C.1 and C.2 show the output of two different types of sensors for a `FL` fault. Let $E = \{E_1, E_2, ..., E_n\}$ be the set of possible event types; a *confidence level* $c(E_i)$ is associated to each event type E_i. From this assumption, Yu et al. formally define a confusion set as follows (adapted to the ADR problem from the original definition of Yu et al. which was specific to vehicles targeting on a battlefield):

Definition C.3.1 *Let $C = \{E_1, E_2, ..., E_m\}$ $(m < n)$ be a subset of E and C sorted by the confidence levels, e.g. $c(V_1) \geq c(V_2) ... \geq c(V_m)$, C is a confusion set if and only if (1) for any event type $E_i \in C$, $E_j \in E$ and $E_j \notin C$, $c(E_i) \geq \sigma > c(E_j)$, where σ is the identification threshold; (2) for any event type $E_i \in C$, $1 \leq i \leq m-1$, $c(E_i) - c(E_{i+1}) \leq \rho$, where ρ is the minimum distance of confidence levels.*

Oil temperature and pressure sensors	
Event	*Confidence Level*
`FL`	0.3
`FCV-fault`	0.6
`FV-fault`	0.1

Table C.1: List of possible events that can be detected by the *oil temperature and pressure sensors* and relative confidence levels

Fuel Valves Sensors	
Event	*Confidence Level*
`FL`	0.7
`FCV-fault`	0.2
`FV-fault`	0.1

Table C.2: List of possible events that can be detected by the *Fuel Valves Sensors* and relative confidence levels

A confusion set is sensor specific and it is dynamically chosen on the basis of the confidence levels and the relative distances between the confidence levels. In fact, in a multi-sensor network, depending on the characteristics of the sensors, some of them *cooperate* to identify an event, others do not so because they are not designed to reason on that specific event.

For instance, on a Boeing 737, both the oil pressure and oil temperature sensors installed on the left central tank are suitable to detect faults related to valves of that tank but, most probably, they are not suitable to diagnose faults to the valves controlled by the Fuel Control Unit of the Auxiliary Power Unit (APU) which is located at the very back of the aircraft. This is a gross example which anyway clarifies the concept in question.

At the occurrence of an event E_i, all sensors S_i for which $E_j \in C_i$ (we choose the sensors that are suitable to reason about the event in question) are selected. For each sensor the frame of

[1] On a Boeing 737, "continued fuel crossfeed use will result in a progressive fuel imbalance" [The Boeing Company 2002]

discernment for which the sensor has a higher confidence level is selected. Formally, an active frame of discernment is defined as follows:

Definition C.3.2 *Let* $C = \{E_1, E_2, ..., E_m\}$, $(m < n)$ *is a confusion set,* $c(E_i)$ *is the confidence level of event* E_i, $(1 \leq i \leq m)$*, a frame of discernment* $\Theta_k = \{E_i, \neg E_i\}$ *is* active *frame of discernment for* C *if and only if for any event type* $E_j \in C$, $c(E_i) \geq c(E_j)$.

Consider the sensors S_1 and S_2 with their confusion sets C_1 and C_2 and their active frames of discernment $\Theta_{1i} = \{E_1, \neg E_1\}$ and $\Theta_{2j} = \{E_2, \neg E_2\}$ respectively. The following three possibilities are available, which are handled in different ways with respect to how the pieces of evidence involved are fused:

1. $\Theta_{1i} = \Theta_{2j}$ (C_1 and C_2 have the same frame of discernment). The *Spt* functions are combined directly using Yager's rule.

2. $\Theta_{1i} \neq \Theta_{2j}$ but $C_1 \cap C_2 \neq \emptyset$. C_1 and C_2 have different frames of discernment but there is some intersection between the confusion sets. This means the two sensors are not exactly 'tuned' for the same set of events but they can cooperate when the inferences concern the events they both cover (this is the case of the oil pressure and temperature sensors of the previous example). The maximum between $m_{i1}(\{E_1\})$ and $m_{2j}(\{E_2\})$ is chosen. Then, (a) if $E_2 \in C$, $\Theta_{1k} = \{E_2, \neg E_2\}$ is selected as the active frame of discernment for C_1. The *Spt* function for the reselected frame of discernment Θ_{1k} is combined with the *Spt* function for Θ_{2k}; (b) otherwise the outputs cannot be fused and the *Spt* function for Θ_{2j} is used as the fused result for sensors S_1 and S_2.

3. $\Theta_{1i} \neq \Theta_{2j}$ and $C_1 \cap C_2 = \emptyset$. C_1 and C_2 have different frames of discernment and there is no intersection between the confusion sets. The combination of conflicting information is avoided and the Spt function for Θ_{2j} is used as the fused result for sensors S_1 and S_2.

Bibliography

Abecker, A. and Elst, L. [2009], 'Ontologies for knowledge management', *Handbook on ontologies*, Springer, **1**, 713—-734.

ACM TIST [2010], Social Recommender Systems, *in* Q. Yang, ed., 'ACM Transactions on Intelligent Systems and Technology', Association for Computational Machinery, Hong Kong, p. 1.

Adomavicius, G. and Tuzhilin, A. [2005], 'Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions', *IEEE transactions on knowledge and data engineering*, IEEE, **17**(6), 734–749.

Aeronautica Civil Of The Republic Of Colombia [1996], Controlled Flight Into Terrain, American Airlines Flight 965, Boeing 757-223, N651AA, Near Cali, Colombia, December 20, 1995, Technical report.

Aeronautical Radio Incorporated (ARINC-653) [2006], 'ARINC 653-2 Standard'.
URL: *https://www.arinc.com*

Amalberti, R. [1999], 'Automation in aviation: A human factors perspective', *Handbook of aviation human factors*, Lawrence Erlbaum Associates Mahwah, NJ, **1**(1), 1–18.

Amilhastre, J., Fargier, H. and Marquis, P. [2002], 'Consistency restoration and explanations in dynamic CSPs - Application to configuration', *Artificial Intelligence*, Elsevier, **135**(1-2), 199–234.

Anderson, C. and Sechler, E. [1986], 'Effects of explanation and counterexplanation on the development and use of social theories.', *Journal of Personality and Social Psychology*, **50**(1), 24–34.

Andrews, P. [2008], Persuasive Computer Dialogue Improving Human-Computer Communication, PhD thesis, The University of York.

Arshad, N. [2003], Dynamic Reconfiguration of Software Systems using Temporal Planning, *in* 'Tools with Artificial Intelligence', Vol. 1, IEEE, Sacramento, California, USA, pp. 39–46.

Arthur, J., Prinzel, L., Kramer, L., Bailey, R. and Parrish, R. [2003], CFIT prevention using synthetic vision, *in* 'Proceeding of the International Society for Optical Engineering (SPIE) Annual Meeting', Vol. 5081, SPIE Publications, pp. 146–157.

Artman, H. [1999*a*], 'Cooperation and situation awareness within and between time scales in dynamic decision-making', *Ergonomics*, Taylor & Francis Group, **42**(11), 1404–1417.

Artman, H. [1999*b*], 'Situation awareness and co-operation within and between hierarchical units in dynamic decision making', *Ergonomics*, Taylor & Francis, **42**(11), 1404–1417.

Aviation Safety [1993], 'Flight 2904'.
URL: *http://aviation-safety.net/database/record.php?id=19930914-2*

Aviation Safety Network [1993], 'Lufthansa Flight 2904'.
URL: *http://aviation-safety.net/database/record.php?id=19930914-2*

Aviation Safety Network [2009], 'Turkish Airlines Flight 1951'.
URL: *http://aviation-safety.net/database/record.php?id=20090225-0*

Aviation-safety.net [2010], 'Aircraft accident - Air Transat Flight 236'.
URL: *http://aviation-safety.net/database/record.php?id=20010824-1*

Avizienis, A., Lapries, J. and Randell, B. [2001], Dependability of Computer Systems: Fundamental Concepts, Terminology and Examples, Technical report, University of Newcastle Upon Tyne, Dept of Computer Science, Newcastle Upon Tyne.

Babylon Dictionary [2010], 'Babylon Dictionary Online'.
URL: *www.babylon.com*

Bahill, A. and Dean, F. [1999], Discovering System Requirements, *in* AP Sage and WB Rouse, ed., 'Handbook of systems engineering and management', Wiley-Interscience, New York, chapter 4, pp. 175–220.

Balabanović, M. and Shoham, Y. [1997], 'Fab: content-based, collaborative recommendation', *Communications of the ACM*, Association for Computational Machinery, **40**(3), 66–72.

Banbury, S., Selcon, S., Endsley, M., Gorton, T. and Tatlock, K. [1998], Being certain about uncertainty: How the representation of system reliability affects pilot decision making, *in* Human Factors and Ergonomics Society, ed., 'Proceedings of the Human Factors and Ergonomics Society Annual Meeting', Vol. 42, SAGE Publications, Chicago, pp. 36–39.

Bass, E., Small, R. and Ernst-Fortin, S. [1997], Knowledge Requirements and Architecture for an Intelligent Monitoring Aid that Facilitate Incremental Knowledge Base Development, *in* W. D. Potter, M. Matthews and M. Ali, eds, 'Industrial and engineering applications of artificial intelligence and expert systems: proceedings of the tenth international conference', Taylor & Francis, Atlanta, Georgia, p. 63.

Bate, I. [2003], 'Architectural considerations in the certification of modular systems', *Reliability Engineering & System Safety*, **81**(3), 303–324.

Bate, I. and Burns, A. [2003], 'An integrated approach to scheduling in safety-critical embedded control systems', *Real-Time Systems*, **653**(1996), 5–37.

Bate, I., Hawkins, R. and Mcdermid, J. [2003], A Contract-based Approach to Designing Safe Systems, *in* P. Lindsay and P. Cant, eds, 'Proceedings of the 8th Australian workshop on Safety critical systems and software', Vol. 33, Australian Computer Society, Inc., p. 36.

Baxter, G., Besnard, D. and Riley, D. [2007], 'Cognitive mismatches in the cockpit: Will they ever be a thing of the past?', *Applied Ergonomics*, **38**(4), 417–423.

Baxter, G. and Ritter, F. [1999], 'Towards a classification of state misinterpretation', *Engineering psychology and cognitive ergonomics.*, **3**, 35–42.

Beach, L. [1993], 'Broadening the definition of decision making: The role of prechoice screening of options', *Psychological Science*, **4**, 215.

Beach, L. [1998], *Image theory: Theoretical and empirical foundations*, Lawrence Erlbaum, Mahwah, New Jersey London.

Bedny, G. and Meister, D. [1999], 'Theory of activity and situation awareness', *International Journal of cognitive ergonomics*, Lawrence Erlbaum, **3**(1), 63–72.

Bélanger, M. and Martel, J.-m. [2005], 'An automated explanation approach for a decision support system based on mcda', *Explanation-Aware Computing: Papers from the 2005 Fall Symposium. Technical Report FS-05-04*, **5**, 04.

Belz, L. [2005], 'EADS sensor fusion software supports warning & control aircraft'.
URL: *http://www.eads.com/1024/en/pressdb/archiv/2005/*

Ben Zur, H. and Breznitz, S. J. [1981], 'The effects of time pressure on risky choices', *Acta Psychologica*, North-Holland Publishing Company, **47**(1), 89–104.

Benders, J. [1962], 'Partitioning procedures for solving mixed-variables programming problems', *Numerische Mathematik*, Springer, **4**(1), 238–252.

Benson III, L. and Beach, L. [1998], The effects of time constraints on the prechoice screening of decision options, *in* L. R. Beach, ed., 'Image theory: theoretical and empirical foundations', 1 edn, Routledge, Oxford, chapter 3, p. 51.

Besnard, D. [2004], 'When mental models go wrong: co-occurrences in dynamic, critical systems', *International Journal of Human-Computer Studies*, **60**(1), 117–128.

Bessiere, C. [1991], Arc-consistency in dynamic constraint satisfaction problems, *in* D. Bobrow, ed., 'Proceedings of the Association for the Advancement of Artificial Intelligence AAAI-91', Vol. 1, AAAI Press, Anaheim, California, pp. 179–190.

Bilgic, M. and Mooney, R. [2005], 'Explaining recommendations: Satisfaction vs. promotion', *Beyond Personalization 2005*.

Billings, C. [1997], 'Aviation Automation, The search for the Human-Centered Approach', *Lawrence Erlbaum*.

Billings, C., Lauber, J., Funkhouser, H., Lyman, G. and EM [1976], NASA aviation safety reporting system, Technical report, NASA Ames Research Center, Moffett Field, CA.

Billsus, D. and Pazzani, M. [1999], 'A hybrid user model for news story classification', *Courses and Lectures - International Centre for Mechanical Sciences*, Springer Verlag, **1**(1), 99–108.

Blackwell, N., Leinster-Evans, S. and Dawkins, S. [1999], 'Developing safety cases for integrated flight systems', *1999 IEEE Aerospace Conference. Proceedings (Cat. No.99TH8403)*, Ieee, pp. 225–240 vol.5.

Blair, C. [1983], 'Nietzsche's" Lecture Notes on Rhetoric": A Translation', *Philosophy & Rhetoric*, JSTOR, **16**(2), 94–129.

Bliek, C. [1998], Generalizing partial order and dynamic backtracking, *in* J. Mostow and C. Rich, eds, 'Proceedings of the National Conference on Artificial Intelligence', Vol. 1, John Wiley & Sons Ltd, Madison, Wisconsin, pp. 319–325.

Bogacz, R., Brown, E., Moehlis, J., Cohen, J. and Holmes, P. [2006], 'The physics of optimal decision making: A formal analysis of models of performance in two-alternative forced-choice tasks.', *Psychological Review*, **113**(4), 700—-765.

Boutilier, C., Brafman, R., Geib, C. and Poole, D. [1997], 'A constraint-based approach to preference elicitation and decision making', *AAAI Spring Symposium on Qualitative Decision Theory*.

Brady, C. [1999], 'www.b737.org.uk'.

Braver, T., Cohen, J., Nystrom, L. and Jonides, J. [1997], 'A parametric study of prefrontal cortex involvement in human working memory.', *NeuroImage*, Orlando Academic Press, **5**(1), 49–62.

Brehmer, B. [1990], 'Strategies in real-time, dynamic decision making', *Insights in decision making: A tribute to Hillel J. Einhorn*, pp. 262–279.

Bresina, J. and Morris, P. [2006], 'Explanations and Recommendations for Temporal Inconsistencies', *Proceedings of the 5th International Workshop on Planning and Scheduling for Space, IWPSS'06*, **3**(1), 1–5.

Brewka, G. [1989], Preferred subtheories: An extended logical framework for default reasoning, *in* 'Proceedings of the Eleventh International Joint Conference on Artificial Intelligence', Elsevier, pp. 1043–1048.

Brunswik, E. [1947], *Systematic and representative design of psychological experiments*, Univ. of California Press.

Bubb-Lewis, C. and Scerbo, M. [1997], 'Getting to know you: Human-computer communication in adaptive automation', *Human-automation interaction: Research and practice*, pp. 92—-99.

Buchanan, B. and Shortliffe, E. [1984], *Rule-based expert systems: the MYCIN experiments of the Stanford Heuristic Programming Project*, Addison-Wesley Reading, MA.

Burke, R. and Hammond, K. [1996], 'Knowledge-based navigation of complex information spaces', *In Proceedings of the 13th National Conference On Artificial Intelligence*, **462**, 468.

Burke, R., Hammond, K. and Yound, B. [1997], 'The FindMe approach to assisted browsing', *IEEE Expert*, IEEE, **12**(4), 32–40.

Bustamante, E., Madhavan, P., Wickens, C., Parasuraman, R., Manzey, D., E.J., B.-H., Meyer, J., Bliss, J., Lee, J. and Rice, S. [2009], Current Concepts and Trends in Human-Automation Interaction, *in* 'Human Factors and Ergonomics Society Annual Meeting Proceedings', Human Factors and Ergonomics Society, pp. 299–303.

Caires, G. [2001], 'F-22 Raptor Achieves Key Program Criteria - First Flight of Raptor 4005 Armed with Block 3.0 Avionics'.
URL: *www.lockheedmartin.com*

Caires, G. and Stout, J. [2002], 'Newest Advanced Integrated Avionics Software Package Flown for First Time Aboard the F-22 Raptor Air Dominance Fighter'.
URL: *www.lockheedmartin.com*

Cairns, P. and Cox, A. [2008], *Research methods for human-computer interaction*, first edit edn, Cambridge University Press New York, NY, USA, Cambridge.

Calderwood, R., Crandall, B., Klein, G. and OH, K. A. I. Y. S. [1987], Expert and novice fire ground command decisions (Technical Report under Contract MDA903-85-C-0327), Technical report, U.S. Army Research Institute, Alexandria, VA.

Cambazard, H., Hladik, P., Déplanche, A., Jussien, N. and Trinquet, Y. [2004], 'Decomposition and learning for a hard real time task allocation problem', *Principles and Practice of Constraint Programming CP2004*, Springer, pp. 153–167.

Campbell, R. and Bagshaw, M. [2002], *Human performance and limitations in aviation*, 3 edn, Wiley-Blackwell, Oxford.

Cannon-Bowers, J. and Salas, E. [1998], 'Team performance and training in complex environments: Recent findings from applied research', *Current Directions in Psychological Science*, JSTOR, **7**(3), 83–87.

Cao, A., Chintamani, K. K., Pandya, A. K. and Ellis, R. D. [2009], 'NASA TLX: software for assessing subjective mental workload.', *Behavior research methods*, **41**(1), 113–7.

Carlsson, M., Kreuger, P. and Astrom, E. [1998], 'Constraint-based resource allocation and scheduling in steel manufacturing', *Practical Aspects of Declarative Languages*, Springer, pp. 335–349.

Caseau, Y., Josset, F. and Laburthe, F. [2002], 'CLAIRE: Combining sets, search and rules to better express algorithms', *Theory and Practice of Logic Programming*, Cambridge Univ Press, **2**(06), 769–805.

Chen, J. and Lee, S. [2003], 'An exploratory cognitive DSS for strategic decision making', *Decision Support Systems*, Elsevier, **36**(2), 147–160.

Cheng, C. and Smith, S. [1997], 'Applying constraint satisfaction techniques to job shop scheduling', *Annals of Operations Research*, Springer, **70**, 327–357.

Christensen-Szalanski, J. and Beach, L. [1984], 'The citation bias: Fad and fashion in the judgment and decision literature.', *American Psychologist*, **39**(1), 75–78.

Chun, H. W., Avenue, T. C. and Kong, H. [1997], Constraint-based resource allocation for air cargo transfer planning, *in* 'Industrial and engineering applications of artificial intelligence and expert systems: proceedings of the tenth international conference, Atlanta, Georgia, USA, June 10-13, 1997', number June, Taylor & Francis, Atlanta, Georgia, p. 161.

Coch, J. [1996*a*], 'Evaluating and comparing three text-production techniques', *Proceedings of the 16th conference on Computational*, Association for Computational Linguistics, **1**(1), 249.

Coch, J. [1996*b*], Overview of AlethGen, *in* 'Demonstrations and Posters of the Eighth International', Vol. 1, Springer Verlag, Herstmonceux, Sussex, pp. 25–28.

Cohen, M. and Freeman, J. [1997], Training the naturalistic decision maker, *in* 'Naturalistic decision making', Lawrence Erlbaum, pp. 257—-268.

Conmy, P. and McDermid, J. [2001], High level failure analysis for Integrated Modular Avionics, *in* 'Sixth Australian workshop on Safety critical systems and software', Vol. 3, ACM Press, St Lucia, Queensland, pp. 13–21.

Cook, G. and Swain, M. [1993], 'A Computerized Approach to Decision Process Tracing for Decision Support System Design*', *Decision Sciences*, **24**, 931–952.

Cook, M., Noyes, J. and Masakowski, Y. [2007], *Decision making in complex environments*, Ashgate Publishing.

Cooper, G. and Harper, R. [1969], The use of pilot rating in the evaluation of aircraft handling qualities (NASA Ames Technical Report NASA TN-D-5153), Technical report, NASA Ames Research Center.

Corbetta, M. and Shulman, G. [2002], 'Control of goal-directed and stimulus-driven attention in the brain', *Nature Reviews Neuroscience*, **3**(3), 201–215.

Coutinho, R. M. a. [2008], 'Aspects on Architecture for Independent Distributed Avionics (AIDA)', *2008 IEEE/AIAA 27th Digital Avionics Systems Conference*, Ieee, pp. 1.A.1–1–1.A.1–9.

Cowan, N. [2001], 'The magical number 4 in short-term memory: A reconsideration of mental storage capacity', *Behavioral and brain sciences*, **24**(1), 87–114.

Cox, G., Sharples, S., Stedmon, A. and Wilson, J. [2007], 'An observation tool to study air traffic control and flightdeck collaboration.', *Applied ergonomics*, **38**(4), 425–35.

Cummings, M. [2006], 'Automation and accountability in decision support system interface design', *J. Technology Studies*, Citeseer, **32**(1), 23–31.

Dadashi, N., Wilson, J. R., Sharples, S., Golightly, D. and Clarke, T. [2011], 'A framework of data processing for decision making in railway intelligent infrastructure', *Human Factors*, pp. 276–283.

Dahlen, B., Konstan, J., Herlocker, J., Good, N., Borchers, A. and Riedl, J. [1998], 'Jump-starting movielens: User benefits of starting a collaborative filtering system with 'dead data'', *University of Minnesota TR*, pp. 98–017.

Dale, R. and Reiter, E. [1995], 'Building natural language generation systems', *Studies in Natural Language Processing. Cambridge University Press*.

Dao, A., Brandt, S., Battiste, V. and Vu, K. [2009], 'The Impact of Automation Assisted Aircraft Separation on Situation Awareness', *Proceedings of the Symposium on Human Interface 2009 on Human Interface and the Management of Information. Information and Interaction. Part II: Held as part of HCI International 2009*, p. 747.

Dawes, R. [1986], 'Proper and improper linear models', *International Journal of Forecasting*, **2**, 5—-14.

De Givry, S., Jeannin, L., Josset, F.-x., Mattioli, J., Museux, N. and Savéant, P. [2002], 'The THALES constraint programming framework for hard and soft real-time applications', *The PLANET Newsletter*, Citeseer, **5**, 1610–0212.

Dekker, S. [2000], Human Factors in Aviation-A natural history, *in* 'Paper presented at the FAI conference in Linköping', Lund University - School of Aviation.

Dekker, S. [2001], 'The re-invention of human error', *Human factors and aerospace safety*.

Dekker, S. and Hollnagel, E. [2004], 'Human factors and folk models', *Cognition, Technology and Work*, **6**(2), 79–86.

Dekker, S., Suparamaniam, N. and Veritas, D. [2005], Divergent images of decision making in international disaster relief work, Technical report, Technical Report 2005-01, Lund University School of Aviation.

Dekker, S. W. a. [2002], 'Reconstructing human contributions to accidents: the new view on error and performance.', *Journal of safety research*, Elsevier, **33**(3), 371–85.

Dekker, S. W. A. [2003], Errors in our understanding of human error : the real lessons from aviation for healthcare, Technical report, Lund University - School of Aviation.

Dempster, A. [1967], 'Upper and lower probabilities induced by a multivalued mapping', *The Annals of Mathematical Statistics*, **38**(2), 325–339.

Diez, M., Boehm-Davis, D., Holt, R. and Pinney, M. [2001], 'Tracking pilot interactions with flight management systems through eye movements', *Proceedings of the 11th International Symposium on Aviation Psychology*, pp. 1—-6.

DKE Committee [2010], 'Data & Knowledge Engineering', *Data & Knowledge Engineering*, Elsevier.

Duchowski, A. [2007], *Eye tracking methodology: Theory and practice*, Springer-Verlag New York Inc, New York, New York, USA.

Duncan, S. and Barrett, L. [2007], 'Affect is a form of cognition: A neurobiological analysis', *Cognition & emotion*, **21**(6), 1184—-1211.

Dzindolet, M., Peterson, S., Pomranky, R., Pierce, L. and Beck, H. [2003], 'The role of trust in automation reliance', *International Journal of Human-Computer Studies*, Elsevier, **58**(6), 697–718.

ECLiPSe [2010], 'ECLiPSe'.
URL: *http://eclipseclp.org/*

Edland, A. and Svenson, O. [1993*a*], 'Judgment and decision making under time pressure: Studies and findings', *Time pressure and stress in human judgment and decision making*, pp. 27—-40.

Edland, A. and Svenson, O. [1993*b*], 'Judgment and decision making under time pressure: Studies and findings', *Time pressure and stress in human judgment and decision making*, pp. 27—-40.

Edwards, W. [1954], 'The theory of decision making.', *Psychological Bulletin*, American Psychological Association, **51**(4), 380.

Ehrhardt, J., Päsler-Sauer, J., Schüle, O., Benz, G. and M [1993], 'Development of RODOS*, A Comprehensive Decision Support System for Nuclear Emergencies in Europe-An Overview', *Radiation Protection*, **50**, 195.

Elkhyari, A., Gueret, C. and Jussien, N. [2002], 'Conflict-based repair techniques for solving dynamic scheduling problems', *Lecture notes in computer science*.

Elkhyari, A., Guéret, C. and Jussien, N. [2004], Constraint programming for dynamic scheduling problems, *in* 'International Scheduling Symposium (ISS'04), Awaji, Hyogo, Japan', pp. 84—-89.

Elliot, T. [2005], Expert decision-making in naturalistic environments: a summary of research, Technical report, Department of Defence – Australian Government.

Endsley, M. [1995*a*], 'A taxonomy of situation awareness errors', *Human factors in aviation operations*, pp. 287–292.

Endsley, M. [1995*b*], 'Measurement of situation awareness in dynamic systems', *Human Factors: The Journal of the Human Factors and Ergonomics Society*, **37**(1), 65–84.

Endsley, M. [1995*c*], 'Toward a theory of situation awareness in dynamic systems', *The Journal of the Human Factors*.

Endsley, M. [1996], 'Automation and situation awareness', *Automation and human performance: Theory and applications*, pp. 163–181.

Endsley, M. [1997*a*], Situation Awareness Information Dominance & Information Warfare., Technical report, Logicon Technical Services Inc., Dayton OH.

Endsley, M. [1997*b*], The role of situation awareness in naturalistic decision making, *in* 'Naturalistic decision making', Lawrence Erlbaum, pp. 269—-283.

Endsley, M. [1999], 'Situation awareness in aviation systems', *Handbook of aviation human factors*.

Endsley, M. and Kiris, E. [1995], 'The out-of-the-loop performance problem and level of control in automation.', *Human Factors*, Human Factors and Ergonomics Society, **37**(2).

Endsley, M. and Strauch, B. [1997], Automation and situation awareness: The accident at Cali, Columbia, *in* 'Proceedings of the ninth international symposium on aviation psychology', Vol. 5 Suppl 1, pp. 877–881.

Estrada, C., Isen, A. and Young, M. [1997], 'Positive Affect Facilitates Integration of Information and Decreases Anchoring in Reasoning among Physicians', *Organizational behavior and human decision processes*, Elsevier, **72**(1), 117–135.

EUROCAE [1996], 'ED-79 / ARP 4754 Certification Considerations for Highly Integrated or Complex Aircraft Systems', *October*, (October).

European Aircraft Developer Team [2010], 'x737 Project'.
URL: *http://www.eadt.eu*

Evans, G. [1984], *Environmental stress*, Cambridge Univrsity Press.

Falzer, P. R. [2004], 'Cognitive schema and naturalistic decision making in evidence-based practices.', *Journal of biomedical informatics*, **37**(2), 86–98.

Felfernig, A. and Burke, R. [2008], 'Constraint-based recommender systems: technologies and research issues', *Proceedings of the 10th international conference on*.

Felfernig, A., Friedrich, G., Schubert, M., Mandl, M., Mairitsch, M. and Teppan, E. [2009], Plausible repairs for inconsistent requirements, *in* 'Proceedings of the 21st International Joint Conference on Artificial Intelligence', p. 791.

Felfernig, A., Isak, K., Szabo, K. and Zachar, P. [2007], 'The VITA financial services sales support environment', *PROCEEDINGS OF THE NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE*, **22**(2), 1692–1699.

Feltz, D. and Landers, D. [1983], 'The effects of mental practice on motor skill learning and performance: A meta-analysis.', *Journal of Sport Psychology. Vol*, **5**(1), 25—-57.

Fiedler, K. [2001], 'Affective influences on social information processing', *Handbook of affect and social cognition*.

Filyner, B. [2003], 'Open systems avionics architectures considerations', *IEEE Aerospace and Electronic Systems Magazine*, **18**(9), 3—-10.

Fiorino, F. [2009], 'Boeing Warns of Possible 737 Altimeter Fault'.
URL: *http://www.aviationweek.com*

Fischhoff, B., Kahneman, D., Slovic, P. and Tversky, A. [1974], 'Judgment under uncertainty: Heuristics and biases', *Science*, **185**, 1124—-1131.

Fischhoff, B. and Lichtenstein, S. [1984], *Acceptable risk*, Cambridge University Press.

Fiske, S. [1993], 'Social cognition and social perception', *Annual review of psychology*, **44**(1), 155—-194.

Fitts, P. and Posner, M. [1967], *Human performance*, Brooks/Cole Publishing Co.

Flanagan, J. [1954], 'The critical incident technique', *Psychological bulletin*, **51**(4), 327—-358.

Flemisch, F. and Onken, R. [2000], 'Detecting usability problems with eye tracking in airborne battle management support', *Usability of Information in Battle Management Operations*, **1**.

Fox, C. and Tversky, A. [1995], 'Ambiguity aversion and comparative ignorance', *The Quarterly Journal of Economics*, Oxford University Press, **110**(3), 585–603.

Fraser, J., Smith, P., Smith, J. and Others [1992], 'A catalog of errors', *International Journal of Man-Machine Studies*, Elsevier, **37**(3), 265–307.

Frayman, F. [2001], User-interaction requirements and its implications for efficient implementations of interactive constraint satisfaction systems, *in* 'Working Notes of the 1st International Workshop on User-Interaction in Constraint Satisfaction', Citeseer, pp. 31–41.

Frei, C. and Faltings, B. [1999], Resource allocation in networks using abstraction and constraint satisfaction techniques, *in* 'Principles and Practice of Constraint Programming-CP1999', Springer, pp. 204–218.

Freuder, E., Likitvivatanavong, C., Moretti, M., Rossi, F. and Wallace, R. [2003], 'Computing explanations and implications in preference-based configurators', *Recent Advances in Constraints*, Springer, pp. 315–336.

Freuder, E., Likitvivatanavong, C. and Wallace, R. [2001*a*], Deriving explanations and implications for constraint satisfaction problems, *in* 'Principles and Practice of Constraint Programming (CP 2001)', Springer, pp. 585–589.

Freuder, E., Likitvivatanavong, C. and Wallace, R. [2001*b*], Explanation and implication for configuration problems, *in* 'Proc. of the 17th Int. Joint Conf. on Artificial Intelligence (IJCAI 2001) Workshop on Configuration', Citeseer.

Gaillard, A. [2008], 'Concentration, stress and performance', *Performance under stress*, pp. 59—-75.

Gaschnig, J. [1987], *Performance measurement and analysis of certain search algorithms*, Carnegie-Mellon University.

Gasti, W., Senior, A., Emam, O., Jorden, T., Knowelden, R. and Fowell, S. [2007], Modular Architecture for Robust Computation, *in* 'International Spacewire Conference 2007', European Space Agency Publications, Dundee.

Gaudin, E., Jussien, N., de Nantes, E. and Rochart, G. [2008], Explained global constraints at work, Technical Report 2, Technical report 04-03-INFO, École des Mines de Nantes.

Gigerenzer, G. [1987], 'Survival of the fittest probabilist: Brunswik, Thurstone, and the two disciplines of psychology', *The probabilistic revolution*, **2**, 49–72.

Gigerenzer, G., Hoffrage, U. and Kleinbölting, H. [1991], 'Probabilistic mental models: A Brunswikian theory of confidence.', *Psychological Review*, **98**(4), 506–528.

Gigerenzer, G. and Todd, P. [1999], *Simple heuristics that make us smart*, Oxford University Press, USA.

Gillis, J. [1993], 'Effects of life stress and dysphoria on complex judgments.', *Psychological reports*, **72**(3), 1355—-1363.

Ginsberg, M. [1993], 'Dynamic backtracking', *Journal of Artificial Intelligence Research*, **1**, 25–46.

Ginsberg, M. and McAllester, D. [1994], GSAT and dynamic backtracking, *in* 'Principles and Practice of Constraint Programming', Springer, pp. 243–265.

Ginty, L. M. and Smyth, B. [2002], 'Comparison-Based Recommendation', *Lecture Notes in Computer Science*.

Goldberg, E., Driedger, N. and Kittredge, R. [1994], 'Using natural-language processing to produce weather forecasts', *IEEE Expert*, **9**(2), 45—-53.

Goldberg, J. and Kotval, X. [1999], 'Computer interface evaluation using eye movements: Methods and constructs', *International Journal of Industrial Ergonomics*, Elsevier B.V., **24**(6), 631–645.

Goldman, A. [2002], 'Simulation theory and mental concepts', *Simulation and knowledge of action*, pp. 1—-19.

Gonzalez, C., Dana, J., Koshino, H. and Just, M. [2005], 'The framing effect and risky decisions: Examining cognitive functions with fMRI', *Journal of Economic Psychology*, Elsevier, **26**(1), 1–20.

Goodchild, P. C. and Whiston, P. J. [1998], 'SHIMA - Small Aircraft / Helicopter Integrated Modular Avionics', pp. 1–11.

Granston, E. and Holler, A. [2001], 'Automatic recommendation of compiler options', *4th Workshop on Feedback-Directed and Dynamic*.

Gregor, S. and Yu, X. [2002], 'Exploring the Explanatory Capabilities of Intelligent System Technologies', *Fuzzy logic: a framework for the new millennium*, Springer, p. 288.

Greitzer, F., Podmore, R., Robinson, M. and Ey, P. [2010], 'Naturalistic Decision Making for Power System Operators', *International Journal of Human-Computer Interaction*, **26**(2), 278–291.

Gruber, T. [1995], 'Toward principles for the design of ontologies used for knowledge sharing?', *International Journal of Human-Computer Studies*, **43**(5-6), 907–928.

Gruner, W. [1990], No Time For Decision Making, *in* 'Proceedings of the US Naval Institute', Vol. 116, pp. 39–41.

Guéret, C. and Prins, C. [1998], 'Classical and new heuristics for the open-shop problem: A computational evaluation', *European Journal of Operational Research*, Elsevier, **107**(2), 306–314.

Guerini, M., Stock, O. and M [2003], 'Persuasion models for intelligent interfaces', *Proceedings of the IJCAI Workshop on Computational Models of Natural Argument*.

Guiotto, A., Martelli, A., Paccagnini, C. and Lavagna, M. [2003], 'SMART-FDIR: Use of Artificial Intelligence in the Implementation of a Satellite FDIR', *Data Systems in Aerospace Conference*.

Hadžić, T., Wasowski, A. and Andersen, H. [2005], Techniques for Efficient Interactive Configuration of Distribution Networks, *in* 'Proceedings of the 20th international joint conference on Artifical intelligence', Morgan Kaufmann Publishers Inc.

Hadzic, T. and Andersen, H. [2004], An introduction to solving interactive configuration problems, Technical Report August, The IT University of Copenhagen.

Hadzic, T., Subbarayan, S., Jensen, R., Andersen, H., Mø ller, J. and Hulgaard, H. [2007], 'Fast backtrack-free product configuration using a precompiled solution space representation', Citeseer, **10**(1), 3.

Halford, G. and Wilson, W. [1998], 'Processing capacity defined by relational complexity: Implications for comparative, developmental, and cognitive psychology', *Behavioral and Brain Sciences*, **21**(6), 803–831.

Hall, D. and Llinas, J. [1997], 'An introduction to multisensor data fusion', *Proceedings of the IEEE*.

Hammond, K. [2000*a*], 'Coherence and correspondence theories in judgment and decision making', *Judgment and decision making: An interdisciplinary reader*, pp. 53–65.

Hammond, K. [2000*b*], *Judgments under stress*, Oxford University Press, USA.

Hanson, E. [2004], Focus of attention and pilot error, *in* 'Proceedings of the 2004 Symposium on Eye Tracking Research & Applications', ACM Press, San Antonio, TX, USA, p. 60.

Hart, S. [2006], 'Nasa-task load index (nasa-tlx); 20 years later', *Human Factors and Ergonomics Society Annual Meeting Proceedings*, **50**(9), 904–908.

Hart, S. and Staveland, L. [1988], 'Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research', *Human mental workload*, Elsevier Science Publishers BV (North-Holland), **1**, 139–183.

Häubl, G. and Trifts, V. [2000], 'Consumer decision making in online shopping environments: The effects of . . . ', *Marketing Science*.

Hayashi, M., Huemer, V. and Lachter, J. [2006], 'Evaluation of an Advanced Fault Management System Display for Next Generation Crewed Space Vehicles', *Human Factors and Ergonomics Society Annual Meeting Proceedings*, Human Factors and Ergonomics Society, **50**(1), 136—-140.

Hayashi, M., Ravinder, U., Beutter, B. and McCann, R. [2009], 'Operator Performance Evaluation of Fault Management Interfaces for Next-Generation Spacecraft', *SAE International Journal of Aerospace*, **1**(1), 164.

Heath, C. and Tversky, A. [1991], 'Preference and belief: Ambiguity and competence in choice under uncertainty', *Journal of Risk and Uncertainty*.

Herlocker, J., Konstan, J., Borchers, A. and J [1999], 'An algorithmic framework for performing collaborative filtering', *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, p. 237.

Herlocker, J., Konstan, J. and Riedl, J. [2000], 'Explaining collaborative filtering recommendations', *Proceedings of the 2000 ACM conference on Computer supported cooperative work*, ACM Press, pp. 241–250.

Hesslow, G. [2002], 'Conscious thought as simulation of behaviour and perception', *Trends in cognitive sciences*, **6**(6), 242—-247.

Hilburn, B., Jorna, P., Byrne, E. and Parasuraman, R. [1997], 'The effect of adaptive air traffic control (ATC) decision aiding on controller mental workload', *Human-automation interaction: Research and practice*, pp. 84–91.

Hill, S., Iavecchia, H., Byers, J. and AC [1992], 'Comparison of four subjective workload rating scales', *The Journal of the Human Factors and Ergonomics Society*, Human Factors and Ergonomics Society, **34**(4), 429—-439.

Hill, W., Stead, L., Rosenstein, M. and Furnas, G. [1995], Recommending and evaluating choices in a virtual community of use, *in* 'Proceedings of the SIGCHI conference on Human factors in computing systems', ACM Press/Addison-Wesley Publishing Co., pp. 194–201.

Hladik, P., Cambazard, H. and Déplanche, A. [2008], 'Solving a real-time allocation problem with constraint programming', *The Journal of Systems & Software*, **81**(1), 131—-149.

Hladik, P. E., Cambazard, H. and Deplanche, A. M. [2005], 'How to solve allocation problems with constraint programming', *Proceedings of the Work In Progress of the 17th Euromicro*, pp. 25–28.

Hoffman, P. [1960], 'The paramorphic representation of clinical judgment', *Psychological Bulletin*, Elsevier, **57**(2), 116–131.

Hogarth, R. [1975], Decision time as a function of task complexity, *in* 'Utility, probability, and human decision making: selected proceedings of an interdisciplinary research conference', Springer, Rome, p. 321.

Hogarth, R. and Kunreuther, H. [1995], 'Decision making under ignorance: Arguing with yourself', *Journal of Risk and Uncertainty*, **10**(1), 15—-36.

Hollands, J. and Wickens, C. [1999], *Engineering Psychology and Human Performance*, Prentice Hall.

Horvitz, E. J., Breese, J. S. and Henrion, M. [1988], 'Decision theory in expert systems and artificial intelligence', *International Journal of Approximate Reasoning*, **2**(3), 247–302.

Huemer, V., Matessa, M. and McCann, R. [2005], 'Fault management during dynamic spacecraft flight: effects of cockpit display format and workload', *2005 IEEE International Conference on Systems, Man and Cybernetics*, **1**.

Hughes, D. and Dornheim, M. [1995], 'Accidents direct focus on cockpit automation', *American Psychological Association (APA)*, Aviation Week and Space Technology, pp. 52–55.

Hughes, R., Balestrini, S., Kelly, K., Weston, N. and Mavris, D. [2006], Modeling of an Integrated Reconfigurable Intelligent System (IRIS) for Ship Design, *in* 'Ships & Ship Systems (S3) Technology Symposium Change, Challenges & Constants'.

Hunter, D. [2006], 'Risk perception among general aviation pilots', *The International Journal of Aviation Psychology*.

Hutchins, S., Kelly, R. and Morrison, J. [1996], Decision Support for Tactical Decision Making Under Stress, *in* 'Proceedings of the Second International Symposium on Command and Control Research and Technology'.

Hutchins, S., Morrison, J. and Kelly, R. [1996], Principles for aiding complex military decision making, *in* 'Proceedings of the Second international command and control research and technology symposium', pp. 25–28.

IBM [1968], 'IBM SPSS'.
URL: *http://www-01.ibm.com/software/analytics/spss/products/statistics/*

ILOG [2010], 'ILOG'.
URL: *www.ilog.com*

Inakoshi, H., Okamoto, S., Ohta, Y. and Yugami, N. [2001], 'Effective decision support for product configuration by using cbr', *Proceedings of the Fourth International Conference on Case-Based Reasoning (ICCBR), Workshop Casebased Reasoning in Electronic Commerce, Vancouver, Canada*.

Intel [1999], 'OpenCV library (http://opencv.willowgarage.com/wiki)'.
URL: *http://opencv.willowgarage.com/wiki/*

Investigation Commission of Ministry of Transport - France [1989], Final report concerning the accident which occurred on June 26th 1988 at Mulhouse-Habsheim (68) to the Airbus A 320, registered F-GFKC, Technical report.

Isen, A. [2001], 'An influence of positive affect on decision making in complex situations: Theoretical issues with practical implications', *Journal of Consumer Psychology*, **11**(2), 75–85.

Isen, A., Nygren, T. and Ashby, F. [1988], 'Influence of positive affect on the subjective utility of gains and losses: It is just not worth the risk', *Journal of Personality and Social Psychology*, **55**(5), 710—-717.

Itier, J.-B. [2007], A380 Integrated Modular Avionics, *in* 'Proceedings of the ARTIST2 meeting on Integrated Modular Avionics', INRIA, Rome, Italy.
URL: *http://www.artist-embedded.org/artist*

Jahanian, F. and Mok, A. [1986], 'Safety analysis of timing properties in real-time systems.', *IEEE Transactions on software engineering*, **12**(9), 890–904.

Janis, I. and Mann, L. [1977], 'Decision making'.

Jannach, D. [2010], *Recommender Systems: An Introduction*, Cambridge University Press.

Jeannott, E., Kelly, C. and Thompson, D. [2003], The development of situation awareness measures in ATM systems. EATMP Report, Technical report, HRS/HSP-005-REP-01.

Jensen, R. [2004], 'CLab: A C++ library for fast backtrack-free interactive product configuration', *Principles and Practice of Constraint Programming, CP 2004*, Springer, pp. 816–816.

Jian, J., Bisantz, A. and Drury, C. [2000], 'Foundations for an empirically determined scale of trust in automated systems', *International Journal of Cognitive Ergonomics*, **4**(1), 53—-71.

Johnson, E. and Payne, J. [1995], 'Adapting to time constraints', *Time pressure and stress in human judgment and decision making*, pp. 167—-178.

Johnson, J. and Sarason, I. [1977], 'Life stress, depression an anxiety: external control as a moderator variable', *Journal of Psychosomatic Research*, **22**(170), 205–208.

Jolliffe, G. and Nicholson, M. [2005], 'Exploring the Possibilities Towards a Preliminary Safety Case for IMA Blueprints', *Constituents of Modern System-safety Thinking*, Springer, pp. 163–181.

Jones, E. and Goethals, G. [1972], 'Order effects in impression formation: Attribution context and the nature of the entity', *Attribution: Perceiving the causes of behavior*, pp. 27–46.

Jones, E., Loyell, B. and Wilson, J. [2010], 'Flightdeckautomation.com'.
URL: *http://www.flightdeckautomation.com*

Junker, U. [2001], QUICKXPLAIN: Conflict detection for arbitrary constraint propagation algorithms, *in* 'IJCAI-2001 Workshop on Modelling and Solving problems with constraints', Elsevier.

Jussien, N. [2001], 'e-constraints: Explanation-based constraint programming', *Proc. CP01 Workshop on User-Interaction in Constraint Satisfaction, Paphos, Cyprus*.

Jussien, N. [2003], 'The versatility of using explanations within constraint programming', *Research Report 03Y04-INFO Ecole des Mines de Nantes*.

Jussien, N. and Debruyne, R. [2007], Explanation-based repair techniques for constraint programming, Technical report, Technical report D3.2.3, Ecole de Mines de Nantes.

Jussien, N., Debruyne, R. and Boizumault, P. [2000], 'Maintaining arc-consistency within dynamic backtracking', *Principles and Practice of Constraint Programming*, Springer, pp. 249–261.

Jussien, N. and Lhomme, O. [2002], 'Local search with constraint propagation and conflict-based heuristics', *Artificial Intelligence*, Elsevier, **139**(1), 21–45.

Jussien, N. and Ouis, S. [2001], 'User-friendly explanations for constraint programming', *ICLP'01 11th Workshop on Logic Programming Environments (WLPE'01), Paphos, Cyprus.*

Kaber, D. and Endsley, M. [2004], 'The effects of level of automation and adaptive automation on human performance, situation awareness and workload in a dynamic control task', *Theoretical Issues in Ergonomics Science*, Taylor & Francis, **5**(2), 113–153.

Kaber, D., Riley, J., Tan, K.-W. and Endsley, M. [2001], 'On the Design of Adaptive Automation for Complex Systems', *International Journal of Cognitive Ergonomics*, **5**(1), 37–57.

Kahneman, D. and Tversky, A. [1979], 'Prospect theory: An analysis of decision under risk', *Econometrica: Journal of the Econometric Society*, **47**(2), 263—-291.

Kahneman, D. and Tversky, A. [1981], *The Simulation Heuristic*, Department of Psychology, Stanford University CA.

Kelly, T. and Weaver, R. [2004], The goal structuring notation - a safety argument notation, *in* 'Proc. DSN 2004 Workshop on Assurance Cases', IEEE Comput. Soc.

KER [2010], *The Knowledge Engineering Review*, Cambridge Journals.

Kern, T. and Kern, A. [1998], *Flight discipline*, McGraw-Hill Professional.

Klapproth, F. [2008], 'Time and decision making in humans', *Cognitive, Affective, and Behavioral Neuroscience*, **8**(4), 509.

Klayman, J. and Ha, Y. [1989], 'Hypothesis testing in rule discovery: Strategy, structure, and content.', *Journal of Experimental Psychology: Learning, Memory, and Cognition*, **15**(4), 596—-604.

Klein, G. [1989], *Recognition-primed decisions*, Klein Associates Inc., Fairborn, OH, USA.

Klein, G. [1993*a*], 'A recognition-primed decision (RPD) model of rapid decision making', *Decision making in action: Models and methods*, pp. 138—-147.

Klein, G. [1993*b*], 'Naturalistic decision making: Implications for design', *State-of-the-Art Report. Dayton, OH: Crew Systems Ergonomics Information Analysis Center, Wright-Patterson Air Force Base*.

Klein, G. [1997*a*], 'The recognition-primed decision (RPD) model: Looking back, looking forward', *Naturalistic decision making*, Lawrence Erlbaum, pp. 285–292.

Klein, G. [1998], *Sources of power*, Mit Press Cambridge, MA.

Klein, G. and Calderwood, R. [1986], 'Rapid decision making on the fire ground', *Human Factors and Ergonomics Society Annual Meeting Proceedings*, **30**(6), 576—-580.

Klein, G., Orasanu, J., Calderwood, R. and Zsambok, C. [1993], *Decision making in action: Models and methods*, Ablex Publishing Co., Norwood, NJ.

Klein, W. [1997*b*], *Nietzsche and the Promise of Philosophy*, State Univ of New York Pr.

Klir, G. [1999], *Uncertainty-based information: Elements of generalized information theory*, Springer Verlag.

Knight, J., Strunk, E. and Sullivan, K. [2003], 'Towards a rigorous definition of information system survivability', *Proceedings DARPA Information Survivability Conference and Exposition*, IEEE Comput. Soc, pp. 78–89.

Koehler, D. [1991], 'Explanation, imagination, and confidence in judgment', *Psychological Bulletin*, **110**(3), 499—-519.

Kontogiannis, T. and Kossiavelou, Z. [1999], 'Stress and team performance: principles and challenges for intelligent decision aids', *Safety science*, **33**(3), 103—-128.

Koriat, A., Lichtenstein, S. and Fischhoff, B. [1980], 'Reasons for confidence', *Journal of Experimental Psychology: Human Learning and Memory*, **6**(2), 107—-118.

Kowalski-Trakofler, K., Vaught, C. and Scharf, T. [2003], 'Judgment and decision making under stress: an overview for emergency managers', *International Journal of Emergency Management*, Inderscience, **1**(3), 278–289.

Krodel, J. and Romanski, G. [2008], Handbook for Real-Time Operating Systems Integration and Component Integration Considerations in Integrated Modular Avionics Systems, Technical Report January, U.S. Department of Transportation.

Laburthe, F. [2000], CHOCO: implementing a CP kernel, *in* 'Proceedings of TRICS: Techniques for Implementing Constraint programming Systems, a post-conference workshop of CP', pp. 71–85.

Laminar Research Inc. [1993], 'X-Plane Flight Simulator'.
URL: *http://www.x-plane.com*

Laminar Research Inc. [2009], 'X-Plane SDK'.
URL: *http://www.xsquawkbox.net/xpsdk/mediawiki/Main_Page*

Langley, P. [1999], 'User modeling in adaptive interfaces', *Courses and lectures - International centre for mechanical sciences*, pp. 357—-370.

Larkin, J. H. and Simon, H. [1987], 'Why a Diagram is (Sometimes) Worth Ten Thousand Words', *Engineering Education*, **99**(1), 65–99.

Lavoie, B., Rambow, O. and Reiter, E. [1997], 'Customizable descriptions of object-oriented models', *Proceedings of the fifth conference on Applied natural language processing*, pp. 253—-256.

Lawson, G., Sharples, S., Clarke, D. and Cobb, S. [2009], 'Development of a Technique for Predicting the Human Response to an Emergency Situation', *Engineering Psychology and Cognitive Ergonomics*, Springer, pp. 22–31.

Layton, C., Smith, P. and Coy, C. M. [1994], 'Design of a cooperative problem-solving system for en-route flight planning: An empirical evaluation', *Human Factors: The Journal of the Human Factors and Ergonomics Society*, **36**(1), 94—-119.

LC Technologies, I. [2010], 'Eyegaze (http://www.eyegaze.com)'.
URL: *http://www.eyegaze.com*

Lecoutre, C. and Szymanek, R. [2006], 'Generalized arc consistency for positive table constraints', *Principles and Practice of Constraint Programming-CP 2006*, Springer, pp. 284–298.

LeDoux, J. [1998], *The emotional brain: The mysterious underpinnings of emotional life*, Touchstone books.

LeDoux, J. [2002], 'Emotion, memory and the brain', *Special Editions — Scientific American*, Touchstone books.

Lee, J. and Moray, N. [1992], 'Trust, control strategies and allocation of function in human-machine systems', *Ergonomics*, Taylor & Francis, **35**(10), 1243–1270.

Lee, J. and Moray, N. [1994], 'Trust, self-confidence, and operators' adaptation to automation', *International Journal of Human-Computer Studies*, **40**(1), 153—-184.

Lee, J. and See, K. [2004], 'Trust in automation: Designing for appropriate reliance', *Human factors*, **46**(1), 50.

Lee, Y.-H., Kim, D., Younis, M. and Zhou, J. [2000], 'Scheduling tool and algorithm for integrated modular avionics systems', *19th DASC. 19th Digital Avionics Systems Conference. Proceedings*, Ieee, pp. 1C2/1–1C2/8.

Leeuwen, P. V., Hesselink, H. H. and Rohling, J. H. T. [2002], 'Scheduling aircraft using constraint satisfaction', *Electronic notes in theoretical computer science*, **76**, 252—-268.

Lehmann, E. [1950], 'Theory of testing hypotheses'.

Lehner, P. and Zirk, D. [1987], 'Cognitive factors in user/expert-system interaction', *Human Factors: The Journal of the Human Factors and Ergonomics Society*, Human Factors and Ergonomics Society, **29**(1), 97–109.

Lerner, J. and Keltner, D. [2000], 'Beyond valence: Toward a model of emotion-specific influences on judgement and choice', *Cognition & Emotion*, **14**(4), 473—-493.

Lerner, J. and Tiedens, L. [2006], 'Portrait of the angry decision maker: How appraisal tendencies shape anger's influence on cognition', *Journal of Behavioral Decision Making*, **19**(2), 115—-137.

Lester, J. and Porter, B. [1997], 'Developing and empirically evaluating robust explanation generators: The KNIGHT experiments', *Computational Linguistics*, **23**(1), 65—-101.

Leveson, N. [1995], *Safeware*, Addison-Wesley.

Levin, I., Schneider, S. and Gaeth, G. [1998], 'All Frames Are Not Created Equal: A Typology and Critical Analysis of Framing Effects.', *Organizational behavior and human decision processes*, **76**(2), 149–188.

Lhomme, O. [1993], Consistency techniques for numeric CSPs, *in* 'International Joint Conference on Artificial Intelligence', Vol. 13, Citeseer, pp. 232–232.

Linden, G., Hanks, S. and Lesh, N. [1997], 'Interactive assessment of user preference models: The automated travel assistant', *Courses and lectures - International centre for mechanical sciences*, pp. 67—-78.

Lipshitz, R. and Shaul, O. [1997], *Schemata and mental models in recognition-primed decision making*, Mahwah, NJ, Erlbaum.

Lipshitz, R. and Strauss, O. [1997], 'Coping with uncertainty: A naturalistic decision-making analysis', *Organizational Behavior and Human Decision Processes*, Elsevier, **69**(2), 149–163.

Llinas, J. and Hall, D. [1998], 'An introduction to multi-sensor data fusion', *IEEE International Symposium on Circuits and Systems*, **6**, 537–540.

Lockheed Martin [2010], 'Lockheed Martin - F-22 Raptor'.
URL: *http://www.lockheedmartin.com/products/f22/index.html*

Loewenstein, G. and Lerner, J. [2003], The role of affect in decision making, *in* 'Handbook of affective science', Oxford University Press, pp. 619—-642.

Lohse, G. and Johnson, E. [1996], 'A comparison of two process tracing methods for choice tasks', *Proceedings of the Twenty-Ninth Hawaii International Conference on System Sciences*, **4**.

Lopes, L. [1992], 'Three misleading assumptions in the customary rhetoric of the bias literature', *Theory and Psychology*, **2**(2), 231–236.

Luce, M., Payne, J. and Bettman, J. [1999], 'Emotional trade-off difficulty and choice', *Journal of Marketing Research*.

Luximon, A. and Goonetilleke, R. [2001], 'Simplified subjective workload assessment technique', *Ergonomics*, Taylor & Francis, **44**(3), 229—-243.

Madsen, J. [2003], Methods for interactive constraint satisfaction, Master's thesis, University of Copenhagen.

Main Commission Aircraft Accident Investigation - Poland [1994], Report on the accident to Airbus A320-211 Aircraft in Warsaw on 14 September 1993, Technical report.

Manhartsberger, M. and Zellhofer, N. [2005], 'Eye tracking in usability research: What users really see', *Usability Symposium, 2005*, pp. 141–152.
URL: *http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.94.2949*

Mano, H. [1992], 'Judgments under distress: Assessing the role of unpleasantness and arousal in judgment formation', *Organizational Behavior and Human Decision Processes*, Elsevier, **52**(2), 216–245.

Mano, H. [1999], 'The influence of pre-existing negative affect on store purchase intentions', *Journal of Retailing*.

Mark, G. and Kobsa, A. [2005], 'The Effects Of Collaboration And System Transparency'.
URL: *http://en.scientificcommons.org/42216458*

Martensson, L. and Singer, G. [1999], Warning systems in commercial aircraft- Perceptual and cognitive problems, *in* 'International Symposium on Aviation Psychology, 10 th, Columbus, OH', pp. 32–36.

Matsumoto, K. and Sakaguchi, T. [1992], 'Knowledge-based systems as operational aids in power system restoration', *Proceedings of the IEEE*, **80**(5), 689.

Matthews, M. and Beal, S. [2002], Assessing situation awareness in field training exercises (Report 1795), Technical report, U.S. Army Research Institute for the Behavioural and Social Sciences.

Mayer, R. E. and Gallini, J. K. [1990], 'When Is an Illustration Worth Ten Thousand Words ?', *Journal of Educational Psychology*, **82**(4), 715–726.

Mazzotta, I., de Rosis, F. and Carofiglio, V. [2007], 'Portia: A User-Adapted Persuasion System in the Healthy-Eating Domain', *IEEE Intelligent Systems*, **22**(6), 42–51.

McCann, R. and Spirkovska, L. [2005], 'Human factors of integrated systems health management on next-generation spacecraft', *Proceedings of the First International Forum on Integrated System Health Engineering and Management in Aerospace, Nov*, pp. 7–10.

Mellers, B., Schwartz, A. and Cooke, A. [1998], 'Judgment and decision making', *Annual Review of Psychology*, Annual Reviews, **49**(1), 447–477.

Merchant, S. and Schnell, T. [2001], 'Eye movement research in aviation and commercially available eye trackers today', *colletm.free.fr*.
URL: *http://colletm.free.fr/archives_colletm/bib/Attention/FinalEyeTrackingReportAug17.pdf*

Meso, P., Troutt, M. D. and Rudnicka, J. [2002], 'A review of naturalistic decision making research with some implications for knowledge management', *Journal of Knowledge Management*, **6**(1), 63–73.

Miller, C. and Larson, R. [1992], An explanatory and 'argumentative' interface for a model-based diagnostic system, *in* 'Proceedings of the 5th annual ACM symposium on User interface software and technology', ACM, p. 52.

Ministry of Civil Aviation - Government of India [1990], Report on Accident to Indian Airlines Airbus A-320 Aircraft VT-EPN at Bangalore, February 14, 1990, Technical report.

Ministry Of Transport [1994], 'Ministry of Transport. (1996). Aircraft Accident Investigation Commission. China Airlines Airbus Industries A300B4-622R, B1816, Nagoya Airport, April 26, 1994. (Report 96-5). Japan'.

Mittal, S. and Frayman, F. [1989], 'Towards a generic model of configuration tasks', *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence (IJCAI-89)*, Elsevier, pp. 1395—-1401.

Mø ller, J., Andersen, H. and Hulgaard, H. [2001], 'Product configuration over the internet', *Proceedings of the 6th INFORMS*, Citeseer.

Montano, G. [2007], 'Reconfiguration Management for Integrated Modular Avionics', *Technical report, Department of Computer Science, The University of York*, The University of York, (September).

Montano, G., Norridge, P., Sullivan, W., Topping, C. and Wishart, A. [2010], Dynamically Reconfigurable Processing Module for Future Space Applications, *in* 'Proceedings of the Data Systems In Aerospace International Conference (DASIA 2010)', European Space Agency (ESA), Budapest, Hungary.

Mooney, R. and Roy, L. [2000], 'Content-based book recommending using learning for text categorization', *Proceedings of the fifth ACM conference on Digital libraries*, pp. 195—-204.

Moray, N. [1987], 'Intelligent aids, mental models, and the theory of machines', *International Journal of Man-Machine Studies*, Elsevier, **27**(5-6), 619–629.

Moray, N. [1996], A taxonomy and theory of mental models, *in* 'Human Factors and Ergonomics Society Annual Meeting Proceedings', Vol. 40, Human Factors and Ergonomics Society, pp. 164–168.

Morris, J. and Fletcher, J. [2002], *Cognitive readiness (IDA Paper P-3735)*, Institute for Defense Analyses, Alexandria, VA.

Morrison, J., Marshall, S., Kelly, R. and Moore, R. [1997], Eye tracking in tactical decision making environments: implications for decision support evaluation, *in* 'Proceedings of the Third International Symposium on Command and Control Research and Technology', National Defense University, Washington, DC, USA, pp. 17–20.

Mosier, K. [2010], 'The Human in Flight: From Kinesthetic Sense to Cognitive Sensibility', *Human Factors in Aviation*, Academic Press, p. 147.

Mosier, K. and Fischer, U. [2009], 'Does Affect Matter in Naturalistic Decision Making?', *Proceedings of NDM9, the Ninth International Conference on Naturalistic Decision Making*, pp. 99–104.

Mosier, K. and Skitka, L. [1996], 'Human decision makers and automated decision aids: Made for each other', *Automation and human performance: Theory and applications*, pp. 201–220.

Muir, B. [1994], 'Trust in automation: Part I. Theoretical issues in the study of trust and human intervention in automated systems', *Ergonomics*, Taylor & Francis, **37**(11), 1905–1922.

Mumaw, R., Sarter, N. and Wickens, C. [2001], 'Analysis of pilots' monitoring and performance on an automated flight deck'.

Murphy, R. [1996], 'Biological and Cognitive Foundations of Intelligent Sensor Fusion', *IEEE Transactions on Systems, Man, and Cybernetics*.

Murphy, R. [1998], 'Dempster-Shafer theory for sensor fusion in autonomous mobile robots', *IEEE Transactions on Robotics and Automation*, **14**(2), 197–206.

Nakayama, M., Takahashi, K. and Shimizu, Y. [2002], 'The act of task difficulty and eye-movement frequency for the'Oculo-motor indices'', *Proceedings of the 2002 symposium on Eye tracking research & applications*, ACM Press, p. 42.

NASA [2010], 'NASA-TLX Homepage'.
URL: *http://humansystems.arc.nasa.gov/groups/TLX/*

NASA and University Of California San Diego [2005], 'NASA REASoN Project'.
URL: *http://geoinfo.sdsu.edu/reason/index.htm*

Nass, C., Fogg, B. and Moon, Y. [1996], 'Can computers be teammates?', *International Journal of Human Computer Studies*, **45**(6), 669–678.

National Transportation Safety Board [1986], China Airlines Boeing 747-SP Accident Report, Technical report.

National Transportation Safety Board [1997], NTSB Identification: DCA97MA017 - Comair flight 3272, Technical report.

National Transportation Safety Board [1998], In-Flight Icing Encounter and Uncontrolled Collision with Terrain, COMAIR Flight 3272, Embraer EMB-120RT, N265CA, Monroe, Michigan, January 9, 1997, Technical report.

NATO [2005*a*], *STANAG 4626: MODULAR AND OPEN AVIONICS ARCHITECTURES PART I - ARCHITECTURE*, Vol. 4626, NATO, Brussels.

NATO [2005*b*], *STANAG 4626: MODULAR AND OPEN AVIONICS ARCHITECTURES PART VI - GUIDELINES FOR SYSTEM ISSUES*, Vol. 3, NATO, Brussels.

Neumann, J. V., Morgenstern, O., Kuhn, H. and A [1953], 'Theory of games and economic behavior'.

Nicholson, M. [1998], Selecting a topology for safety-critical real-time control systems, Phd thesis, The University of York.

Niedenthal, P. and Kitayama, S. [1994], *The heart's eye: Emotional influences in perception and attention*, Academic Press San Diego, CA.

Nietzsche, F. [1888], *Twilight of the Idols*.

Nietzsche, F. [1922], 'Begriff der Rhetorik. u: Gesammelte Werke'.

Nisbett, R. and Ross, L. [1980], 'Human inference: Strategies and shortcomings', *Social Judgment*.

Noble, D. [1993], 'A model to support development of situation assessment aids', *Decision making in action: Models and methods*, Ablex Publishing, pp. 287–305.

Norman, D. a. [1990], 'The 'Problem' with Automation: Inappropriate Feedback and Interaction, not 'Over-Automation'', *Philosophical Transactions of the Royal Society B: Biological Sciences*, **327**(1241), 585–593.

Norman, D. and Bobrow, D. [1975], 'On data-limited and resource-limited processes', *Cognitive psychology*, Elsevier, **7**(1), 44–64.

Nygren, T. [1991], 'Psychometric properties of subjective workload measurement techniques: Implications for their use in the assessment of perceived mental workload', *Human Factors: The Journal of the Human Factors and Ergonomics Society*, **33**(1), 17—-33.

O Sullivan, B., O Callaghan, B. and Freuder, E. [2005], Corrective explanation for interactive constraint satisfaction, *in* 'International Joint Conference on Artificial Intelligence', Vol. 19, Citeseer, p. 1531.

O'Hare, D., Roscoe, S., Vette, G. and Young, M. [1992], *Flightdeck performance: The human factor*, Iowa State University Press, Ames.

Olson, W. and Sarter, N. [2000], 'Automation management strategies: Pilot preferences and operational experiences', *The International Journal of Aviation Psychology*, Taylor & Francis, **10**(4), 327–341.

Orasanu, J. [1997], 'Stress and naturalistic decision making- Strengthening the weak links', *Decision making under stress- Emerging themes and applications(A 99-12526 01-53), Aldershot, United Kingdom, Ashgate, 1997,*, pp. 43–66.

Orasanu, J. and Fischer, U. [1997], 'Finding decisions in natural environments: The view from the cockpit', *Naturalistic decision making*, pp. 343–357.

Orlady, H. and Orlady, L. [2002], 'Human factors in multi-crew flight operations', *Aeronautical Journal*, **106**(1060), 321–324.

Paakko, M., Myllymäki, P., Holsti, N. and Tirri, H. [2001], Bayesian Networks for Advanced FDIR, Technical report, European Space Agency.

Painter, J., Kelly, W., Trang, J., Lee, K. and PA [1997], 'Decision support for the general aviation pilot', *IEEE International Conference on Systems, Man and Cybernetics*, **1**, 88—-93.

Pang, W. and Goodwin, S. [1996], Application of CSP Algorithms to Job Shop Scheduling Problems, *in* 'The 2nd International Symposium on Operations Research and Its Applications', Citeseer.

Papamichail, K. and French, S. [1999], 'Generating feasible strategies in nuclear emergencies-a constraint satisfaction problem', *Journal of the Operational Research Society*, Palgrave Macmillan, **50**(6), 617–626.

Parasuraman, R. [2000], 'Designing automation for human use: empirical studies and quantitative models', *Ergonomics*, **43**(7), 931–951.

Parasuraman, R., Molloy, R. and Singh, I. [1993], 'Performance Consequences of Automation-Induced 'Complacency'', *The International Journal of Aviation Psychology*, **3**(1), 1–23.

Parasuraman, R., Mouloua, M., Molloy, R. and Hilburn, B. [1996], 'Monitoring of automated systems', *Automation and human performance: Theory and applications*, pp. 91–115.

Parasuraman, R. and Riley, V. [1997], 'Humans and automation: Use, misuse, disuse, abuse.', *Human Factors*, **39**(2).

Parasuraman, R., Sheridan, T. and CD [2008], 'Situation awareness, mental workload, and trust in automation: Viable, empirically supported cognitive engineering constructs', *Cognitive Engineering*, **2**(2), 140–160.

Parasuraman, R., Sheridan, T. and Wickens, C. [2000], 'A model for types and levels of human interaction with automation', *IEEE Transactions on Systems, Man and Cybernetics, Part A*, **30**(3), 286–297.

Parkinson, P., River, W. and Kinnan, L. [2003], 'Safety-critical software development for integrated modular avionics', *Embedded System Engineering*, ELECTRONIC DESIGN AUTOMATION LTD., **11**(7), 40–41.

Payne, J., Bettman, J. and Johnson, E. [1993], *The adaptive decision maker*, Cambridge Univ Pr.

Payne, J., Bettman, J. and Luce, M. [1998], 'Behavioral decision research: An overview', *Measurement, judgment, and decision making*, pp. 303—-359.

Payne, J. and Braunstein, M. [1978], 'Risky choice: An examination of information acquisition behavior', *Memory and Cognition*.

Pazzani, M. and Muramatsu, J. [1996], 'Syskill and Webert: Identifying interesting web sites', *Proceedings of the national conference on artificial intelligence*, pp. 54—-61.

Pennington, N. and Hastie, R. [1988], 'Explanation-based decision making: Effects of memory structure on judgment.', *Journal of Experimental Psychology: Learning, Memory, and Cognition*, **14**(3), 521—-533.

Peterson, C. and Beach, L. [1967], 'Man as an intuitive statistician', *Psychological Bulletin*, Elsevier, **68**(1), 29–46.

Petre, M. [1995], 'Why looking isn't always seeing: readership skills and graphical programming', *Communications of the ACM*, **38**(6), 33—-44.

Popper, K. [1972], 'The Logic of Scientific Discovery, 6th impression', *London: Hutchinson*.

Poulton, E. [1976], 'Arousing environmental stresses can improve performance, whatever people say', *Aviation, Space, and Environmental Medicine*, **47**(11), 1193—-1204.

Prossner, P. [1995], MAC-CBJ: maintaining arc-consistency with conflict-directed backjumping, Technical report, Research Report 95 177, Department of Computer Science, University of Strathclyde.

Pu, P. and Chen, L. [2006], *Trust building with explanation interfaces*, ACM Press, New York, New York, USA.

Quesada, J., Kintsch, W. and Gomez, E. [2005], 'Complex problem-solving: a field in search of a definition?', *Theoretical Issues in Ergonomics Science*, Taylor & Francis, **6**(1), 5–33.

Rahman, M. [2007], 'High Velocity Human Factors: Human Factors of Mission Critical Domains in Nonequilibrium', *Human Factors and Ergonomics Society Annual Meeting Proceedings*, **51**(4), 273—-277.

Rahman, M. [2009], 'Understanding Naturalistic Decision Making Under Life Threatening Conditions', *9th Bi-annual International Conference on Naturalistic Decision Making (NDM9)*, (June), 121–128.

Ramezani, M., Bergman, L., Thompson, R. and Burke, R. [2008], 'Selecting and Applying Recommendation Technology', *maya.cs.depaul.edu*.

Rasmussen, J. [1983], 'Skills, rules, and knowledge; signals, signs, and symbols, and other distinctions in human performance models.', *IEEE transactions on systems, man, and cybernetics*, **13**(3), 257—-266.

Ratcliff, R. and Rouder, J. [1998], 'Modeling response times for two-choice decisions', *Psychological Science*, **9**(5), 347—-356.

Rayner, K. [1998], 'Eye movements in reading and information processing: 20 years of research.', *Psychological bulletin*, **124**(3), 372.

Reber, A. [1989], 'Implicit learning and tacit knowledge.', *Journal of experimental psychology: general*, **118**(3), 219—-235.

Reckhow, K. [1994], 'Importance of scientific uncertainty in decision making', *Environmental Management*, **18**(2), 161–166.

Reid, G. and Nygren, T. [1988], 'The subjective workload assessment technique: A scaling procedure for measuring mental workload', *Human mental workload*, **185**, 218.

Reiter, E., Mellish, C. and Levine, J. [1995], 'Automatic generation of technical documentation', *Applied Artificial Intelligence*, **9**(3), 259—-287.

Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P. and Riedl, J. [1994], GroupLens: an open architecture for collaborative filtering of netnews, *in* 'Proceedings of the 1994 ACM conference on Computer supported cooperative work', ACM, pp. 175–186.

Ridge, E. and Kudenko, D. [2007], Screening the parameters affecting heuristic performance, *in* 'Proceedings of the 9th annual conference on Genetic and evolutionary computation', ACM, pp. 180–180.

Riley, V. [1996], 'Operator reliance on automation: Theory and data', *Automation and human performance: Theory and applications*, pp. 19–35.

Roberts, N. and Dotterway, K. [1995], 'The Vincennes incident: Another player on the stage?', *Defense & Security Analysis*, Routledge, **11**(1), 31–45.

Rochart, G. and Jussien, N. [2007], Explanations for global constraints: instrumenting the stretch constraint, Technical report, École des Mines de Nantes, technical report, no. 03-01-INFO.

Rochart, G., Jussien, N. and Laburthe, F. [2003], Challenging explanations for global constraints, *in* 'CP03 Workshop on User-Interaction in Constraint Satisfaction (UICS 03)', Citeseer.

Roscoe, A. [1987], *The practical assessment of pilot workload*, Advisory Group for Aerospace Research and Development, AGARD-AG- 282. Neuilly Sur Seine, France.

Roscoe, A. and Ellis, G. [1990], A subjective rating scale for assessing pilot workload in flight: A decade of practical use, Technical report, Royal Aerospace Establishment.

Ross, L., Lepper, M. and Hubbard, M. [1975], 'Perseverance in self-perception and social perception: Biased attributional processes in the debriefing paradigm', *Journal oj Personality and Social Psychology*, **32**(5), 880–802.

RTCA Inc. [1992], 'RTCA/DO-178B: Software Considerations in Airborne Systems and Equipment Certification', *Washington DC: RTCA Inc*.

Rushby, J. [2002], 'Modular certification', *NASA Contractor Report CR-2002-212130, NASA*, (December).

Russo, J. and Dosher, B. [1983], 'Strategies for multiattribute binary choice', *Journal of Experimental Psychology: Learning, Memory, and Cognition*, **9**(4), 676—-696.

Saaty, T. [1980], *The analytic hierarchy process: planning, priority setting, resource allocation.*, McGraw-Hill, New York.

Saikayasit, R. and Sharples, S. [2009], 'The Influence of Shared-Representation on Shared Mental Models in Virtual Teams', *Engineering Psychology and Cognitive Ergonomics*, pp. 269–278.

Salas, E., Bowers, C. and Cannon-Bowers, J. [1995], 'Military team research: 10 years of progress', *Military Psychology*, Routledge, **7**(2), 55–75.

Salas, E. and Cannon-Bowers, J. [2003], 'The science of training: A decade of progress', *Annual review of psychology*, Annual Reviews 4139 El Camino Way, PO Box 10139, Palo Alto, CA 94303-0139, USA, **52**(1), 471—-499.

Salas, E., Driskell, J. and Hughes, S. [1996], 'The study of stress and human performance', *Stress and human performance(A 97-27090 06-53), Mahwah, NJ, Lawrence Erlbaum Associates, Publishers, 1996*, pp. 1—-45.

Salmon, P., Stanton, N., Walker, G. and Green, D. [2006], 'Situation awareness measurement: A review of applicability for C4i environments', *Applied Ergonomics*, **37**(2), 225–238.

Salthouse, T. [1992], 'Cognition and Context.', *Science (New York, NY)*.
URL: *http://www.ncbi.nlm.nih.gov/pubmed/17789642*

Salvendy, G. [1997], *Handbook of human factors and ergonomics*, John Wiley \& Sons New York.

Sarma, V. and Raju, S. [1991], 'Multisensor data fusion and decision support for airborne target identification', *IEEE Transactions on systems, man, and cybernetics*, **5**, 1224–1230.

Sarter, N. [1995], "Knowing when to look where" - Attention allocation on advanced automated flight decks, *in* 'International Symposium on Aviation Psychology, 8 th, Columbus, OH', pp. 239–242.

Sarter, N. B. and Schroeder, B. [2001], 'Supporting Decision Making and Action Selection under Time Pressure and Uncertainty: The Case of In-Flight Icing', *The Journal of the Human Factors and Ergonomics Society*, Human Factors and Ergonomics Society, **43**(4), 573–583.

Sarter, N. and Woods, D. [1994], 'Pilot Interaction With Cockpit Automation II: An Experimental Study of Pilots' Model and Awareness of the Flight Management System', *The International Journal of Aviation Psychology*, Taylor & Francis, **4**(1), 1–28.

Sarter, N. and Woods, D. [1995], 'Strong, Silent and Out-of-the-Loop: Properties of advanced (cockpit) automation and their impact on human-automation interaction', *Cognitive Systems Engineering Laboratory, Ohio State University, Columbus, OH, Technical Report CSEL*.

Sarter, N., Woods, D. and Billings, C. [1997], 'Automation surprises', *Handbook of human factors and ergonomics*.

Sarwar, B., Karypis, G., Konstan, J. and J [2001], 'Item-based collaborative filtering recommendation algorithms', *Proceedings of the 10th International Conference on World Wide Web*, ACM Press, pp. 285–295.

Scerbo, M. [1996], 'Theoretical perspectives on adaptive automation', *Automation and human performance: Theory and applications (Human Factors in Transportation)*, Lawrence Erlbaum Associates, pp. 37–63.

Schie, E. V. and Pligt, J. V. D. [1995], 'Influencing risk preference in decision making: The effects of framing and salience', *Organizational Behavior and Human Decision Processes*, **63**(3), 264—-275.

Schiex, T. [2005], *Soft Constraints Processing*, INRA, Toulouse.

Schraagen, J., Militello, L. and Ormerod, T. [2008], *Naturalistic decision making and macrocognition*, Ashgate Pub Co.

Schrage, D. and Vachtsevanos, G. [1999], 'Software-enabled control for intelligent UAVs', *Proceedings of the 1999 IEEE International Symposium on Computer Aided Control System Design (Cat. No.99TH8404)*, Ieee, **10**(1.21), 528–532.

Schubert, M., Felfernig, A. and Mandl, M. [2010], 'FastXplain: Conflict Detection for Constraint-Based Recommendation Problems', *Trends in Applied Intelligent Systems*, Springer, pp. 621–630.

Sentz, K., Student, P. and Ferson, S. [2002], *Combination of Evidence in Dempster-Shafer Theory*, Sandia National Labs., Albuquerque, NM (US); Sandia National Labs., Livermore, CA (US).

Shafer, G. [1976], *A Mathematical Theory of Evidence*, Princeton University Press.

Shanteau, J. [1992], 'Competence in experts: The role of task characteristics', *Organizational Behavior and Human Decision Processes*, ACADEMIC PRESS INC, **53**, 252–252.

Shardanand, U. and Maes, P. [1995], Social information filtering: algorithms for automating 'word of mouth', *in* 'Proceedings of the SIGCHI conference on Human factors in computing systems', ACM Press/Addison-Wesley Publishing Co., pp. 210–217.

Sharples, S., Balfe, N., Golightly, D. and Millen, L. [2009], 'Understanding the Impact of Rail Automation', *Human Factors*, pp. 590–599.

Sharples, S., Cobb, S., Moody, A. and Wilson, J. [2008], 'Virtual reality induced symptoms and effects (VRISE): Comparison of head mounted display (HMD), desktop and projection display systems', *Displays (Elsevier)*, **29**(2), 58–69.

Sheridan, T., Verplank, W. and Brooks, T. [1978], Human and Computer Control of Undersea Teleoperators, *in* 'The 14th Annual Conference on Manual Control', p. 343.

Shortliffe, E. H., Buchanan, B. G. and Feigenbaum, E. A. [1979], 'Knowledge engineering for medical decision making: A review of computer-based clinical decision aids', *Proceedings of the IEEE*, **67**(9).

SICStus Prolog [2010], 'SICStus Prolog'.
URL: *www.sics.se/sicstus*

Simmonds, S. and Nesterov, S. [2010], 'Evolution of Complex Safety-Critical Avionics Systems in an NCW Environment', *defence.gov.au*, **18**(2), 229–254.

Simon, H. [1955], 'A behavioral model of rational choice', *The quarterly journal of economics*, Oxford University Press, **69**(1), 99–118.

Singer, G. and Dekker, S. [2000], 'Pilot performance during multiple failures: An empirical study of different warning systems', *Transportation Human Factors*, Lawrence Erlbaum, **2**(1), 63–76.

Skitka, L., Mosier, K. and Burdick, M. [1999], 'Does automation bias decision-making?', *International Journal of Human Computer Studies*, **51**(5), 991–1006.

Skov, R. and Sherman, S. [1986], 'Information-gathering processes: Diagnosticity, hypothesis-confirmatory strategies, and perceived hypothesis confirmation', *Journal of Experimental Social Psychology*, Elsevier, **22**(2), 93–121.

Slamecka, N. and Graf, P. [1978], 'The generation effect: Delineation of the phenomenon', *Journal of Experimental Psychology*, **16**, 272–279.

Slovic, P. [1972], 'From Shakespeare to Simon: Speculations and some evidence about man's ability to process information', *Oregon Research Institute Bulletin*.

Slovic, P. [1975], 'Choice between equally valued alternatives', *Journal of Experimental Psychology: Human Perception and Performance*, **1**(3), 280—-287.

Slovic, P. [1987], 'Perception of risk science', *Science*, **236**, 280—-285.

Slovic, P. [1999], 'Trust, emotion, sex, politics, and science: surveying the risk-assessment battlefield.', *Risk analysis : an official publication of the Society for Risk Analysis*, **19**(4), 689–701.

Smart Eye AB [2010], 'Smart-Eye (http://www.smarteye.se)'.
URL: *http://www.smarteye.se*

Smets, P. [1994], 'What is Dempster-Shafer's model?', *Advances in the Dempster-Shafer theory of evidence*.

Smith, B. [2005], 'Modelling for constraint programming', *Lecture Notes for the First International Summer School on Constraint Programming. Available at: http://www. math. unipd. it/frossi/cp-school*, Citeseer, (September).

Smith, E. and Kosslyn, S. [2007], *Cognitive Psychology: Mind and Brain*, Pearson Prentice Hall, Upper Saddle River, NJ.

Smith, J. and Kida, T. [1991], 'Heuristics and biases: Expertise and task realism in auditing.', *Psychological Bulletin*, **109**(3), 472–489.

Smith, K. and Hancock, P. [1995], 'Situation awareness is adaptive, externally directed consciousness', *Human Factors: The Journal of the Human Factors and Ergonomics Society*, Human Factors and Ergonomics Society, **37**(1), 137–148.

Smith, P. and Geddes, N. [2003], 'A cognitive systems engineering approach to the design of decision support systems', *Handbook of human-computer interaction. Lawrence Erlbaum, Mahwah, NJ*, pp. 656–675.

Smith, S., Cortellessa, G., Hildum, D. and Ohler, C. [2005], 'Using a scheduling domain ontology to compute user-oriented explanations', *Planning, Scheduling and Constraint Satisfaction: From Theory to Practice*, IOS Press, p. 179.

Snyder, M. and Swann, W. [1978], 'Hypothesis-testing processes in social interaction', *Journal of Personality and Social Psychology*, **36**(11), 1202—-1212.

Spitzer, C. [2000], *The avionics handbook*, CRC Press.

Sqalli, M. and Freuder, E. [1996], Inference-based constraint satisfaction supports explanation, *in* 'Proceedings of the National Conference on Artificial Intelligence', pp. 318–325.

Staab, S., Studer, R., Schnurr, H. and Y [2005], 'Knowledge processes and ontologies', *Intelligent Systems,*, **16**(1), 26—-34.

Stallman, R. and Sussman, G. [1977], 'Forward reasoning and dependency-directed backtracking in a system for computer-aided circuit analysis', *Artificial intelligence*, Elsevier, **9**(2), 135–196.

Stanford University [2010], 'Stanford Encyclopedia of Philosophy'.
URL: *http://plato.stanford.edu*

Stanton, N. [2001], 'Situational awareness and safety', *Safety Science*, **39**(3), 189–204.

Stanton, N. [2005], *Human factors methods: a practical guide for engineering and design*, Ashgate Publishing.

Staw, B., Sandelands, L. and Dutton, J. [1981], 'Threat rigidity effects in organizational behavior: A multilevel analysis', *Administrative science quarterly*, pp. 501—-524.

Stephenson, Z. R., Nicholson, M. and McDermid, J. A. [2005], Product-Line Technology Recommendations for Integrated Modular Systems, *in* 'Proceedings of the International System Safety Conference'.

Stephenson, Z. R., Nicholson, M. and McDermid, J. A. [2006], Flexibility and Manageability of IMS Projects, *in* 'Proceedings of the 24th International System Safety Conference', Albuquerque, New Mexico, USA.

Stern, G., McCants, T. and Pettine, P. [1982], 'Stress and illness: Controllable and uncontrollable life events' relative contributions', *Personality and Social Psychology Bulletin*, SPSP, **8**(1), 140.

Stokes, A., Barnett, B. and Wickens, C. [1987], Modeling stress and bias in pilot decision-making, *in* 'Proceedings of Annual Conference of the HF Association of Canada'.

Stokes, A. and Kite, K. [1994], *Flight stress: Stress, fatigue, and performance in aviation*, Avebury Aviation.

Storbeck, J. and Clore, G. [2007], 'On the interdependence of cognition and emotion', *Cognition & emotion*, **21**(6), 1212—-1237.

Strat, T. [1987], The generation of explanations within evidential reasoning systems, *in* 'Proceedings of the 10th International Joint Conference on Artificial Intelligence, Milan, Italy', Citeseer, pp. 1097–1104.

Strunk, E. and Knight, J. [2004], 'Assured Reconfiguration of Embedded Real-Time Software', *Proc. International Conference on Dependable Systems and Networks*, pp. 367–376.

Strunk, E., Knight, J. and Aiello, M. [2004], 'Distributed reconfigurable avionics architectures', *Digital Avionics Systems Conference, 2004. DASC*.

Studer, R. [1998], 'Knowledge engineering: Principles and methods', *Data & Knowledge Engineering*, **25**(1-2), 161–197.

Subbarayan, S., Jensen, R., Hadzic, T., Andersen, H., Hulgaard, H. and Mø ller, J. [2004], Comparing two implementations of a complete and backtrack-free interactive configurator, *in* 'Proceedings of the CP-04 Workshop on CSP Techniques with Immediate Application', Citeseer, pp. 97–111.

Sutterfield, B., Hoschette, J. A. and Anton, P. [2008], 'Future integrated modular avionics for jet fighter mission computers', *2008 IEEE/AIAA 27th Digital Avionics Systems Conference*, IEEE, pp. 1.A.4–1–1.A.4–11.

Swartout, B., Patil, R., Knight, K. and Russ, T. [1996], 'Toward distributed use of large-scale ontologies', *Proc. of the Tenth Workshop on*, pp. 1–20.

Taylor, P., Mosier, K. L., Skitka, L. J., Heers, S., Burdick, M. and Field, M. [2009], 'Cockpits Automation Bias : Decision Making and Performance in High-Tech Cockpits', *The International journal of aviation psychology*, **8**(1), 47—-63.

The Boeing Company [2002], *Boeing 737-300/-400/-500 Operations Manual*, The Boeing Company.

TheFreeDictionary [2010], 'The Free Dictionary'.
URL: *http://www.thefreedictionary.com*

Thrustmaster [2010], 'Thrustmaster T-Flight Hotas X'.
URL: *http://www.thrustmaster.com*

Tiedemann, P., Hadzic, T., Henney, T. and Andersen, H. [2006], 'Interactive distributed configuration', *Principles and Practice of Constraint Programming*, Springer, pp. 761–765.

Tobii Technology [2010], 'TOBII Eye-Tracking System (www.tobii.com)'.

Tom, S., Fox, C., Trepel, C. and Poldrack, R. [2007], 'The neural basis of loss aversion in decision-making under risk', *Science*, AAAS, **315**(5811), 515.

Trapp, M. and Schürmann, B. [2003], On the modeling of adaptive systems, *in* 'International Workshop on Dependable Embedded Systems', Citeseer.

Trappenberg, T. [2010], *Fundamentals of computational neuroscience*, Oxford Univ Press, Oxford.

Trepel, C., Fox, C. and Poldrack, R. [2005], 'Prospect theory on the brain? Toward a cognitive neuroscience of decision under risk', *Cognitive Brain Research*, Elsevier, **23**(1), 34–50.

Trujillo, A., Bruneau, D. and Press, H. [2008], 'Predictive information: Status or alert information?', *2008 IEEE/AIAA 27th Digital Avionics Systems Conference*, Ieee, pp. 4.A.4–1–4.A.4–9.

Tsang, E. [1993], *Foundations of constraint satisfaction*, Academic press London.

Tversky, A. and Kahneman, D. [1973], 'Availability: A heuristic for judging frequency and probability', *Cognitive psychology*, **5**(2), 207–232.

Tversky, A. and Kahneman, D. [1984], 'Choices, values, and frames', *American Psychologist*, **39**(4), 341—-350.

Tversky, A. and Kahneman, D. [1991], 'Loss aversion in riskless choice: A reference-dependent model', *The Quarterly Journal of Economics*, **106**(4), 1039–1061.

Tversky, A. and Kahneman, D. [1992], 'Advances in prospect theory: Cumulative representation of uncertainty', *Journal of Risk and uncertainty*, Springer, **5**(4), 297–323.

Tversky, A. and Koehler, D. [1994], 'Support theory: A nonextensional representation of subjective probability.', *Psychological Review*, **101**(4), 547–567.

Tversky, A., Sattath, S. and Slovic, P. [1988], 'Contingent weighting in judgment and choice.', *Psychological review*, **95**(3), 371–384.

UK Ministry of Defence [1966], 'ASAAC 00-74', **16**(6), 338–338.

Vamanu, D., Slavnicu, S., Slavnicu, E. and Vamanu, B. [2004], 'Decision support systems in nuclear emergencies: a scenario-based comparison of domestic and reference tools', *Radiation protection*.

Van Charante, E., Cook, R., Woods, D., Yue, L. and Howie, M. [1992], Human-computer interaction in context: Physician interaction with automated intravenous controllers in the heart room, *in* 'Proceedings of the Fifth IFAC/IFIP/IFORS/IEA Symposium on Analysis, Design and Evaluation of Man-Machine Systems. The H-ague, Netherlands'.

van der Linden, J. [2001], 'Assigning Satisfaction Values to Constraints: An Algorithm to Solve Dynamic Meta-Constraints', *Arxiv preprint cs/0109014*.

Van Der Linden, J. [2002], 'Meta-Constraints to Aid Interaction and to Provide Explanations.', *Technology*, Citeseer.

van der Meer, E. R., Wasowski, A. and Andersen, H. R. [2006], Efficient interactive configuration of unbounded modular systems, *in* 'Proceedings of the 2006 ACM symposium on Applied computing - SAC '06', ACM Press, New York, New York, USA, p. 409.

Vidulich, M. and Hughes, E. [1991], 'Testing a subjective metric of situation awareness', *Human Factors and Ergonomics Society Annual Meeting*, Human Factors and Ergonomics Society, **35**(18), 1307–1311.

Vidulich, M., Ward, F. and Schueren, J. [1991], 'Using the subjective workload dominance (SWORD) technique for projective workload assessment', *Human Factors: The Journal of the Human Factors and Ergonomics Society*, **33**(6), 677—-691.

Visentin, G. [2007], Autonomy in ESA Planetary Robotics Missions, *in* 'ESA Workshop on Autonomous Systems'.

Wallace, R. and Freuder, E. [2001], Explanations for whom, *in* 'Proceedings of CP 2001 Workshop on User-Interaction in Constraint Satisfaction'.

Watkins, C., LLC, S. and Rapids, G. [2006], Integrated modular avionics: managing the allocation of shared intersystem resources, *in* '2006 IEEE/AIAA 25th Digital Avionics Systems Conference', pp. 1–12.

Weaver, W. [1948], 'Science and complexity', *American Scientist*, **36**(4), 536—-544.

Weenig, M. and Maarleveld, M. [2002*a*], 'The impact of time constraint on information search strategies in complex choice tasks', *Journal of Economic Psychology*, Elsevier, **23**(6), 689–702.

Weenig, M. and Maarleveld, M. [2002*b*], 'The impact of time constraint on information search strategies in complex choice tasks', *Journal of Economic Psychology*, **23**, 689–702.

Weibenbacher, G., Herzner, W. and Althammer, E. [2005], Allocation of Dependable Software Modules under Consideration of Replicas, *in* 'ERCIM Workshop on Dependable Software Intensive Embedded Systems, Porto, Portugal', number 511764.

Weller, J., Levin, I., Shiv, B. and Bechara, A. [2007], 'Neural correlates of adaptive decision making for risky gains and losses', *Psychological Science*, SAGE Publications, **18**(11), 958.

Wickens, C. and Flach, J. [1988], 'Information processing', *Human factors in aviation*, pp. 111—-155.

Wiener, E. [1988], 'Cockpit automation', *Human factors in aviation*, pp. 433–461.

Wierwille, W. and Eggemeier, F. [1993], 'Recommendations for mental workload measurement in a test and evaluation environment', *Human Factors: The Journal of the Human Factors and Ergonomics Society*, Human Factors and Ergonomics Society, **35**(2), 263–281.

Wills, L., Kannan, S., Heck, B., Vachtsevanos, G., Restrepo, C., Sander, S., Schrage, D. and Prasad, J. [2000], 'An open software infrastructure for reconfigurable control systems', *Proceedings of the 2000 American Control Conference. ACC (IEEE Cat. No.00CH36334)*, American Autom. Control Council, (June), 2799–2803.

Woodcock, J. and Davies, J. [1996], *Using Z: specification, refinement, and proof*, Vol. 39, Prentice Hall.

Woods, D. [1994], 'Cognitive demands and activities in dynamic fault management: abductive reasoning and disturbance management', *Human factors in alarm design*, pp. 63–92.

Woods, D. [1996], 'Decomposing automation: Apparent simplicity, real complexity', *Automation and human performance: Theory and applications*, pp. 3–17.

Woods, D., Johannesen, L., Cook, R. and Sarter, N. [1994], 'Behind human error: Cognitive systems, computers, and hindsight', *State-of-the-Art Report SOAR*, pp. 94–01.

Woods, D., Patterson, E. and Roth, E. [2002], 'Can we ever escape from data overload? A cognitive systems diagnosis', *Cognition, Technology & Work*, **4**(1), 22—-36.

Woolfolk, R., Parrish, M. and Murphy, S. [1985], 'The effects of positive and negative imagery on motor skill performance', *Cognitive Therapy and Research*, **9**(3), 335—-341.

Yager, R. [1987], 'On the Dempster-Shafer framework and new combination rules', *Information Sciences: an International Journal*, **41**(2), 93–137.

Yellott, J. [1971], 'Correction for fast guessing and the speed-accuracy tradeoff in choice reaction time', *Journal of Mathematical Psychology*, **8**(159–199).

Yerkes, R. and Dodson, J. [1908], 'The relation of strength of stimulus to rapidity of habit-formation', *Journal of comparative neurology and psychology*, Wiley Online Library, **18**(5), 459–482.
URL: *http://onlinelibrary.wiley.com/doi/10.1002/cne.920180503/abstract*

Young, L. [1983], 'Right-brained decision support systems', *ACM SIGMIS Database*, **14**(4), 36.

Young, M., Stanton, N. and Harris, D. [2007], 'Driving automation: learning from aviation about design philosophies', *Journal of Vehicle Design*, **44**(0), 0–29.

Younis, M., Zhou, J. and McElroy, J. [2000], 'Resource scheduling in dependable integrated modular avionics', *Proceeding International Conference on Dependable Systems and Networks. DSN 2000*, IEEE Comput. Soc, pp. 14–23.

Yu, B., Sycara, K., Giampapa, J. and Owens, S. [2004], 'Uncertain information fusion for force aggregation and classification in airborne sensor networks', *American Association for Artificial Intelligence*.

Zadeh, L. [1979], *On the validity of Dempster's rule of combination of evidence*, Vol. 79, Electronics Research Laboratory, Univ. of California.

Zakay, D. [1993], 'The impact of time perception processes on decision making under time stress', *Time pressure and stress in human judgment and decision making*, pp. 59—-72.

Zakay, D. and Wooler, S. [1984], 'Time pressure, training and decision effectiveness', *Ergonomics*, Taylor & Francis, **27**(3), 273–284.

Zanker, M. [2008], 'A collaborative constraint-based meta-level recommender', *Proceedings of the 2008 ACM conference on Recommender*, pp. 139—-146.

Zielinski, P. [2009], 'OpenGazer - http://www.inference.phy.cam.ac.uk/opengazer/'.

Zsambok, C., Beach, L. and Klein, G. [2002], A literature review of analytical and naturalistic decision making, Technical report.

Zsambok, C. and Klein, G. [1997], *Naturalistic decision making*, Lawrence Erlbaum.