Contract no. AST3-CT-2003-501848

ISAAC

Improvement of **S**afety **A**ctivities on **A**eronautical **C**omplex Systems

Instrument: Specific Targeted Research Project

Thematic Priority: Aeronautics and Space

Publishable Final Report

Period covered: from 01.02.2004 to 31.01.2007 Date of preparation: 28/02/2007

Start date of project: 01.02.2004 Duration: 36 months

Project coordinator name: Antonella Cavallo
Project coordinator organisation name: Alenia Aeronautica S.p.A.

Partners: Airbus France, Airbus UK Ltd., Airbus Deutschland GmbH, SAAB AB, AleniaSIA, Istituto Trentino di Cultura, Office National d'Etudes et de Recherches Aérospatiales, OFFIS e.V., Prover Technology AB, Dassault Aviation

Project co-funded by the European Commission within the Sixth Framework Programme (2002-2006)	
Dissemination Level	
PU	Public

ISAAC Document Reference: ISAAC/01/000044/A/WP6/MRP
Document Issue: A
Partner Internal Document Reference: 65/NT/0000/T810d/070071

DISTRIBUTION LIST		
Copy type [1]	Company and Location	Recipient
@, C, D	European Commission Directorate General RTD Directorate H : Space & Transport Unit H3: "Aeronautics" CDMA 4/169 21, rue Champ de Mars B-1049 Brussels	M. Brusati
ftp	All ISAAC Partners via ftp repository. Notification via reflector.	All ISAAC Partners

[1] M = Master copy, @ = Email, C = Controlled copy (paper), D = Electronic copy on disk

RECORD OF REVISION		
Issue	Date	Reason for Revision
A	28/02/2007	Issue A

Table of Contents

DISTRIBUTION LIST **2**

RECORD OF REVISION **3**

1. LIST OF ABBREVIATIONS **6**

2. LIST OF REFERENCES **8**

3. INTRODUCTION **9**

4. PROJECT EXECUTION **9**

4.1 Publishable executive summary 9

4.2 Methodologies, Impacts and Applications 16

4.2.1 EPC – ESACS Platform Consolidation 17

4.2.1.1 Summary 17

4.2.1.2 Methodology 17

4.2.1.3 Impact and Applications 18

4.2.2 HLR – High Level Representation 19

4.2.2.1 Summary 19

4.2.2.2 Methodology 19

4.2.2.3 Impact and Applications 19

4.2.3 SAP – Safety Architecture Patterns 20

4.2.3.1 Summary 20

4.2.3.2 Methodology 20

4.2.3.3 Impact and applications 22

4.2.4 TQA – Timing and Quantitative Analysis 24

4.2.4.1 Summary 24

4.2.4.2 Methodology 24

4.2.4.3 Impact and Applications 25

4.2.5 CCA – Common Cause Analysis 26

4.2.5.1 Summary 26

4.2.5.2 Methodology 26

4.2.5.3 Impact and Applications 29

4.2.6 HEA - Human Error Analysis 31

4.2.6.1 Summary 31

4.2.6.2 Methodology 34

4.2.6.3 Impact and Applications 38

4.2.7 MRA – Mission Reliability Analysis 40

4.2.7.1 Summary 40

4.2.7.2 Methodology 40

4.2.7.3 Impact and Applications 43

4.2.8 TDS – System Testability/Diagnosability 45

4.2.8.1 Summary 45

4.2.8.2 Methodology 46

4.2.8.3 Impact and Applications 49

***5.* DISSEMINATION AND USE** **50**

6. CONCLUSION.....60

1. List of abbreviations

A/C	Aircraft
AIF	Airbus France (France)
Airbus D	Airbus Deutschland (Germany)
ALA	Alenia Aeronautica S.p.A. (Italy)
AUK	Airbus UK (United Kingdom)
AP	Auto Pilot
ARP	Aerospace Recommended Practice (SAE)
CCA	Common Cause Analysis
CMP	Commonalities on Methodology and Process
DASSAV	Dassault Aviation (France)
EPC	ESACS Platform Consolidation
ESACS	Enhanced Safety Assessment for Complex Systems (FP5 project)
FAA	Federal Aviation Administration
FAR	Federal Aviation Regulation
GOMS	Goals, Operators, Methods and Selection rules
GSM	Goal State Means
HEA	Human Error Analysis
HLR	High Level Representation, UML, Mode Logic
HMI	Human Machine Interface
IP	Industrial Partner
ITA	Integrability: Translators and Algorithms, Libraries
ITC-IRST	Istituto Trentino di Cultura- Istituto per la Ricerca Scientifica e Tecnologica (Italy)
JAA	Joint Aviation Authorities
JAR	Joint Aviation Requirements
KM	Knowledge Management
MRA	Mission Reliability Analysis
OFFIS	OFFIS e.V. (Germany)
ONERA	Office National d'Etudes et de Recherches Aérospatiales (France)
PROVER	Prover Technology AB (Sweden)
RTCA	(Previously) Radio Technical Commission for Aeronautics
SAAB	Saab AB (Sweden)
SAE	Society of Automotive Engineers, Inc.
SAP	Safety Architecture Patterns
SIA	AleniaSIA S.p.A. (Italy)
SOAR	States, Operators And Reasoning
TDS	Systems Testability/ Diagnosability
TP	Technology Partner
TQA	Timing/ Quantitative Analysis

UML	Unified Modeling Language
WP	Work Package

2. List of references

1. ISAAC Annex I to the Contract AST3-CT-2003-501848 “Description of Work”, dated 10/10/2003 (Confidential)

3. Introduction

This document consists in the ***final activity report*** containing a summary of the project activities and results over the full duration.

4. Project execution

This section includes summary description of project objectives, contractors involved, work performed and end results, elaborating on the degree to which the objectives were reached.
It briefly describes the methodologies and approaches employed and relates the achievements of the project to the state-of-the-art.
It explains the impact of the project on its industry or research sector.

4.1 Publishable executive summary

A publishable executive summary of the ISAAC project is reported in the following.

Title: Improvement of Safety Activities on Aeronautical Complex Systems

Acronym:	ISAAC		
Contract Nr.:	AST3-CT-2003-501848		
Total Cost:	9.496.751 €	**EU Contribution:**	5.361.941 €
Starting Date:	01/02/2004	**Duration:**	36 months
Web-site:	www.isaac-fp6.org		

Project logo:

Coordinator: *Dr. Antonella Cavallo*
Tel.: +39 011 9960 508
Fax: +39 011 9960 515
E-mail: acavallo@aeronautica.alenia.it
Organisation: Alenia Aeronautica S.p.A.
Site: Caselle Sud
Department: Aeronavigabilita' ed Efficacia del Sistema Torino
Strada Malanghero 17, IT-10072 Caselle, Torino, Italy

EC Officer: *Dr. Marco Brusati*
Tel.: +32 2 29 948 48
Fax: +32 2 29 667 57
E-mail Marco.Brusati@cec.eu.int

Partners (name, country code):

Airbus France	(FR)
Airbus UK Ltd.	(UK)
Airbus Deutschland GmbH	(DE)
Saab AB	(SE)
AleniaSIA	(IT)
Istituto Trentino di Cultura	(IT)
Office National d'Etudes et de Recherches Aérospatiales	(FR)
OFFIS e. V.	(DE)
Prover Technology AB	(SE)
Dassault Aviation	(FR)

Priority /Priority Component (e.g. Strategic Objective, etc.)

The project answers to the FP6 Thematic Priority 4 Aeronautics and Space, top level objective:

(1.3.1.3) "**Improve Aircraft Safety and Security**"

The development of the methods and tools for the Safety Activities foreseen in ISAAC will help the European Engineering Capability to achieve the following policy objectives:

- the reduction of the accident rate by 50% and 80% in the short and long term respectively
- to obtain 100% capability for avoiding or recovering from human errors.

Background

Avionic systems are becoming more complex (heterogeneous components, large functions number, interaction with operators through advanced interfaces). Therefore it is becoming harder to manage all aspects of safety assessment and to maintain the required safety levels.
A FP5 previous project ESACS (Enhanced Safety Assessment for Complex Systems, www.esacs.org) has shown the benefit of using formal techniques to assess aircraft safety.
ISAAC builds upon and extends the ESACS results to go a step further into the improvement and integration of safety activities of aeronautical complex systems.

Project Objectives

ISAAC project aims to increase the capability and efficiency for safety and systems engineers to perform safety assessment resulting in safe systems. The proposed methodology, built on formal method techniques, is an integrated part in a model based development process where safety and reliability aspects are examined in early steps of development.
A goal is to consolidate the ESACS results by improving analysis for dynamic aspects like sequencing or temporal behavior.
Another goal is to extend the scope of the integrated environment among designers and safety/reliability engineers.
To take into account results from tools used in performing particular risk and zonal safety analysis and to use this information to inject unintended interactions within "intended-functionality"-independent but co-located systems. To evaluate the relationship between the human and the machine offering a complex human - complex machine interaction model. To automate the analyses to determine the impact of degraded situations on system operating modes and over pre-defined missions. To exploit the use of ESACS formal verification techniques to deal with testability aspects.

Description of the work

To reach the above goals, the ISAAC work has followed detailed technical and scientific objectives organized into three complementary dimensions, which are structured into basic topics.

First dimension: Consolidation of ESACS work
Including the following topics: integration with higher level notations for requirements, extension of traditional techniques to timing aspects and quantitative analysis, further development of platform/tools already started in ESACS.

Second dimension: Extension to other safety related aspects
Including the following topics: Human Errors, Common Cause Analysis, Mission Analysis and Testability.

Third dimension: Commonalities
Common methodological recommendations and common tools and libraries that facilitate exchanges among tools will be identified in order to provide a more comprehensive tool-supported coverage of the safety process.

The areas of investigations are indicated in the following figure.

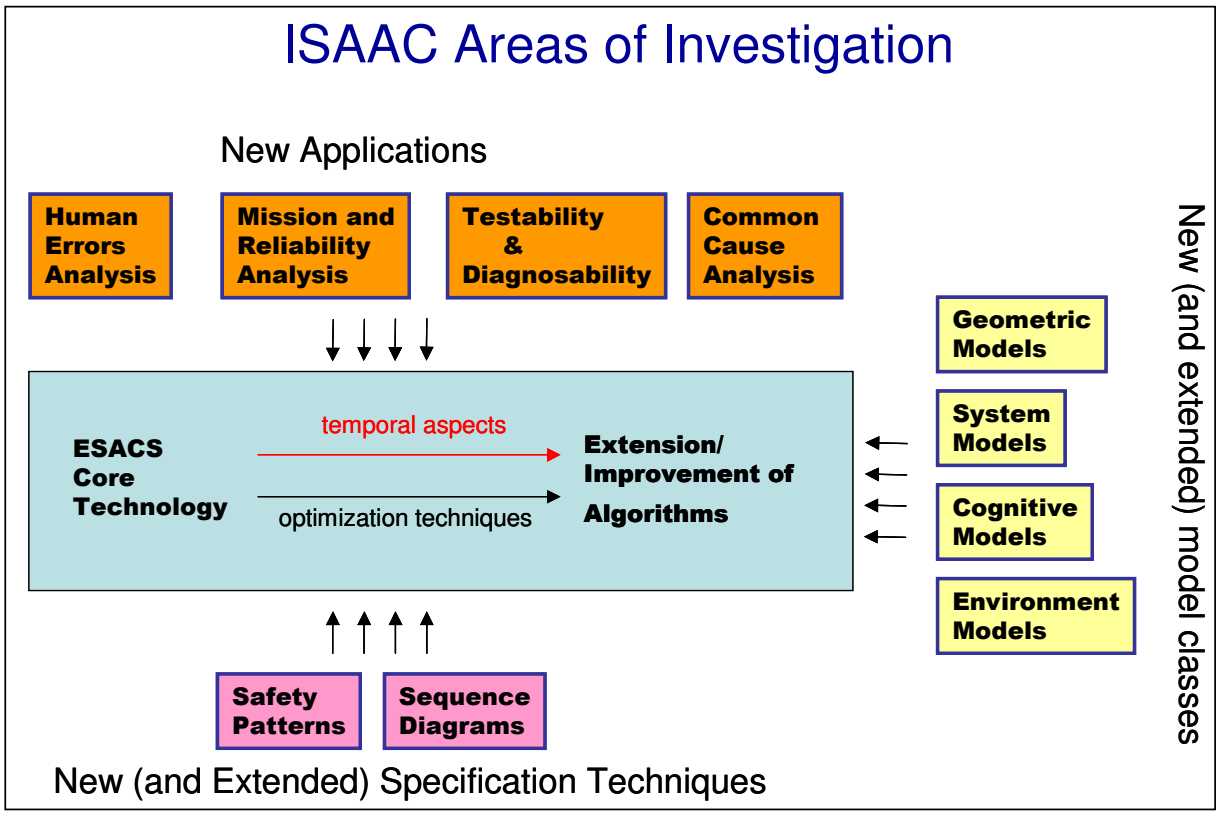

Figure 1 Areas of investigation

Work performed up to the end of the project

Methodologies and tools were developed according to the identified requirements and applied to the case studies identified by the Industrial Partners.

Results

A comprehensive environment including methodologies supported by tools for performing the analyses taking into account the various aspects related to the safety.

The ISAAC framework relies on the use of models normally generated along the product design phases.

They are representative of nominal and failure behaviors:

- functional models
- geometrical models
- risk models
- cognitive pilot models
- failures models

These models are combined with environment models and are elaborated by means of formal verification and simulation techniques to automatically derive safety analyses for the verification of the requirements.

The following picture describes the ISAAC framework.

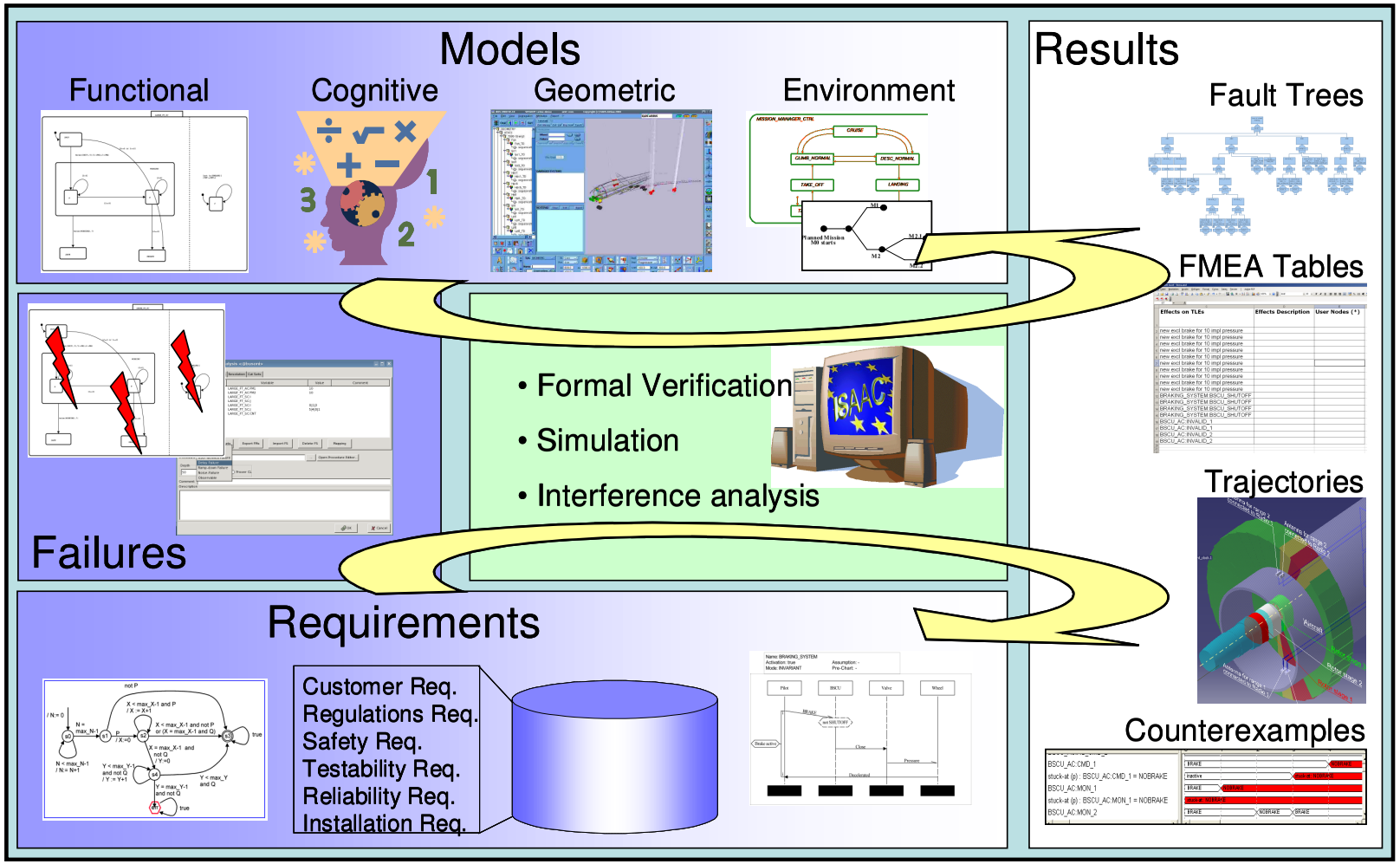

Figure 2 ISAAC Framework

Main benefit is that activities of designing and doing analysis can be performed more easily in an iterative manner resulting in a more effective development process, where the results of the analysis can influence the design in short period of time. Moreover, the traceability of safety issues and of relevant design changes is improved enhancing the visibility in the perspective of the certification process.

The public results are available in the project web site www.isaac-fp6.org.

The following figure describes the main areas of application of the ISAAC results.

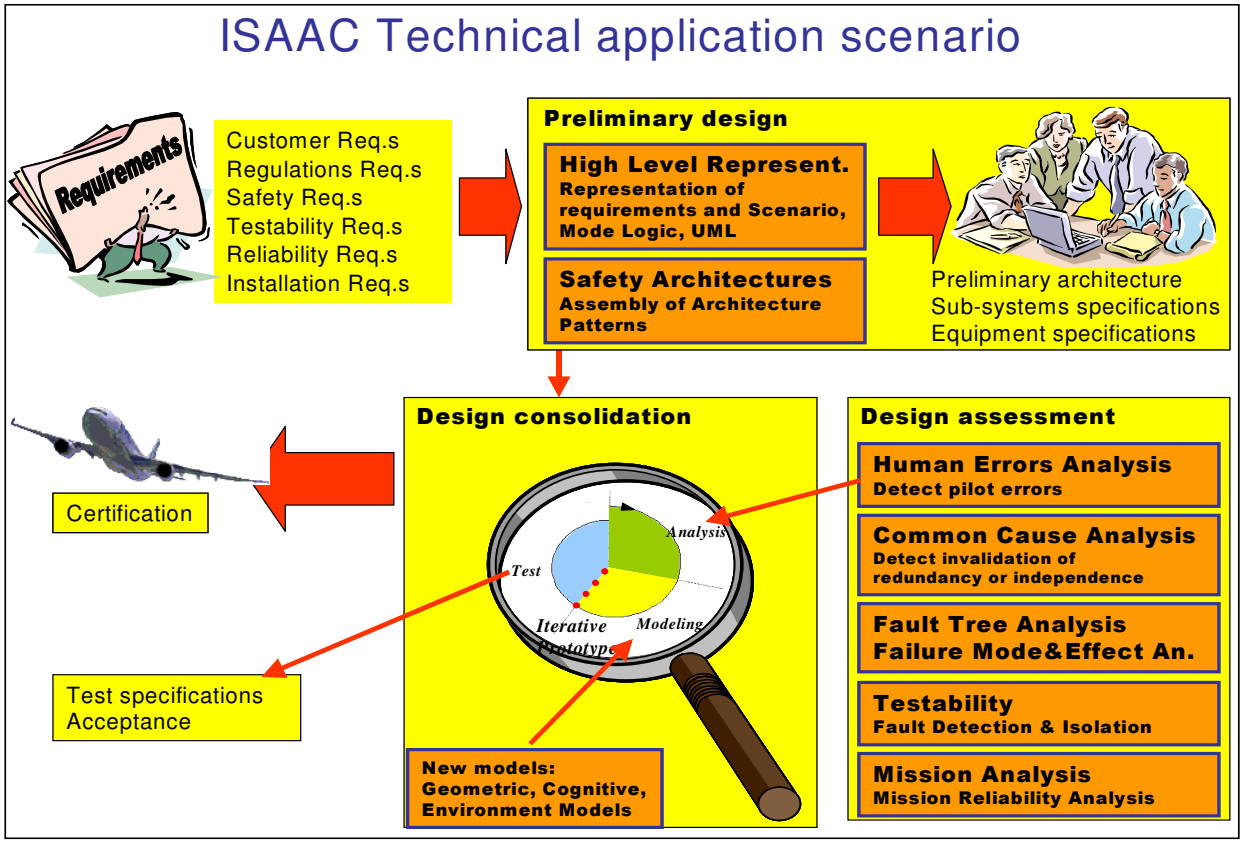

Figure 3 Technical application scenario

4.2 Methodologies, Impacts and Applications

In the following each topic (theme) of the project is presented through:

- the methodology with the approaches employed and achievements related to the state-of-the-art
- the impact on industrial application and research sector.

4.2.1 EPC – ESACS Platform Consolidation

4.2.1.1 Summary

The main objective of the EPC theme was to integrate new analyses into the existing integration lines and update the methodology developed during ESACS to the additional objectives set out for ISAAC. This entails both the further development of the existing methodology / themes and the necessary work to adapt the implementation lines to the new themes of ISAAC.

4.2.1.2 Methodology

This section describes the methodology employed in the EPC theme. Although the underlying basic methodology is the same to all implementation lines, they differ in certain extents due to both the different case tools employed and different focus to the approach by the respective Technological Partners due to the degree of involvement into other themes. A more detailed description for the different implementation lines can be found in the EPC appendix to this document.

The basic methodology, as originally developed in ESACS, is shown in Figure 4. In the first step, a system model is captured in one of the case tools supported by the different implementation lines. In the second step, functional failure modes are captured in the tools provided by the EP. This leads to the Extended System model. In the third step, the user specifies the safety requirements that are to be assessed.

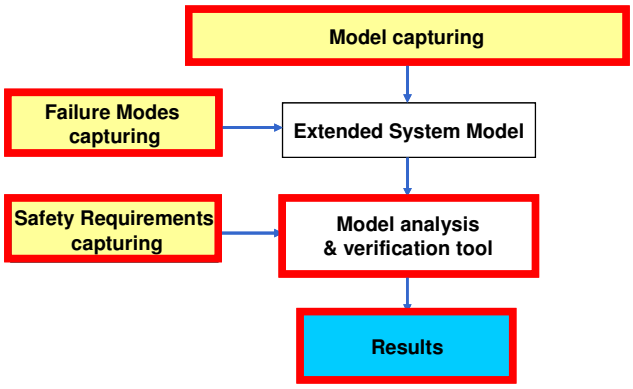

Figure 4: Basic methodology

The safety requirements and the extended system model, together with other settings such as abstraction, which depend on the actual implementation line, form the analysis task which is executed by the underlying formal verification tools in the fourth step. The results emanating from this can then be displayed in traditional safety analysis tools such as FaultTree+ or in Excel tables, depending on the type of analysis carried out.

4.2.1.3 Impact and Applications

The main objective of the EPC theme is the continued development of the implementation platform developed during ESACS. As such the central points are similar to those that were central in ESACS:

- Use the same formal model for the system development and for safety analysis
    - Identify problem in the design as early as possible to avoid costly deep cycles in the development process.
    - Ensure that the model used for safety analysis is identical to the one used for system design.
    - Enable easy re-evolution of a system after design changes have been made.

- Use algorithms based on formal verification techniques to
    - Automate safety analysis wherever possible
    - Evaluate dynamic dependencies between failure modes and events (see also section on TQA)

Within ISAAC one of the main tasks was to implement and integrate the concepts developed in the other themes (e.g. TQA, HLR, CCA, MRA, TDS etc.) into the existing implementation lines. Besides this there were also work done in order to improve the core techniques themselves. A significant improvement has been the possibility to specify intermediate events and determine their interdependencies. This was an essential element in order to improve the FTA and FMEA analyses. For example several alternative approaches have been developed that allow to overcome the ESACS limitation of “flat fault-tree” and to produce meaningful structured fault-trees. Further work includes improvements to cope with complex models and the representation of analysis results.

4.2.2 HLR – High Level Representation

4.2.2.1 Summary

In ESACS we observed that often one of the main problems is to create a consistent definition of the safety analysis task. There exist a large number of techniques for this purpose but often they suffer from one the following limitations. Either they are easy to understand and apply but are not expressive enough to specify complex properties, or they are very powerful (e.g. temporal logic formulas) but require intensive training before they can be used productively.
In the HLR theme we addressed this problem and provided the possibility to use sequence diagrams for the specification of top-level events.

4.2.2.2 Methodology

During the ESACS project it became apparent that there is a need for a better formalism to capture safety requirements. There should be a smooth transition between requirements and the system design. In order to achieve this goal the notation used to capture system and safety requirements should be:

- User friendly so that it can easily be used by the system and safety engineers
- Powerful enough to capture typical requirements, especially system dynamics
- Close to the original formalism used to capture the behavior of the system model to aid in documentation and traceability of results.

The Unified Modeling Language (UML) is a visual modeling language that provides formalisms that meet all of the above criteria. Especially Sequence Diagrams can be used to capture system activity in a way that it is usable for describing system and safety requirements. As Sequences Diagrams have now been integrated into some of the modeling tools used in ISAAC (e.g. Statemate) this helps with the third of the above points. Documentation and traceability are greatly improved when the system behavior and the requirements can be captured and updated together.
Once all requirements have been captured and formalized as Sequences Diagrams it is now possible to use them as candidate event during the automated safety analyses.

When Sequence Diagrams are used to capture requirements they usually describe the intended system behavior, to use them as events (either top-level or intermediate) they have to be converted so that they describe the undesired situation where the requirement is violated. These derived Sequence Diagrams are basically a complement of the original requirement Sequence Diagram and are interpreted in the following way: all Sequence Diagrams that describe candidate events are executed in parallel with the system model. An event is considered to have happened when the complete sequence described in the diagram has been observed.

4.2.2.3 Impact and Applications

Within the High Level Representation (HLR) theme we have investigated methods to improve specification of system properties. Techniques that allow an easier capturing of sequences and time dependencies have been found with sequence diagrams. They can be used to identify sequences of events that then be used to define safety requirements or candidates for intermediate and top-levels events. It has been investigated how they can best be used within the ISAAC setting. For some implementation lines support for sequence diagrams has been integrated and evaluated.

4.2.3 SAP – Safety Architecture Patterns

4.2.3.1 Summary

Safety Architecture Patterns (SAP) objective is to develop concepts, methodologies and tools that enable both the designers and safety engineers to elaborate a preliminary model of an aircraft system by assembling safety architecture patterns. A safety architecture pattern is a piece of architecture linked to good use rules. Such formalism is an efficient way to quickly assess early drafts of system architectures with regard to safety requirements to be fulfilled. This approach also favors the results sharing between the safety engineer and the design engineer since it is based on a common view and understanding of the system both with regard to design constraints and safety requirements.
In brief the SAP purpose is to:

- Provide means to quickly prototype system safety architectures.
- Assist the allocation of safety requirements to the system components.
- Validate the formal safety requirements allocation.

To fulfill these objectives, the methodology proposes capitalizing expert know-how by SAP that are pre-proved and can be safely reused. To do this, engineers must have at their disposal SAP libraries associated with safety properties fulfilled under specified conditions, a methodology to build a SAP-based model and a structure to memorize the choice made during design process.

4.2.3.2 Methodology

Investigations and studies have been performed according to four directions:

- State of art and practices related to safety architectures
- Definition and extension of SAP libraries
- Proposal of a process for using SAPs
- Search to support this process.

In parallel a SAP editor has been prototyped and a support to structure requirements (so called “companion structure”) was initiated.

4.2.3.2.1 State of art and practices related to safety architectures

Since the first research works on fault-tolerance and the use of redundancy, many applications and studies have been driven. We first of all made a review of fault tolerant mechanisms used in safety architecture. Only a very limited work identical to ours had been already undertaken and reported in the literature. Few papers were found on behavioral formal models of safety architecture but not enough explicit on the safety features of the patterns.
One second acknowledgment comes from the important difference between the number and the variety of fault tolerant architectures presented in the literature and the low number of principles used for their construction: Detection, masking, reconfiguration. In practice the SAPs can be built by combining these three principles astutely.

4.2.3.2.2 Definition and extension of SAP libraries

The main goal of SAP is to capitalize the expert's solutions in the field of safety. The definition of SAPs has to put in evidence relevant attributes from the safety point of view. The proposed definition contains three kinds of information:

- Informational features such as structure, behavior, good use condition...
- Safety properties of SAP and on its environment

- Formal behavior model.

SAP definition enables engineers to make easier a choice of architecture adapted to the problem to be solved. When several solutions are possible, the SAP definition can be used as an argument to motivate the best choice.

Safety properties that the SAP will fulfill under some conditions met on its environment are written both textually and with formal notations. The latter allows the validation both of the SAPs and the models built with these SAPs. The properties can formally be checked.

Regarding the extension of libraries of SAPs, in the previous project ESACS the models built using SAPs considered only failure modes of type "*total loss*". Extensions were made in ISAAC to take into account "*erroneous*" failure modes. Extensions were also done to integrate others SAPs such as for example command-monitoring mechanisms. As a general rule, engineers can build adapted patterns based on available generic ones, such as voters.

4.2.3.2.3 Process of SAP use

At each design stage, engineers start with sets of requirements that are not necessarily expressed with the same vocabulary as the one used in the SAP library. They have first to reformulate them accordingly. Then the search for a safety satisfactory architecture is made. Several possible cases are considered:

- The engineer can choose an existing system (e.g. hydraulics), which fulfills these requirements. Then, this part of design stops at this level.
- Another simple case is that there is a pattern that fulfills these requirements; this appropriate SAP is then selected.
- In most cases, a single pattern does not satisfy itself the complete requirements. On the other hand a combination of several patterns does it. This combination is then established.
- In some cases no combination of patterns can meet the requirements. In such a scenario, the engineer, guided by the principles stated previously, must then build the new SAP starting from existing patterns.

The use of a SAP leads to derive requirements on interfacing systems. The last step consists in verifying that these derived requirements are fulfilled as well.
This 3-step process (reformulation of the requirements, search for a safety satisfactory architecture, verification of the fulfillment of the derived requirements) is repeated until all the requirements are fulfilled.

4.2.3.2.4 Proposal of companion structure to support the process

During ISAAC we proposed a dedicated data structure called “companion structure” to memorize the requirements allocated to the sub-systems and also the assumption for their use. This information constitutes the important characteristics of a well-delimited part of a system. The companion structure permits indeed to easily modify or replace a part of the system in the model.

The companion structure is also used to capture dependencies between elements of the system design:

- It highlights set of requirements that fully characterize the role of pieces of architecture from the safety point of view. This enables to work in a modular way: design and assess as much as possible each module independently one from the others.

- It puts in evidence the cascade of requirements from system requirement to sub-systems or component requirements. It indicates how system requirements are shared out among components and their environments.
- It provides justification of these cascades. During the design, choices of components are made to ensure requirements that can be divided between claims about themselves and those on their environments. These choices are made successively and are dependent each other.

In conclusion, the proposed companion structure model is made of concepts of requirements, sub-systems, rules of choice with the various links that they have together.

4.2.3.3 Impact and applications

The **Safety Architecture Patterns** (SAP) technique is an effective means to give a support to safety engineers who need to quickly assess early drafts of system architectures. It also significantly favors the interrelationships and exchanges of data, mainly requirements, between the safety engineers and the designers since they can work on common model of the system preliminary designs.
The use of SAPs by manufacturers for the purpose of formal safety modeling activities can be foreseen in short /medium-term. It is all the more true because in the 3rd year of ISAAC project, we have produced guidance to quickly prototype early system architectures based on the concept of the “companion structure”. This is a first step of an industrial application of SAP technique.

At the end of the project, SAP exploitable results are the following ones:

- Provide efficient guidance to quickly prototype system architectures, to build and assess system architecture against design and safety requirements. Such a guidance permits:
    - To clarify the exchanges/interactions between designers and safety expert
    - To quickly and correctly design preliminary system architectures
    - To re-use parts of validated system architectures or technical solutions for the purpose of new modeling activities (capitalization of the know-how).
- Provide efficient guidance to elaborate valuable safety argumentation of the system design. Such guidance permits to elicit the breakdown of high-level safety requirements into low-level safety requirements according to an acceptable architecture (regarding to other requirements: technical feasibility, maturity, etc.). It also reinforces the safety requirements allocation process. Lastly, this is an effective way for harmonizing the requirements coming from various sources or rationales.

Therefore, the main benefits expected from the concept of SAP can be found in at least the three following key issues:

1) How *to build an architecture* answering almost automatically to the safety requirements.
2) How to *apportion high-level safety requirements* from aircraft level to a function architecture partitioned by systems.
3) How to favor the *cooperative work* between the designers and safety experts with an efficient *traceability* of the safety requirements.

During ISAAC, we have developed independent tools to support the method in complement to safety assessment existing industrial toolkit such as CECILIA/OCAS platform:

- An editor as well as a browser for managing the library of SAPS. This tool permits:
    - To capitalize parts of architectures together with the associated requirements in view of re-use them in the models

- To describe the patterns (characteristics, etc) to better understand their usage and added values and restriction on their use.

The innovative aspects of this tool are the following ones:

- Solution independent of a modeling platform (need only to be compliant with Altarica language)
- Separation of two kinds of activities: the know how collection from the activity of designing a specific model
- A requirement management module (companion structure). This is a dedicated tool to refine and trace safety requirements consistently as long as the model is expanded. It provides means to reinforce the safety requirements allocation process.

- An editor to build and manage the companion structure.

The innovative aspects of this tool are the following ones:

- Provide means to consolidated the relationship between the safety requirements.
- Solution independent of a modeling platform and of the modeling language

Although these two tools are only prototypes developed for research applications, this is a first step towards and industrial use of SAP's.

The interest of using the companion structure concept goes beyond SAP technical theme and could be applied to other field of activities.

4.2.4 TQA – Timing and Quantitative Analysis

4.2.4.1 Summary

The TQA theme intentions stem from the ESACS project where the possibility to perform analyses addressing temporal system behaviour was demonstrated. In ISAAC the main objective has been to explore new analysis techniques taking temporal aspects into account.

Several techniques are proposed to investigate various temporal properties related to safety and reliability of dynamic systems. It is not yet possible to automatically deduce all temporal aspects of safety analysis on a system-wide level. It is however possible to investigate temporal aspects on the level of individual cut sets and failure events. This makes it possible for safety engineers to better understand the temporal properties of failure scenarios.

Another investigated topic is quantitative analysis, which – when including temporal aspects - may lead to less pessimistic probability estimations. These analyses still need to be done mostly manually but the platforms include the necessary calculation techniques to support them.

4.2.4.2 Methodology

The TQA methodology cannot be described as a flow of working since it describes a number of techniques, which can be used for various analyses following the general ISAAC methodology, see the following figure. The TQA theme presentation therefore presents concepts and techniques – related to taking temporal aspects into account – separately but which can always be seen as supporting some specific step in the general methodology.

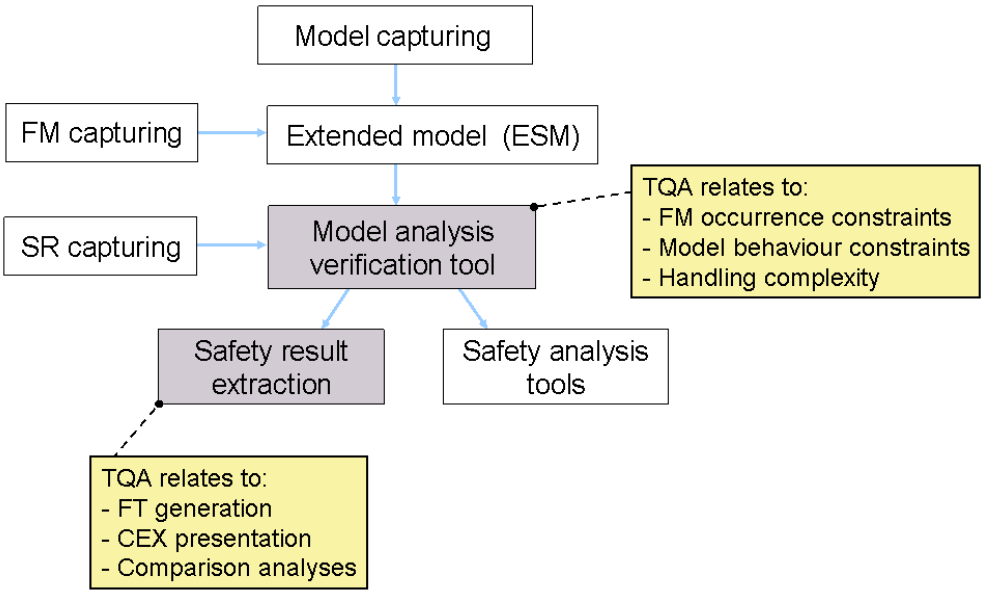

Figure 5 TQA relation to the general ISAAC analysis methodology

The TQA theme work can be dealt in two main topics:

- *Timing*, which have investigated and defined new techniques to utilize and evaluate the time-discrete event system model behaviour by taking temporal conditions into account.
- *Quantification*, which have explored techniques to quantify undesired system behaviour when including temporal aspects.

The first task was introducing the notion of *counter example* (CEX), i.e. a sequence of variable values - for the system being investigated - demonstrating how a requirement is violated. We coupled this to the traditional understanding of minimal cut sets, leading to the formal definition of minimal dynamic cut sets (MDCS).

We have defined and implemented techniques to handle various temporal properties of interest e.g. delays and durations for failure events and a possibility of constraining system dynamic behaviour, e.g. to differentiate the analyses between various phases.

There are various methods presented how to automatically generate hierarchical fault trees – when not taking temporal aspects into account. Beside this, there have also been studies for creating fault trees include temporal conditions. This temporal extension is not limited to a new notation; it also clarifies the semantics of a fault tree in the context of dynamic systems.

The amount of information related to the behaviour of large dynamic systems can be hard to manage both for humans and computers. To address this issue different ways are presented to simplify the evaluation and visualization of CEX.

Finally, ideas are outlined how temporal knowledge - e.g. importance of ordering of failures or cut set dependence of system phase - can be taken advantage of in quantification. This may result in more accurate reliability estimations compared to traditional, usually pessimistic, FTA assessments.

4.2.4.3 Impact and Applications

We have implemented techniques, which make it feasible to calculate minimal cut sets when taking temporal aspect into consideration. It is also possible to investigate temporal aspects on the level of individual cut sets and failure events. These methods make it possible for safety engineers to better understand the temporal properties of failure scenarios.

Ideas of generating fault trees including temporal aspects have been presented and can be further explored. In practice techniques are implemented, which automatically generate fault trees but without temporal considerations.

Another area of study has been to do quantitative analysis including temporal aspects that may lead to less pessimistic probability estimations. These analyses still need to be done mostly manually but the platforms include the necessary calculation techniques to support them. These quantification techniques need to be further investigated.

4.2.5 CCA – Common Cause Analysis

This chapter describes the CCA methodology. The first section provides a summary of the overall capability. The second section outlines the methodology and the third section describes the benefits that may be gained and the industrial domains that the methods can be applied.

4.2.5.1 Summary

The Common Cause Analysis (CCA) theme has delivered a method and tools to account for external events in a systems development seamlessly from a particular risk analysis (PRA) through to the safety analysis, and then back to the geometric environment where the results are presented and additional tools (allowing to perform customised measures) are used to determine what options there are for actions to be taken. These additional tools can also be used to support zonal safety analysis (ZSA) in order to examine each physical zone and to ensure that equipment installation and potential physical interference with adjacent systems do not violate the independence requirements of the systems.

4.2.5.2 Methodology

This section provides an overview of the final method established by the ISAAC Partners for facilitating the safety analysis of systems that are subjected to *Particular Risks* or *zonal segregation constrains* and to facilitate in understanding what actions should be taken to satisfy the requirements.

Particular Risk Analysis

The Society of Automotive Engineering's Aerospace Recommended Practice "SAE-ARP 4761", describes *PRA* as studies of *"particular risk events which are outside the system(s) concerned but which may violate event independence claims"* because they *"may influence several zones at the same time"*.

Some examples of particular risk events commonly considered in the field of Aeronautics are: Uncontained Engine Rotor Failure (UERF), Auxiliary Power Unit (APU) burst, tyre burst.…

The presented method and implementation is in line with that reported in SAE-ARP 4761 and implies the use of both industrial commercial tools (like Computer Aided Design (CAD) and functional modelling tools) as well as new software applications developed ad hoc by the ISAAC Consortium (such as new facilities that have been integrated in the existing ISAAC platform as well as the development of mapping facilities for interfacing between geometric and functional environments)

The process specifically carried out to address the account of particular risk analysis in the safety assessment, presented in Figure 6, includes a forward and a return path:

- **forward path** (from the geometrical to the functional world): The forward path starts from
    - a risk model that represents the effect that a particular risk under investigation has on all 'affected aircraft systems' installations;

- then the information relevant to the affected items that have been impacted by common individual trajectories is transferred to the functional model,
- where a qualitative safety analysis is carried out (Fault Tree Cut-set analysis).
- At the end of the process the safety criticality of risk fragment trajectories are identified.
- This information can then be used to perform a quantitative assessment of the probability of a particular risk leading to a final effect.

- **return path** (from the functional to the geometrical world): Having completed a qualitative analysis,
  - the notion of criticality can be assigned to the components involved in a cut-set
  - through an interpretive step an indication of worst criticality can be assigned to trajectories as well.
  - The above two assignments are transferred back to the geometrical environment
  - The two assignments are indicated through the presentation of
    - cut-set groups
    - or colour coded trajectories where the colour is associated to the criticality that a trajectory occurrence implies.
  - A failure probability is calculated for the occurrence of the particular risk fragment and is compared to the requirements.
  - This will help the safety engineer to decide the appropriate solution that addresses the identified problem areas.

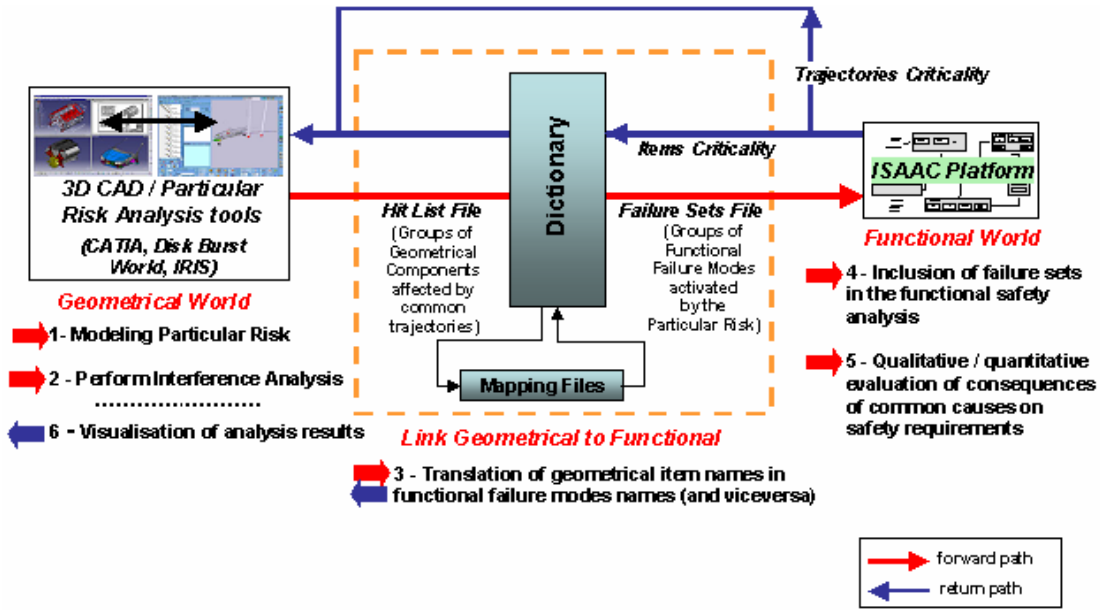

Figure 6 PRA Method

The method, depicted in Figure 6 above includes the following main steps:

- *Modelling of particular risk phenomenon in the geometrical layout.*
  This step consists of the construction of “risk failure models” for various types of particular risks, in other words, the construction of geometric models that represent the volume that the fragments fill and the relevant ejection cone (trajectory). The risk failure models have

been built in agreement with standards (e.g. the CS-25 *Certification Specifications on Large Aeroplanes,* AMC 20-128A for risk failure models relevant to uncontained engine rotor failure-UERF) and are parameterised so as to enable reuse of the model for different engine stages and for different aircraft.

- *Extension of the geometric model with the particular risk failure model and execution of an interference analysis between the fragment trajectories and aircraft systems.*
  The selected risk failure model is integrated in the geometrical model of an aircraft system/s in order to obtain an "extended geometrical model" (system installation + risk failure model). The result from the multiple interference analyses between each of the various fragments trajectories (determined by the possible angular positions of the fragments that together cover the full risk volume that describes the agreed affected volume defined by the standards mentioned before) and the aircraft systems (that we will refer to as *hit list*) is recorded in a common format.

- *Translation of results from the interference analysis to a format that is useable in the functional environment (in other words, into failure modes of the relevant functional formal models) through a dictionary.*
  This step consists of the elaboration of a list of "failure sets", each of them including the list of failure modes, all triggered by the fragment ejected along a given trajectory or a group of trajectories that share an identical set of impacted components.

- *Inclusion of failure sets in the functional safety analysis*.
  The failure sets are imported in the ISAAC platform and are considered as additional failure modes to be accounted for in the safety analysis tasks, which is similar to a traditional fault tree analysis, to determine the combination of failure modes (so in this case also to determine which failure sets correspond to each fragment trajectories) that lead to the violation of a safety requirement.

- *Qualitative Evaluation of the impact of a particular risk on the violation of a safety requirement.*
  This step consists of assigning levels of criticality based on the effect that any single fragment has on all concerned aircraft level functions as a consequence of the particular risk event invalidating claims of independence.

- *Quantitative Evaluation of the impact of a particular risk on the violation of a safety requirement.*
  This step consists in using the fragment geometry, airworthiness regulations guidelines for calculating fragment risk probabilities and through the previous means of determining the safety implication of each fragment to determine whether regulations objectives have been met.

- *Visualisation of analysis results*
  Once having determined the safety criticality of trajectories by means of the qualitative analysis described above the information can be transferred back to the geometrical environment in order to be visualized. An established criticality definition for the aircraft items is also transferred back and visualized, depending on the user need. This will help the safety engineer to decide the appropriate countermeasures to the identified problem areas.

- *Customized measures for PRA*
  Having imported the cutsets into the geometric environment then through the use of customised forms it is possible to carry out measurements to understand the degree of penetration of fragments on equipment so that it is possible to understand by how much a piece of equipment should be moved and in what direction in order to satisfy the claim of independence that has been violated by a particular risk fragment trajectory.

Zonal Safety Analysis

The Society of Automotive Engineering's Aerospace Recommended Practice "SAE-ARP 4761", describes *ZSA* as studies that "*should examine each physical zone of the A/C to ensure that equipment installation and potential physical interference with adjacent systems do not violate the independence requirements of the systems"*

For the purpose of performing ZSA *customized measures* are created to check the segregation distance between the components or the components between two systems. E.g. hydraulic and electrical system. These customised measures are created automatically using the tools developed within the geometrical modelling environment, e.g. Catia v5. Comprehensive reports are created that contain all the details from the ZSA measures. These measures are also saved as part of the design model.

4.2.5.3 Impact and Applications

To understand the Impact and Applications that the CCA capability offers and brings with it we have to consider the benefits, advantages, disadvantages, the maturity of the tools and the number of tools needed to maintain such a capability.

The CCA methodology has the following potential applications and the impact that is hoped but also brings with it some difficulties:

Advantages/ Benefits

- Validation of the systems installation against safety requirements can be started in the early phases of aircraft development and maintained or repeated through out the development lifecycle.
- It is possible to complete a Particular Risk Analysis iteration in significantly reduced time scales when compared to the purely manual approach.
- It should be possible to have a greater level of optimisation of the final product from the point of view of architecture specification and installation.
- It should enable a more optimal design and installation to be achieved with respect to installation or redesign of architecture.

Disadvantages

- The amount of data that is generated can easily become overwhelming in a very short period of time.
- The demand on computing resources is approaching the current limits of the available machines.

- The capability is delivered through a chain of tools that are developed by 4 different tool developers. Changes in the geometry or functional tools require reworking the tool that enables the exchange of the information. Considering the frequency of tool development the frequency of modification of all the tools can become very high

Time to market and considerations that need to be addressed to achieve this.

- The tools that customise measurements for specific interests can be deployed with little further development
- In order to be able to exploit the full capability of ISAAC Particular Risk Analysis it has to be considered that this capability depends on a set of tools that need to be supported together. In order to apply these tools on an aircraft program the tools need to come with a support strategy and tool vendor commitment so that the tools are available and maintained, and their maintenance is coordinated for the duration of an aircraft program. Anyway, each part of the ISAAC Particular Risk Analysis can be deployed since now.

Future development

- Automate the resolution of derived installation requirement conflicts through the use of formal methods tools customised to search for viable layout of systems in the presence of particular risks.

4.2.6 HEA - Human Error Analysis

This chapter describes the HEA-methodology. Section 6.1 provides a summary of the work that was performed during the ISAAC project and the end results along the HEA-requirements. It presents in which way the requirements were covered. Section 6.2 summarises the individual steps of the methodology and described how the results of the HEA-theme extend the state of the art. Section 6.3. describes the advantages and difficulties of the methodology and addresses the impact on its application in the industrial or research sector.

4.2.6.1 Summary

The main objective of the HEA theme in ISAAC was to adapt the ESACS methodology to the requirements of an industrial human error analysis. The general target of human error analysis in aeronautics is to identify potential pilot errors and the safety impact on flight. The most general requirement for the HEA theme was "*To provide a tool-supported methodology for performing the HE analysis directly on extended formal system models taking into account cognitive limits of the pilot*". Instrumental to this requirement was a cognitive architecture developed by OFFIS in previous studies and provided to the ISAAC project. This model was used as the basis to fulfill the requirement "*To set-up an environment to simulate the pilot interaction between pilot model and design model in different operational scenarios*".

The OFFIS *cognitive architecture* focuses on a cognitive routine learning process that was modeled based on a simulator study with four pilots at the Lufthansa Flight Training Center conducted by Lüdtke and Möbus in 2004 (Lüdtke and Möbus 2004). As a result they found that a subset of pilot errors may be explained by "learned carelessness". This psychological theory (Frey and Schulz-Hardt 1997) states that humans have a tendency to neglect safety precautions if this has immediate advantages, e.g. it saves time. Careless behavior emerges if safety precautions have been followed several times but would not have been necessary, because no hazards occurred. Then, people deliberately omit safety precautions because they are considered a waste of time. The absence of hazardous consequences acts as a negative reinforcer of careless behaviour. Learned carelessness is a process which is characteristic for human nature because we have to simplify in order to be capable to perform efficiently in a complex environment. We implicitly degrade our mental model to optimise it for routine situations. Unfortunately this may be disastrous in slightly deviating scenarios. Thus it is crucial to consider this process in system design. Lüdtke and Möbus modeled learned carelessness based on the mechanism of rule composition and tested with a formal design model of a Piper Cheyenne autopilot inside a simulation platform. An in-depth analysis of two different procedures (altitude change and auto flight engagement) was performed. These trials demonstrated that the implemented functionality of the simulation platform and the pilot model are sufficient enough to simulate the interaction between pilot and system model in varying flight scenarios. The comparison of the model behaviour and real pilot behaviour shows that our model commits errors that comply with errors observed in the empirical study: Seven routine errors of subject A can be successfully reconstructed and two routine errors of subject B and D can be correctly predicted. Furthermore, Lüdtke, Möbus and Thole (2002) have shown that the real pilot behaviour exhibits empirical phenomena (speed-up and interleaving) that indicate rule composition. Comparable effects can be evidenced in the model behaviour. The cognitive model is composed of two essential parts: a cognitive architecture and a procedure model. The cognitive architecture was implemented in a modular way, which means that the individual components have common interfaces for data exchange and scheduling. This allows to exchange and add modules in order to extend the

cognitive functionality of the model, e.g. in order to add a workload mechanism or multiple task execution. The cognitive model and the simulation platform developed by Lüdtke and Möbus have been used as a starting point for the ISAAC activities. Starting from their initial results the cognitive model and simulation platform were investigated during ISAAC with further more complex procedures, like arrival and takeoff. In this way the methodology was improved according to the requirements of the industrial partners working towards a mature cognitive model applicable in the industrial context.

Dedicated tools have been developed to support the application of the cognitive model as an extension of the existing ESACS platform. An procedure editor has been built that allows to edit procedures in a GOMS (Goals, Operators, Methods, Selection Rules)-like format (Card, Moran and Newell 1983) that proofed to be intuitively for the industrial partners who tested the HEA-tool. The editor allows to present the goal-subgoal hierarchy of procedure models in a graphically way. The procedure modeled in the procedure editor is the normative procedure. That means, applying these rules must not lead to a violation of safety requirements. Furthermore, a structure for scenario models has been devised. These models allow to specify patterns of flight situations in which the procedure may be applied. Based on these models a batch mode for the simulation platform has been implemented in which scenario instances are automatically derived from the scenario models (random variables are instantiated) and are simulated without necessary intervention by the users. The batch mode generates scenarios based on provided frequencies and scenario patterns and performs simulation runs until the learning process has reached a stable state. By simulating the interaction between the procedure and system model inside the simulation platform the following question shall be answered: How will pilots implicitly adapt/simplify the mental representation of the normative procedure taking into account learned carelessness? In case of a negative answer possible solutions to the problem shall be analyzed by optimizing the “safety nets”, (e.g., by improving the user interfaces, and the operational procedures in the cockpit). In this way the simulation fulfills the requirement “To devise a method for identification of cognitive non-adequate design structures”.

Two case studies provided by the industrial partners served to test the overall HEA methodology and the requirement “To investigate if the predicted HEs (learned carelessness) are realistic or if integration to the prototype model is needed.” The results of the application of case studies have shown that the behavior of the pilot model is plausible on some aspects and implausible on others, notably those where concurrent behavior (multi-tasking, cf. below) is requested (like monitoring the speed of the A/C while steering its trajectory on the ground during the initial phase of the takeoff) or when the dissociation of the crew between a PF (pilot flying) and PNF (pilot not flying) is involved. With regard to the predicted human errors the industrial partners in ISAAC acknowledged that a number of erroneous actions due to learned rules have been observed and then discussed with pilots who have good knowledge of the scenarios that were simulated. Results gathered so far appear to be plausible in terms of observable behavior. However, subject matter experts highlighted that learned carelessness might be a cause for such errors only under specific environmental circumstances (e.g. high workload, rush operations). This allows deriving an indication for improving the environment model as well as the cognitive architecture (taking into account workload and workload inducing factors) in future development. Testers stated that though the pilot model focuses on learned carelessness only, it is possible to derive indications that may be useful to support the identification of system improvements. But it was noted that due to the scope of the current method it is necessary to thoroughly discuss and evaluate the results with experts.

Apart from “learned carelessness” there are many other capabilities and limitations of the human cognitive system, that have to be considered during design in order to optimize the “safety nets”. In ISAAC we focused on one of these features to investigate the general applicability of cognitive modeling for safety analysis. This leads to stringent constraints of what can be taken into account during modeling, analysis and how the generated predictions have to be interpreted. Potential additional error producing cognitive processes and potential implementations of these have been discussed and conceptually described. In future activities the results of ISAAC could be used to naturally extend the cognitive mechanism to take into account also other aspects beyond “learned carelessness”, like human multi-tasking and resulting workload. Apart from the automatic injection of failures into the procedure model by the learning process, a manual error injection mechanism has been implemented .

The analysis by human simulation as described up to this point allows to answer the questions if it is likely that pilots implicitly adapt/simplify the mental representation of the normative procedure taking into account learned carelessness. A further requirement was to a support a second, subsequent analysis in order to answer the question, if this adaptation may lead to pilot errors and in consequence to violations of safety requirements? There are two ways to analyse this question. First, the simplified procedure may be faced, in addition to the scenarios that produced it, also with new scenarios, where new problems are posed to the pilot and see how the model reacts, possibly producing errors. Second, in order to exhaustively analyze the impact of human errors on safety requirements we apply model the checking techniques developed in the EPC-theme to automatically produce a fault tree. This tree presents causal relationships between violated safety requirements and simplified rules. The safety requirements are specified in the notation that was introduced in the ESACS project.

4.2.6.2 Methodology

In this subsection an overview of the final methodology of the ISAAC Human Error Analysis is given and the relation to the state of art is described.

4.2.6.2.1 Overview of the HEA-Methodology

The main elements of the methodology, including inputs and outputs are showed in the following Figure 7.

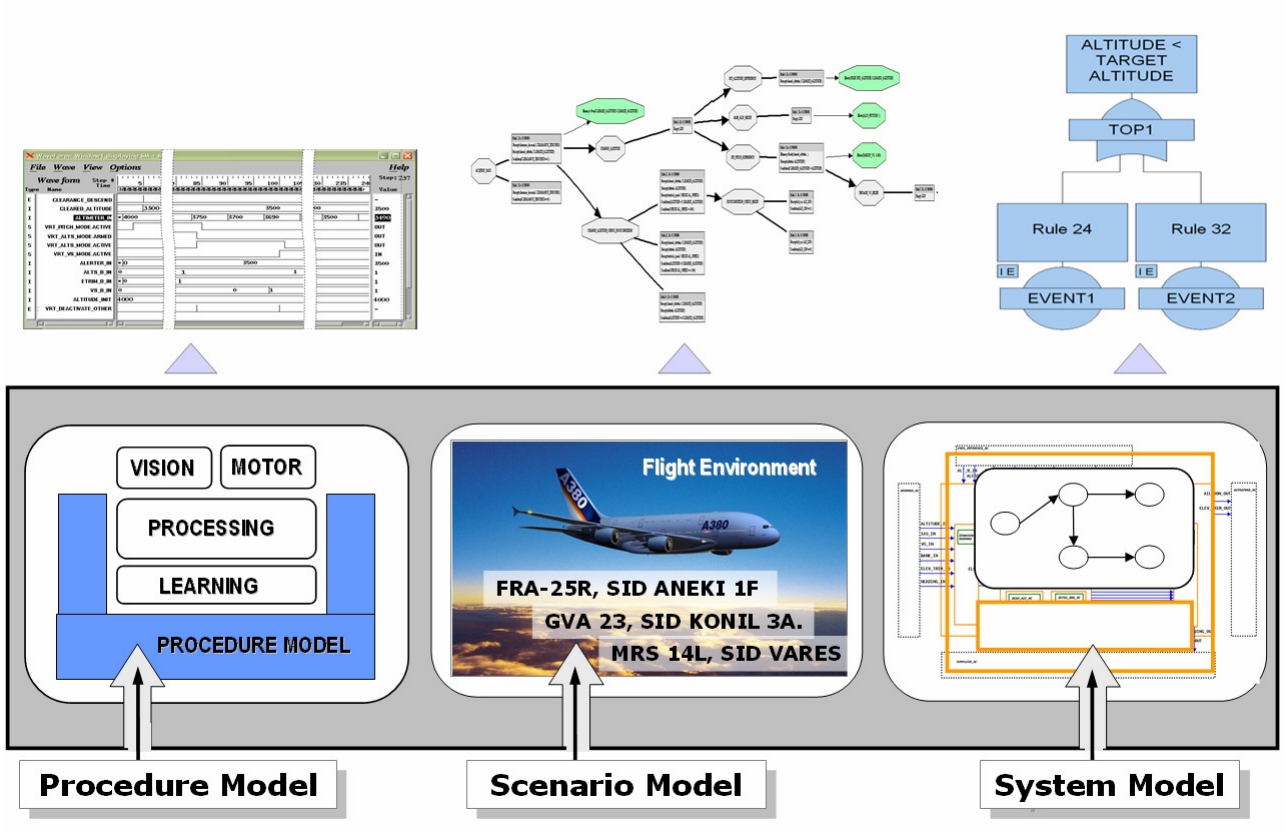

Figure 7 Human Errors Analysis Framework

The main steps for using the methodology are described in the following.

1.) Prepare a Statemate design model of the system under investigation: The main target of the analysis is a formal system model. The modelling challenge here is to define a level of abstraction that describes the system from the pilot's point of view. In ISAAC we focused on mode based systems and developed a highly reusable modelling structure encompassing the mode logic (set of modes and mode transitions) and the corresponding control laws as main parts.

2.) Prepare a model of the procedure under investigation using the ProcedureEditor: The procedure model formally describes how to operate a system in order to perform a certain flight task (like takeoff or airport arrival) - a mental representation of a flight procedure. It consists of "if-then" rules, which proved to be an easy to understand format for the users and which is at the same time consistent with the internal format required by the cognitive architecture (GSM production rules). To support construction of procedure models, a Procedure Editor was developed in the project.

3.) Prepare a set of scenario templates: A procedure can be applied in many operational situations. These different application situations are what we call "scenarios". One requirement for the ISAAC-HEA method is to perform a human error analysis sufficiently evocative to discover human machine behaviours that are difficult to predict without human modeling. Therefore, the choice of the scenarios is important and has an impact on the HEA result as well as on the representativeness of the models. The intention in defining the scenarios is to be realistic and to consider varying categories of operational situations. In general, a scenario is defined by an initial state and a list of events (e.g. clearances) and describes day-to-day normal revenue flights. Instead of defining concrete values for the variables representing events and initial state data it is possible to define probability distributions for variables (variable distributions). Based on these distributions the variables are automatically instantiated at the beginning of each simulation run. Thus we call the scenario models scenario templates from which (an arbitrary number of) concrete scenario instances are derived. The methodology foresees to model these scenario templates in Statemate. In this way the user can use the same formalism for the system model and the scenario model. In ISAAC we developed a modeling structure for scenario templates. In this structure it is possible to define a set of templates. For the set of scenario templates the user has to define a probability distribution (template distribution) relative to a certain population. For example, the population may be "takeoffs in Europe with a A340 – 300". The intention of assigning a template distributions is to assess how representative the scenarios are for this population of takeoffs. During the simulation of pilot-system interaction scenarios are chosen randomly based on the frequency. Thus the frequency has a major impact on the emergence of careless/simplified rules due to the learning mechanism.

4.) Start the human simulation: All three models (system, procedure, scenario) are coupled inside a simulation platform in order to simulate the pilot-system interaction. The simulation platform adds an environment model as a fourth component. This model encompasses the flight dynamics of the aircraft and is necessary to close the loop between pilot and system. The platform synchronizes the four models based on a common time concept. The user can start the platform either in manual or in batch mode:

4.a) Start the simulation in manual mode: In the manual mode the system model is animated using the Statemate Simulation Environment. Only one simulation run is performed . After the simulation run the user can inspect a simulation trace presenting the dynamics of a selectable set of variables. The trace contains cognitive variables (percept actions, motor actions and procedure rules performed during the run) as well as system variables (e.g. altitude, speed, flight modes).

4.b) Start the simulation in batch mode: To support the simulation of an arbitrary number of scenario instances the simulation platform offers a batch mode. In batch mode the following steps are repeated until the cognitive routine learning process has generated a *stable* version of the procedure model:

4.1. A scenario template is randomly chosen taking into account the probabilistic template distribution.

4.2. From that template a scenario instance is randomly derived taking into account the probabilistic variable distribution.
4.3. The simulation run is started.
4.4. During the simulation the pilot model interprets the procedure model. Each time a task is finished (no more subgoals to process) the learning mechanism updates the rule strength parameters and adds new simplified rules.
4.5. The simulation run is stopped as soon as one of the termination conditions of the scenario instance specified by the user is fulfilled.
4.6. If the maximum number of simulation runs (specified by the user) has not been reached up to now, the next simulation run is started (start again with 4.1), else the batch simulation is terminated.

5.) The user can inspect the simulation results:

5.1. The Procedure Editor offers a feature to automatically generate tree views of the procedure model. The tree views presents the procedure in a graphical format showing the hierarchy of goals and subgoals. The view shows normative as well as learned rules. Learned rules can be further inspected using a History View, that presents from which source rules the learned rule has been derived. These interdependencies between source rules and learned rules is called the “learning history”.
5.2. The user can obtain statistical information on the simulated runs. These are: number of runs simulated, number of runs that terminate with a scenario failure; number of tasks performed with success, number of tasks ended with failure; for each goal the rules both normative and learned with the relevant strength factor value. For each run: the indication of the chosen scenario, the success/failure indication of each task and the indication if/which new rules where applied
5.3. The user can investigate the traces of the simulated runs by means of a trace viewer. This in order to understand better the consequences of the application of the new rules on the predefined scenarios.

6.) Perform a Fault Tree Analysis for the learned state of the procedure: In order to analyse if the application of the simplified rules may violate a safety requirement, the HEA was integrated with the FTA feature of the STSA tool. This allows to add a procedure model to FTAs. The resulting fault tree shows procedure rules (as basic events) leading to the top level event. The top level event represents a violation of a safety requirement. Safety Requirements are formalised in the same way as in the FTA methodology.

4.2.6.2.2 Relation to the State of the Art

The focus of our analysis is on mode errors, where an action is performed that is correct in some modes but not in the present one. Mode errors lead to "automation surprises" (Sarter, N. B., Woods, D. D. & Billings, C. 1997), where an operator no longer understands what the system is doing. During the design process the need for modes and mode transition conditions has to be balanced against the probability of mode errors. Mode related problems have been identified by numerous researches, e.g. by Sherry et al. (Sherry, L., Feary, M., Polson, P., Mumaw, R., Palmer, E. 2001) in the Vertical Navigation function of Flight Management Systems. Human error has been a major concern for a long time (with the main interest arising in the 70s) and it is taken into account at many stages of the aircraft, systems, and cockpit design. The methods used however mostly rely on human expertise (e.g., the test pilots, in-house human factors specialists),

operational feedback from similar aircraft (e.g. experience feedback systems, reporting systems), and (costly) experiments conducted at the simulator when a prototype of the aircraft (or of advanced systems) is available. The interest of the ISAAC-HEA approach is that it attempts to be predictive, and applies readily on the specifications of the system. This would allow to introduce an approach to system design based on formal prototyping, where different versions of the systems are described in different specifications, and tested against human error already early in the design process. Moreover the approach should help in better choosing the “worst cases” scenarios from a human factors point of view that are the most relevant to consider for design and test purposes.

One of the first executable pilot models was the AIDE-model of Amalberti and Deblon (Amalberti, R. & Deblon, F. 1992), which simulates operating skills of military pilots. Their intention was to apply the model in assistance systems. In the domain of design support, the approach of Crow, Javaux and Rushby (Javaux, D. 2002, Crow, J., Javaux, D. & Rushby, J. 2000) is similar to the ISAAC-approach: Javaux uses a cognitive model to predict mental adaptations of mode transition models. A precondition for the application of Javaux’s method is the availability of expert ratings of the frequencies of mode transitions. These ratings are input to a Hebbian learning mechanism. Crow, Javaux and Rushby have shown how such mode transition models can be integrated with formal design models to predict possible human errors by verification. The ISAAC-HEA human simulation approach provides the opportunity to perform analyses also in cases when no expert ratings are available, which is often the case for new systems. Other human simulation based approaches like MIDAS (Corker, K. M. 2000)and APEX (Freed, M.A., Remington, R.W. 2000) differ from our approach because they do not model human learning mechanisms. Nevertheless they rely on similar cognitive processes like “learned carelessness”. But they require the end-product of human learning, simplified procedure models, as input, which has to be provided by the system designers or human factors experts, whereas we apply a psychologically plausible learning mechanism.

As explained, “learned carelessness” was implemented in form of a rule composition mechanism. This implementation was motivated by the compilation mechanism in ACT-R (Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y . 2004). Rule compilation in ACT-R melts two rules into a single new rule that has the effect of both old rules. Certain intermediate steps (e.g. subgoals) are eliminated. But in ACT-R this process does not lead to erroneous rules, because elimination of percept actions is explicitly prevented (see Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y . 2004), page 1045). Since our goal is to predict potential pilot errors caused by omitted percept actions, we modified the rule compilation mechanism and elaborated the generation of incorrect rules. More details on this elaboration can be found in (Lüdtke and Möbus 2004).

Polson and Javaux (Polson, P.G. & Javaux, D. 2001) developed a model that explains “why pilots do not always look at the FMA” based on a learning process with different subsequent learning phases. In an associative learning phase pilots associate various cues with mode changes. In a subsequent selection phase the most useful method (reliable cues that are easy to perceive) to verify the current mode state is repeatedly applied. Finally, methods neglected during the selection stage are eliminated. The latter phase is based on the frequential simplification process introduced by Javaux (Javaux, 1998) to explain mode errors in context of the B737 and A320 autopilot. The ISAAC pilot model considers the selection and simplification stage but neglects the associative learning stage. Thus the ISAAC model currently predicts that pilots will either verify the mode state (by looking at the FMA) or will totally neglect this verification - black or white perspective. The provision of an associative learning phase would allow to analyse if it is likely that pilots replace looking at the FMA by other cues (like a positive load factor as an indication for capture mode activation) and if this replacement is really reliable in all scenarios.

Polson and Javaux emphasise that not looking at the FMA is not a violation but the result of an implicit learning process taking into account reliability, cognitive cost and frequency of methods.

The ISAAC model follows the same assumption. Rules are simplified, if relying on memory values proves to be successful at a lower cost. These simplifications are very likely to happen especially during routine performance of tasks by skilled people. They can not be reduced by training but only by design. It is a process which is characteristic for human nature because we have to simplify in order to be capable to perform efficiently in a complex environment. We implicitly degrade our mental model to optimise it for routine situations. Unfortunately this may be disastrous in slightly deviating scenarios.

4.2.6.3 Impact and Applications

This section concludes the HEA methodology by describing added values advantages and difficulties.
The HEA-methodology has an potential impact in the following fields:

- an earlier collaboration between design departments and human machine department due to the fact that human actions and failure modes are included in the analysis.
- It will be necessary to think through what possible failure modes that exists for the pilot actions.
- It will be possible in an earlier phase to find out which pilot actions can lead to safety requirements not being fulfilled.
- The predicted pilot errors can be applied to focus the subsequent human error investigations in flight simulation with test pilots.

During ISAAC it has been demonstrated how a cognitive modelling approach can be applied to support the industrial human error analysis. The methodology is not yet ready for industrial exploitation. The following lists give an overview of the added value and difficulties of the current development state and highlights the degree of maturity.

Added value:

- The method provides a modular structure that allows to model mode based systems from the pilots point of view.
- The method provides a formalism and an editor to model and present flight procedures. The users of the methodology inside the ISAAC project got an instant grip of the GOMS-like formalism for procedure models
- The method provides a structure to model patterns of flight scenarios from which scenario instance are generated during runtime.
- The method provides a cognitive architecture and a simulation platform to animate/execute procedure models
- The method allows to manually inject pilot errors in procedure models and to simulate the resulting impact on the pilot-system interaction.
- The method provides several mechanism to evaluate the behaviour of the pilot model:
  - Simulation monitor showing the evolvement of the variables during simulation,
  - Trace viewer showing the evolvement of the variables after the simulation,
  - Procedure Viewer, History and Evolvement Viewer to present learned rules
  - Statemate simulation mechanism to inspect the evolving state of the system model charts during manual simulation mode,
  - Flight simulator windows showing the behaviour of the aircraft during simulation.

- The method allows to generate Human Error fault tree presenting procedure rules that may lead to violations of safety requirements.

Difficulties:

- The method currently only supports discrete system models, and thus only discrete controllers could be applied. But, for the purposes of the ISAAC project discrete controllers proved to be realistic enough.
- The procedure formalism currently supports no multi-tasking aspects (like task priorities), no crew cooperation, no sophisticated perception, and no continuous tasks (like steering). These limitations are due to the scope of the cognitive architecture, which was intended from the beginning due to time and budget constraints during the project.
- At present there are still some limitations of the scenario formalism, again due to the scope of the cognitive architecture: no interruptive events, no crew diversity, and no bad weather.
- A dynamic rendition of a digital pilot model (like JACK in MIDAS) is not available
- The result of the evaluation was that the behavior of the pilot model is plausible on some aspects and implausible on others, notably those where concurrent behavior (multi-tasking, cf. below) is requested (like monitoring the speed of the A/C while steering its trajectory on the ground during the initial phase of the takeoff) or when the dissociation of the crew between a PF (pilot flying) and PNF (pilot not flying) is involved. The major problem was due to the order of the accomplishment of each action : it was necessary to rewrite more accurately the strict and partial orders of the actions in the procedures model.

Considering the advantages and difficulties, certain parts of the methodology are ready for industrial exploitation in the short run, others need further research efforts in future activities.
Ready for industrial exploitation is the Procedure Editor. It can be used to document, present and formalize flight procedures.

Results that have the potential for further development in the research sector:

- Guidance for the analysis of human errors in man-machine interaction
- Simulation Platform allowing closed-loop simulation of pilot-system (e.g. autopilot design) interaction in realistic scenarios
- Cognitive model (reusable modular cognitive architecture)

4.2.7 MRA – Mission Reliability Analysis

4.2.7.1 Summary

Mission analysis target is to determine the impact of degraded situations on the system operational modes and over pre-defined missions that define the scenarios in which the system being developed will be used.
ISAAC approach is to find extension of the techniques set-up in a previous project (ESACS, FP5) to open the possibility of automating such analyses, e.g. helping in compiling a Minimum Equipment List in the first aircraft/system development phase, then supporting specific operational Mission Failure and Mission Reliability Analysis.
The advantages of this activity is the possibility of using a uniform methodology and environment to conduct tasks and analyses that are currently carried out using different tools, manual analyses, etc. with the potential benefits of a better integration of the various activities and analyses that are part of the engineering process of complex systems.

4.2.7.2 Methodology

The work was performed according to the following steps:

- identification of requirements
- development of methodology and tool
- application to case studies
- improvement of methodology and tool according to the feedback from the application

The basic requirement *"to perform mission analysis on a **model** written in the same notation of design and, where available, on the design model"* determined the approach followed in developing the methodology that is described in the following Figure 8 (Model based approach).

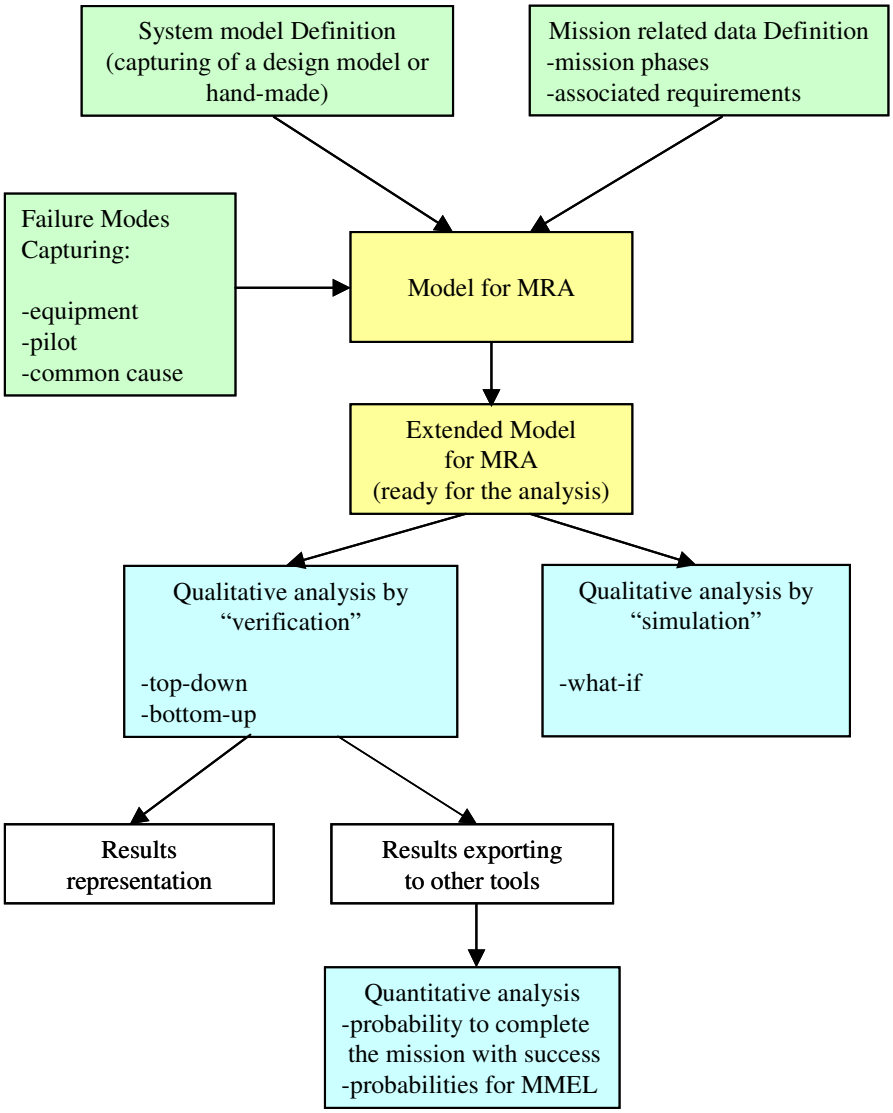

Figure 8 MRA methodology

In practice the idea is to start from a *"formal model"* of the system under investigation, then to add to it information relevant to the operational context in which the system is used (the *"mission"*) in order to be able to perform a mission analysis.

According to this idea, the first issue dealt was the *"definition of guidelines for the modelisation*" in the sense that a *"model for mission analysis"* should contain, in addition to the system behavior, also the representation of the mission in terms of *"phases"* and of associated *"requirements".*
Moreover, considering the occurrence of failures, the model should also be able to take into account for contingency missions that run adjacent to or branch off from the planned mission.

Relevant to the "failures", these can be of several nature:

- equipment (relevant to system components)
- pilot (relevant to pilot errors in doing some actions, e.g. pilot do not activate a pushbutton, pilot activate the pushbutton in the wrong position)
- common cause (e.g. an uncontained engine rotor burst that affects several components simultaneously)

The investigation performed in this sense has identified ways for failure representation that are shared with other themes of the project.

The *"equipment failures"* can be either modeled by the user or can be selected from a "library" (e.g. output variable stuck at a certain value).

An investigation was performed on how to include pilot actions/malfunctions as extension to the model. The original idea was to integrate the "pilot model", developed in Human Errors Analysis theme, in the context of MRA task.
Nevertheless, considering the time/effort limits of the project, it was not possible to perform this kind of experiment in the project time frame. This could be a subject for a future development after ISAAC.
At present an alternative way can be adopted, consisting in modeling the pilot actions and relevant modes for "incorrect behavior" directly into the model for MRA.

In case a *"common cause"* (deriving from a particular risk phenomenon like engine rotor disk burst or wheel tyre burst) is considered, the *"failure set"* notion is used, where a failure set is a set of failure modes that are activated simultaneously (or in a cascading way) due to some common cause.

Having defined the system model, the scenario model together with the associated requirements, and having captured the failure modes, it is possible to perform the analysis.

The analysis typology can be grouped into two types:

- qualitative
- quantitative

In case of the qualitative analysis, the targets are the following:

- starting from a top level event (e.g. the mission unsuccess or some mission requirement) to find the scenario traces leading to it (top-down analysis)

- starting from a specific combination of FMs to find if it leads to the mission unsuccess, affects the possibility to complete some mission phase or leads to the violation of some mission requirements (bottom-up analysis)

In case of the quantitative analysis, the targets are:

- to compute the mission reliability probability (MRP), i.e. the probability to complete the mission with success.
- to evaluate the mission probability value in all cases when each time a specific equipment is considered as "failed" (i.e. probability = 1 to be failed). The results of the calculation allow the user to verify and adjust the Master Minimum Equipment List (MMEL).

For the qualitative analysis, verification and simulation techniques can be used, where verification allows to automatically find in exhaustive manner the combinations of failures that impact on the mission continuation and “what-if” simulations allow to find the effects of failures depending on the operational phase.

The ISAAC tool also allows to define *“observables”* and/or *“intermediate events*”. These consist in variables that are monitored during the proof and that are then used in the elaboration of the results in order to create a better user-view for the output fault tree (e.g. by creating a sub-level under the top level event-the mission unsuccess that present the unsuccess cut.-sets grouped by mission phase). An example is reported in the following Figure 9.

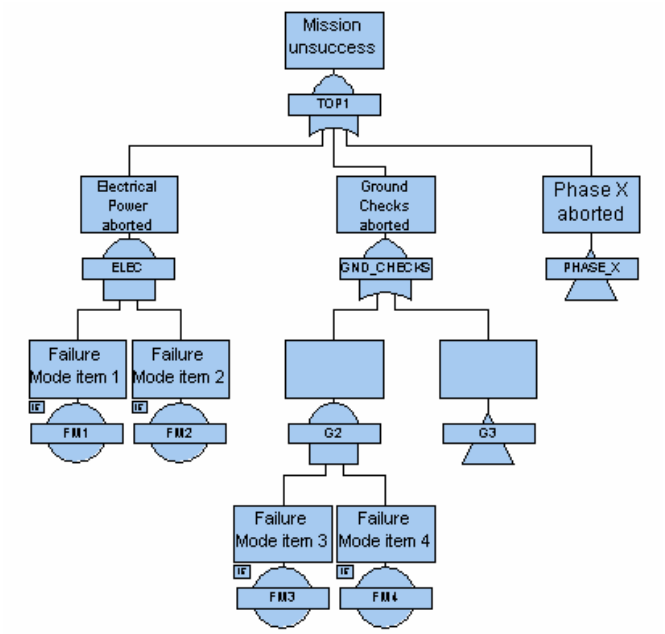

Figure 9 MRA Fault Tree

Relevant to the quantitative analysis, at present it is not integrated in the ISAAC tool, so it is necessary to export the qualitative results towards other tools in order to perform the calculation.

4.2.7.3 Impact and Applications

In order to describe the potential of the new method for application in industrial context and for further development in other researches, at first the pros and cons identified during the project are reported in the following.
The advantages of the new methods is the possibility of using a uniform methodology and environment to conduct tasks and analyses that are currently carried out using different tools, manual analyses, etc. with the potential benefits of a better integration of the various activities and analyses that are part of the engineering process of complex systems.
More precisely:

- Formal representation of operational scenarios and requirements results in an improvement of exchanges between safety-reliability engineers, designers and customer.
- Modular architecture allows the re-usability of system and scenario models.
- Model based design associated to specific modeling guidelines and simulation/verification techniques allow to address in more automated and comprehensive way issues related to the intended system operational use since the early stages of system's development.
- With the support of an adequate modeling it is possible to take into account for contingency missions that are deviations or reversionary missions from the preferred and original one.
- The possibility to simulate a system model in an operational context, allowing to perform a series of "what-if" simulations and finding the various effects on the mission of failure modes occurrence depending on the operational phase.
- Verification techniques allow to find in exhaustive manner the combinations of failures that impact on the mission continuation.
- The counter example can be quickly viewed and investigated.
- It is possible to consider several causes of failures inside an operational context, like a common cause failure deriving from a particular risk or pilot errors.

At present disadvantages are:

- performances of the tools when using verification techniques to perform the analysis
- quantitative analysis is not included in the same tool used for qualitative analysis, therefore it is needed to extract the results from ISAAC tool and to pass them to other tools for the calculation
- at present testability and maintenance aspects are not considered

The above considerations lead to the conclusion that the new method can partly be applied in the industrial context in short time.
This is valid for: construction of models appropriate for mission analysis, what-if simulation to determine the effects of failure modes occurrence depending on the operational phase. Several kind of failure modes can be considered: system items failure modes due to internal problems, failures induced by external problems like a common cause (e.g. tyre burst) and pilot errors.
Relevant to the use of verification techniques for the analysis, it can be said that these can be used in systems not so big in size and complexity (due to performance issues), otherwise the MRA subject, the case study, needs to be split into sub-parts.
Finally, for some ISAAC tool it is already possible to perform the quantitative analysis.

There are some aspects that have the potential for future development in the research field:

- to develop a formalized and structured way for defining manoeuvres for the mission phases (library of manoeuvres and requirements)
- to identify a method on how to capture environment information inside the MRA model and on how the verification tool can use environment info in order to represent and highlight them in the output fault tree
- to integrate testability and maintenance aspects in MRA context for more comprehensive analysis and for the purpose of developing an availability model
- to integrate result from HEA theme, i.e. to be able to use the "cognitive pilot model" in MRA tasks

4.2.8 TDS – System Testability/Diagnosability

4.2.8.1 Summary

The target of the **System Testability and Diagnosability** (TDS) theme is to extend the formal verification techniques and the methodology developed in ESACS (and improved in ISAAC) to deal with aspects related to testability of complex systems. This theme thus widens the scope of the ESACS platform by extending the applicability field of the platform to new disciplines related to the system design and analysis process.
The topics investigated within the TDS theme can be divided into two broad areas, namely *testability analysis* and *diagnosability analysis.*

The **testability** of a (complex) system is the capability of a (complex) system to precisely detects its failures (especially those safety critical), to alert the crew about the occurrence of unsafe or degraded system operating conditions through the generation of appropriate warnings (and/or maintenance messages) and, possibly, to take (or suggest) corrective actions. System testability is involved in maintaining safety levels required to a given safety critical application. Thus the objective of the testability analyses is the "*verification problem*", that is the starting point is a specification of the system and of its built in monitoring and the goal is the verification of its testability property.

The **diagnosability** of a (complex) system refers to the possibility to synthesize a diagnoser for the (complex) system. That is, given a system S and some requirements for the diagnoser (e.g., the possibility to detect fault X), verify whether it is possible to synthesize a diagnoser that satisfies the requirements. Thus the objective of the diagnosability analysis is the "*synthesis problem*", that is the starting point is a specification of the system, of one failure mode and of the *observables*[*] available from the system and the goal is to understand if it is possible to detect the failure mode under investigation with the available information and how. Where this is not possible (due to lack of information) the goal is to understand which additional information is needed.

The main objective of the TDS theme in ISAAC was to investigate testability aspects, whereas the aspects related to diagnosability have been left as an optional activity. Overall, we can say that all the objectives of the TDS theme have been reached in a satisfactory way, and all the essential requirements have been addressed.

The topics investigated in the TDS theme, and the outcome of the corresponding activities can be summarised as follows:

Fault Detection Analysis. The objective of Fault Detection Analysis is finding if and how faults are detected. Faults, in this context, are defined as the "activation" of one or more failure mode. By detection we mean the issue of a particular signal (or set of signals) every time the fault shows (i.e. the failure mode is activated).

The technical approach used to address the analysis was to re-use techniques implemented for the EPC theme, in particular FMEA analysis. Furthermore, the computation of the so-called Fault Coverage Index, which is a measure to quantify the effectiveness of the fault detection analysis, has been addressed. The generation of FMEA table for Fault Detection Analysis has been

[*] System parameters that are accessible by the monitoring system.

implemented both in the NuSMV-SA and in the STSA implementation lines. The available verification engines are VIS and NuSMV-SA (BDD-based).

Fault Isolation Analysis. The goal of Fault Isolation Analysis is starting from one, or more, messages and finding which failure modes cause it. In particular, it is checked in this step, whether the collection of messages displayed in certain situations is sufficiently detailed to identify the causing failure (combinations). Furthermore, the computation of the so-called Fault Isolation Index, which measures the effectiveness of the fault isolation analysis, has been addressed. The technical approach used to address the analysis was again to re-use techniques implemented for the EPC theme, in particular FTA analysis. The output of the FTA analysis is a fault tree, associating to a given message the minimal cut sets (or failure sequences) of failure modes causing the message to be issued.

Fault Isolation Analysis is available both in the NuSMV-SA and in the STSA implementation lines. The available verification engines are VIS and NuSMV-SA (the latter, both BDD-based and SAT-based).

False Alarm Analysis. The goal of False Alarm Analysis is to find if some messages are issued when no failure modes are present. Technically, the False Alarm Analysis was considered to be a particular sub-case of the Fault Isolation Analysis. Regarding the implementation, the same remarks as for Fault Isolation Analysis do apply.

Diagnosability Analysis. Some automatic techniques, used to synthesise a diagnoser, have been investigated. This approach falls under the scope of diagnosability analysis. In particular, the automatically synthesised diagnoser is guaranteed to satisfy the detection requirements by construction. Also by construction, automatically generated diagnosers are guaranteed not to raise false alarms. If the synthesis process is not able to find any diagnoser, then no diagnoser satisfying the given requirements exists.

4.2.8.2 Methodology

In this section we describe the methodology and the technical approach used to address the points outlined in the previous section.

Fault Detection Analysis. The inputs of a fault detection analysis are:

- A specification of what are the faults of which we want to verify the detection
- A specification of what are the messages issued by the diagnoser (so that the formal verification engine can recognize them)

The outputs of a fault detection analysis are:

- The list of messages that always occur when the faults activate and the values that these messages always have.

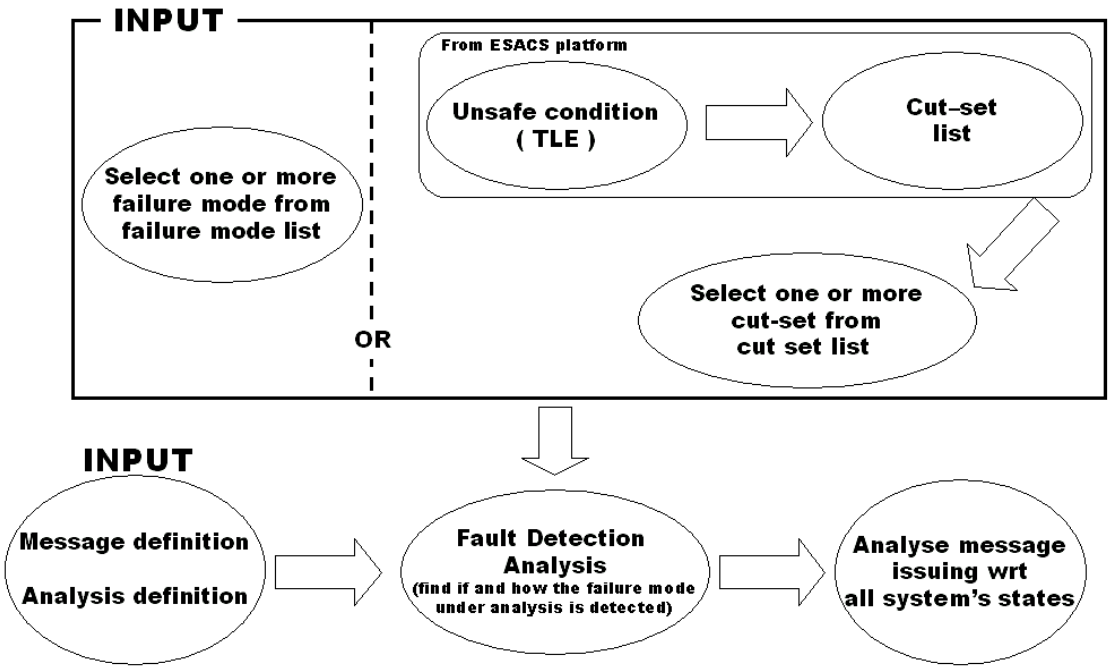

Figure 10 – Input/Output of fault detection analysis

The technical approach used to address the analysis was to re-use techniques implemented for the EPC theme. In particular, once the definition of the messages and of the failure modes is given, the fault detection analysis can be performed via a bottom-up analysis, that is, FMEA analysis. The traditional FMEA analysis performed in the EPC theme had to be modified in this context, to take into account the different semantics. In the EPC context, FMEA table have typically an "existential semantics", that is, a pair <FM config, E> in the table means that *there exists a path*, leading to the event E, showing the given failure mode configuration. In the TDS context, we have a "universal semantics", that is, a pair <FM config, M> in the table means that *for every pat*h showing the given failure mode configuration, eventually the message M will be raised.

The output of the FMEA analysis is a table describing the association between the failure modes (or combination thereof) and the messages activated.

The fault coverage index has been computed according to the following formula:

$$\%D = \frac{\text{Number of detected failures}}{\text{Total number of failures} - \text{Total number of failures Not Scored for BIT}} \cdot 100$$

Fault Isolation Analysis. The inputs of a fault isolation analysis are:

- The message (or the messages) that we want to understand from what failures modes are generated

And the outputs are:

- List of failure modes (or combinations thereof) that cause each message to be issued:

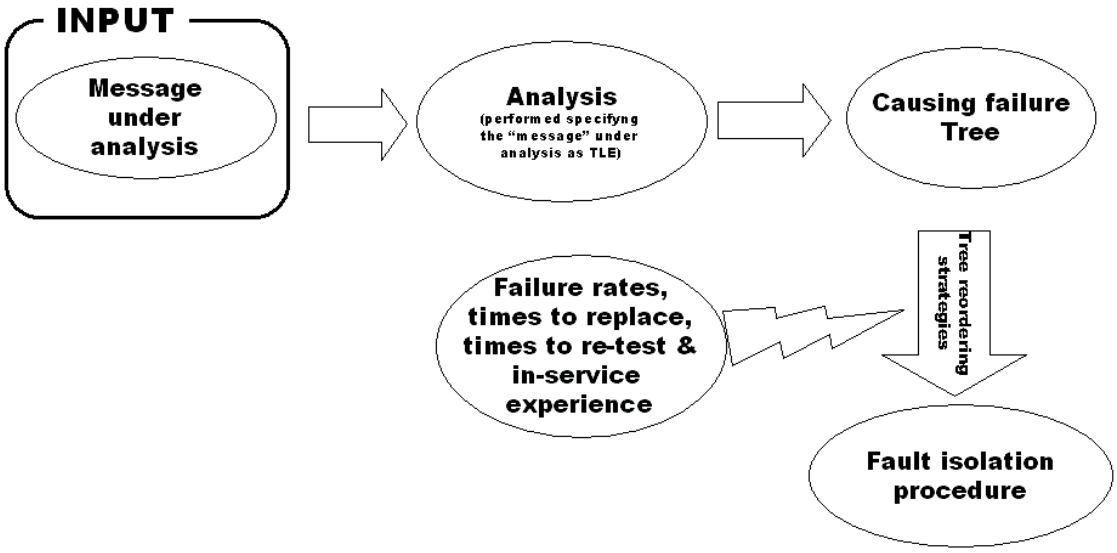

Figure 11 - Input/Output of fault isolation analysis

The technical approach used to address the analysis was again to re-use techniques implemented for the EPC theme. In particular, once the definition of the messages and of the failure modes is given, the fault detection analysis can be performed via a top-down analysis, that is, FTA analysis. The output of the FTA analysis is a fault tree, associating to a given message the minimal cut sets (or failure sequences) of failure modes causing the message to be issued.

The fault isolation index has been computed according to the following formula:

$$F = \sum_{g} MIN\left\{1, \frac{p(g)}{\sum_{g} p(g)}\, g\right\}$$

where *F* is a number comprised between 1 (best case) and *N* (worst case, where *N* is the total number of failure scored for BIT). In the best "design-for-testability" scenario each message is associated to one and only one failure mode. In general, however, each message is associated to one or more groups of failures (think at the cut-set of the causing failure tree). Each group is composed by one or more failures. So *g* is the number of different groups of failures, *p(g)* is the probability of the group *g* to occur.

False Alarm Analysis. The goal of False Alarm Analysis is to find if some messages are issued when no failure modes are present.

Technically, the False Alarm Analysis was considered to be a particular sub-case of the Fault Isolation Analysis. Given a particular message, the objective of the analysis is to assess whether it is possible that the message is issued without any failure mode being activated. This can be accomplished by building the fault tree for the message, as for Fault Isolation Analysis, and checking whether there is an empty cut set (that is, an empty fault tree is generated).

Diagnosability Analysis. Automatic synthesis was used for the ITC-IRST case study. This approach has the advantage that the synthesis of the diagnoser is completely automatic.

Extending the model in order to integrate the automatically generated diagnoser can also be done automatically.

Technically, the approach followed for the automatic synthesis of the diagnoser is based on planning techniques. Inputs to the planner are the system model and the fault(s) to be diagnosed. Given the system model and a fault to be diagnosed, using planning techniques it is possible to find a plan which, when followed by the diagnoser, is able to detect the fault. This is an example of *active diagnosis* (that is, the diagnoser may issue commands to the system model in order to detect the fault). The synthesis works under the hypothesis of partial observability (that is, the diagnoser is able to test only a subset of the signals in the system model). As a result, the plan may or may not exist (depending on the level of observability). If the plan exists, automatic techniques based on model checking can be used to synthesize it.

4.2.8.3 Impact and Applications

The objective of the TDS theme was to improve the current testability/diagnosability practices, and have the testability analysis people “speaking” the same language and using the same environment developed within the ESACS project for design and safety engineers.

As a general conclusion, we can say that the outcome of the evaluation activity performed to assess the methodology and the associated tools was positive.
The main benefits expected in the industrial context from the TDS theme can be summarised as follows:

- Model based design and verification techniques allow to address in more automated and comprehensive way testability issues since the early stages of system’s development
- The testability engineers can “speak” the same language and use the same environment developed for design and safety engineers
- The possibility to simulate a system model integrating testability aspects represents an added value for testability engineers, allowing them to perform a series of “what-if” simulations directly on the system model
- Fault isolation analysis provides an added value for the design engineer during the diagnoser design phase, and for the testability engineer during the assessment of the testability aspects
- False alarm analysis provides an added value during the diagnoser design phase, in order to understand if it is possible to have spurious alarms

It has to be said that at the end of the ISAAC project the TDS theme will have a methodology and some tool aspects that will be ready for a preliminary industrial application.
The scope of this preliminary industrial application should be to evaluate the ISAAC TDS approach with respect the traditional one in a real situation.
As impact on future activities we believe that some other tool aspects and functions, as well as “diagnosability analysis” are worth to be further developed in other follow-up ISAAC TDS research projects.

5. Dissemination and use

This section includes a publishable summary of each exploitable result the project has generated extracted from the Plan for Management of the Knowledge.

Exploitable result 1

Description:

- HEA simulation platform for Human-Machine-Interaction. The platform is used to investigate pilot behaviour in modern aircraft cockpits (e.g.: A340) during different flight scenarios, like for example takeoff or approach maneouvres. The main component of the simulation platform is a cognitive model, featuring learned carelessness as a possible source of pilot errors. This model is embedded within a complex simulation environment, consisting of an external flightsimulator environment, a Statemate system model (system under investigation, e.g. autoflight system), a Statemate scenario model (takeoff, descent) and scenario specific flight procedures.
  The goal is to find occurrences within the procedures that may be simplified due to learned carelessness which then may lead to pilot errors (e.g. entering faulty values, eliminated mode checks) and hazards on the aircraft level.

Market application:

- Currently the platform is used in the research context.

Stage of development:

- Work in progress. The simulation was initially developed by OFFIS and further improved during the ISAAC project.

Collaboration sought or offered:

- Collaboration in a RTD project in the 7th EU Framework Programme

Collaboration details:

- The goal is to extend the simulation platform in order to predict potential human errors taking into account a broader range of cognitive processes. In ISAAC it has been shown that cognitive models have the potential to be used for Human Error Analysis in order to predict potential pilot errors and their impact on flight safety. But a necessary prerequisite is a cognitive model that produced valid predictions for a larger set of pilot error types. Thus, a dedicated research project is planned to build up a knowledge foundation on human performance and underlying cognitive processes (with regard to deviations from normative activities) and use this knowledge to extend the cognitive model and the simulation platform.

Intellectual property rights granted or published: N/A

Contact details:

- Dr. Andreas Luedtke,
  OFFIS Safety Critical Systems, Escherweg 2, 26121 Oldenburg - Germany
  Email: luedtke@offis.de
  Homepage http://www.offis.de

Exploitable result 2

Description:

- HEA Procedure Editor: Was developed as a component of the HEA simulation platform in Project ISAAC, but is also usable as standalone version to create procedures for a cognitive model or other application contexts, where formalisation of procedures is useful. The editor allows creation of hierarchical goaltrees (goal, subgoal) with GOMS-style rules for each goal. Rules can contain motor and percept actions, conditions and memory functionalities. The editor also offers a viewer component which creates a graphical representation of the procedure. Procedures are stored in an .xml format. Exports exist for .txt and .tex format.

Market application:

- Currently the editor is used within the simulation platform in the research context.

Stage of development:

- Stable version exists, but also work under progress, depending on the simulation platform and the cognitive model.

Collaboration sought or offered:

- OFFIS offers collaboration to tailor the Procedure Editor to the needs of industrial partners. This includes improving the usability.

Collaboration details:

- This tailoring could be done in direct cooperation between OFFIS and industrial partners.

Intellectual property rights granted or published: N/A

Contact details:

- Dr. Andreas Luedtke,
  OFFIS Safety Critical Systems, Escherweg 2, 26121 Oldenburg - Germany
  Email: luedtke@offis.de
  Homepage http://www.offis.de

Exploitable result 3

Description:

- The FSAP safety analysis platform. The FSAP/NuSMV-SA platform aims at supporting design and safety engineers in the development and in the safety assessment of complex systems. The platform consists of a graphical user interface (FSAP) and an engine (NuSMV-SA), based on the NuSMV model checker. It is based on the concept of a repository to allow sharing of information between design and safety. The platform is designed to support different phases of the development and safety assessment process and to support different development and safety assessment practices. To achieve these goals, FSAP/NuSMV-SA provides a set of basic functions, which can be combined in different ways to perform complex tasks. The platform automates the generation of artifacts that are typical of reliability analysis, for example fault trees. The major benefits from the use of FSAP/NuSMV-SA are a tight integration between the design and the safety teams, and the automation of some of the activities related both to the verification and to the safety analysis of systems in a uniform environment.

Market application:

- Safety assessment of complex systems, included, but not limited to, avionics, railways, automotive, energy production, industrial control

Stage of development:

- Current version used for research purposes. Possible commercialization under way.

Collaboration sought or offered:

- Collaborations with other research /industrial partners under investigation.

Collaboration details: N/A

Intellectual property rights granted or published:

- None.

Contact details:

- FSAP Team
  Automated Reasoning Systems Division
  ITC-irst
  Via Sommarive 18, 38123 Trento, Italy
  E-mail: fsap@fbk.eu
  Homepage: http://sra.fbk.eu/tools/FSAP/

Exploitable result 4

Description:

- STSA (STatemate Safety Analysis) is an integrated tools platform to enable automated analysis and assessment of dynamic safety critical systems. STSA supports the safety assessment process in the avionics domain as described in the ARP 4761 and closely follows classics methods like fault-tree analysis (FTA) and failure-mode and effects analysis (FMEA). The system builds upon an approach that uses well established formal methods (e.g. model-checking) for the analysis of formal system models. The platform has been developed with the intention to provide an integrated and concise work environment to systematically evaluate the consequences of failures. Different analysis task provide the possibility to check the validity of assumptions and establish dependencies that can be used to refine existing and derive additional safety requirements.

Market application:

- STSA has been developed for use throughout the development process of safety critical embedded systems. The original application is in the avionics domain but the techniques are also applicable to other domains such as automotive and railways.

Stage of development:

- The tool platform has been developed in research projects for the avionics domain (ESACS / ISAAC). The core functionality is stable and has been evaluated using case studies during the ISAAC project. Adaptation to other applications domains and for commercial users is intended.

Collaboration sought or offered:

- Collaboration in a RTD project in the 7th EU Framework Programme
- Collaboration with industrial partners to integrate the developed techniques and tools into their development process

Collaboration details:

- OFFIS will to demonstrate the use and applicability of the STSA for industrial partners and offers to support the integration of the tools into the development process. Where necessary adaptations to the tool platform (e.g. to support different CASE tools) can be developed together with the industrial partners.

Intellectual property rights granted or published:

- N/A

Contact details:

- Thomas Peikenkamp
  OFFIS Safety Critical Systems, Escherweg 2, 26121 Oldenburg - Germany
  Email: peikenkamp@offis.de
  Homepage http://www.offis.de

Exploitable result 5

Description:

- OFFIS has developed methods and tools that enable the complete detection of system deviations by means of failure detection. These techniques identify the relevance of all possible system deviations to the full extend by checking all paths possible within a system including paths of infinite length. These techniques have been integrated into STSA (STatemate Safety Analysis), an integrated tool platform that enables the automated analysis and assessment of dynamic safety critical systems. The system builds upon an approach that uses well established formal methods (e.g. model-checking) for the analysis of formal system models. The platform has been developed with the intention to provide an integrated and concise work environment to systematically evaluate the consequences of failures. Different analysis task provide the possibility to check the validity of assumptions and establish dependencies that can be used to refine existing and derive additional safety requirements.

Market application:

- STSA has been developed for use throughout the development process of safety critical embedded systems. The original application is in the avionics domain but the techniques are also applicable to other domains such as automotive and railways

Stage of development:

- The tool platform has been developed in research projects for the avionics domain (ESACS / ISAAC). The core functionality is stable and has been evaluated using case studies during the ISAAC project. Adaptation to other applications domains and for commercial users is intended.

Collaboration sought or offered:

- Collaboration in a RTD project in the 7th EU Framework Programme
- Collaboration with industrial partners to integrate the developed techniques and tools into their development process

Collaboration details:

- OFFIS will to demonstrate the use and applicability of the STSA for industrial partners and offers to support the integration of the tools into the development process. Where necessary adaptations to the tool platform (e.g. to support different CASE tools) can be developed together with the industrial partners.

Intellectual property rights granted or published:

- N/A

Contact details:

- Thomas Peikenkamp
  OFFIS Safety Critical Systems, Escherweg 2, 26121 Oldenburg - Germany
  Email: peikenkamp@offis.de
  Homepage http://www.offis.de

Exploitable result 6

Description:

- The ISAAC projects developed methods and tools that enable to analyze the coverage of Build-in test equipment (BITE) regarding all possible scenarios of a system that is specified using a formal model. They allow to determine to what degree BITE is able to produce messages for all possible failure scenarios and to determine to which degree it is possible to distinguish different failure scenarios by means of the issued messages. These techniques have been integrated by OFFIS into STSA (STatemate Safety Analysis), an integrated tool platform that enables the automated analysis and assessment of dynamic safety critical systems. The system builds upon an approach that uses well established formal methods (e.g. model-checking) for the analysis of formal system models. The platform has been developed with the intention to provide an integrated and concise work environment to systematically evaluate the consequences of failures. Different analysis task provide the possibility to check the validity of assumptions and establish dependencies that can be used to refine existing and derive additional safety requirements.

Market application:

- STSA has been developed for use throughout the development process of safety critical embedded systems. The original application is in the avionics domain but the techniques are also applicable to other domains such as automotive and railways

Stage of development:

- The tool platform has been developed in research projects for the avionics domain (ESACS / ISAAC). The core functionality is stable and has been evaluated using case studies during the ISAAC project. Adaptation to other applications domains and for commercial users is intended.

Collaboration sought or offered:

- Collaboration in a RTD project in the 7th EU Framework Programme
- Collaboration with industrial partners to integrate the developed techniques and tools into their development process

Collaboration details:

- OFFIS will to demonstrate the use and applicability of the STSA for industrial partners and offers to support the integration of the tools into the development process. Where necessary adaptations to the tool platform (e.g. to support different CASE tools) can be developed together with the industrial partners.

Intellectual property rights granted or published:

- N/A

Contact details:

- Thomas Peikenkamp
  OFFIS Safety Critical Systems, Escherweg 2, 26121 Oldenburg - Germany
  Email: peikenkamp@offis.de
  Homepage http://www.offis.de

Exploitable result 7

Description:

- OFFIS has developed methods and tools that enable the interactive simulation of failure-scenarios using the interactive simulation platforms of case tools to describe the behavior of systems for different failure scenarios. User friendly representations of failure scenarios allow deriving interactive simulation of system models by failure scenarios computed via formal safety analysis. These techniques have been integrated into STSA (STatemate Safety Analysis) , an integrated tool platform that enables the automated analysis and assessment of dynamic safety critical systems. The system builds upon an approach that uses well established formal methods (e.g. model-checking) for the analysis of formal system models. The platform has been developed with the intention to provide an integrated and concise work environment to systematically evaluate the consequences of failures. Different analysis task provide the possibility to check the validity of assumptions and establish dependencies that can be used to refine existing and derive additional safety requirements.

Market application:

- STSA has been developed for use throughout the development process of safety critical embedded systems. The original application is in the avionics domain but the techniques are also applicable to other domains such as automotive and railways

Stage of development:

- The tool platform has been developed in research projects for the avionics domain (ESACS / ISAAC). The core functionality is stable and has been evaluated using case studies during the ISAAC project. Adaptation to other applications domains and for commercial users is intended.

Collaboration sought or offered:

- Collaboration in a RTD project in the 7th EU Framework Programme
- Collaboration with industrial partners to integrate the developed techniques and tools into their development process

Collaboration details:

- OFFIS will to demonstrate the use and applicability of the STSA for industrial partners and offers to support the integration of the tools into the development process. Where necessary adaptations to the tool platform (e.g. to support different CASE tools) can be developed together with the industrial partners.

Intellectual property rights granted or published:

- N/A

Contact details:

- Thomas Peikenkamp
  OFFIS Safety Critical Systems, Escherweg 2, 26121 Oldenburg - Germany
  Email: peikenkamp@offis.de
  Homepage http://www.offis.de

Exploitable result 8

Description:

- OFFIS has developed methods and tools that enable the adaptation of propositional abstraction techniques for use with automatic safety analysis of dynamic systems. They provide a user-friendly abstraction mechanism that make more complex model available for automated safety analysis. These techniques have been integrated into STSA (STatemate Safety Analysis) , an integrated tool platform that enables the automated analysis and assessment of dynamic safety critical systems. The system builds upon an approach that uses well established formal methods (e.g. model-checking) for the analysis of formal system models. The platform has been developed with the intention to provide an integrated and concise work environment to systematically evaluate the consequences of failures. Different analysis task provide the possibility to check the validity of assumptions and establish dependencies that can be used to refine existing and derive additional safety requirements.

Market application:

- STSA has been developed for use throughout the development process of safety critical embedded systems. The original application is in the avionics domain but the techniques are also applicable to other domains such as automotive and railways

Stage of development:

- The tool platform has been developed in research projects for the avionics domain (ESACS / ISAAC). The core functionality is stable and has been evaluated using case studies during the ISAAC project. Adaptation to other applications domains and for commercial users is intended.

Collaboration sought or offered:

- Collaboration in a RTD project in the 7th EU Framework Programme
- Collaboration with industrial partners to integrate the developed techniques and tools into their development process

Collaboration details:

- OFFIS will to demonstrate the use and applicability of the STSA for industrial partners and offers to support the integration of the tools into the development process. Where necessary adaptations to the tool platform (e.g. to support different CASE tools) can be developed together with the industrial partners.

Intellectual property rights granted or published:

- N/A

Contact details:

- Thomas Peikenkamp
  OFFIS Safety Critical Systems, Escherweg 2, 26121 Oldenburg - Germany
  Email: peikenkamp@offis.de
  Homepage http://www.offis.de

Exploitable result 9

Description:

- OFFIS has developed methods and tools that enable scenario based specification of safety requirements of dynamic systems by means of sequence diagrams. They provide a user-friendly way of specifying complex temporal requirements by means of intuitive scenario descriptions and extend concepts for describing safety critical situations by including timing information and rigorous timing analysis techniques. These techniques have been integrated into STSA (STatemate Safety Analysis), an integrated tool platform that enables the automated analysis and assessment of dynamic safety critical systems. The system builds upon an approach that uses well established formal methods (e.g. model-checking) for the analysis of formal system models. The platform has been developed with the intention to provide an integrated and concise work environment to systematically evaluate the consequences of failures. Different analysis task provide the possibility to check the validity of assumptions and establish dependencies that can be used to refine existing and derive additional safety requirements.

Market application:

- STSA has been developed for use throughout the development process of safety critical embedded systems. The original application is in the avionics domain but the techniques are also applicable to other domains such as automotive and railways

Stage of development:

- The tool platform has been developed in research projects for the avionics domain (ESACS / ISAAC). The core functionality is stable and has been evaluated using case studies during the ISAAC project. Adaptation to other applications domains and for commercial users is intended.

Collaboration sought or offered:

- Collaboration in a RTD project in the 7th EU Framework Programme
- Collaboration with industrial partners to integrate the developed techniques and tools into their development process

Collaboration details:

- OFFIS will to demonstrate the use and applicability of the STSA for industrial partners and offers to support the integration of the tools into the development process. Where necessary adaptations to the tool platform (e.g. to support different CASE tools) can be developed together with the industrial partners.

Intellectual property rights granted or published:

- N/A

Contact details:

- Thomas Peikenkamp
  OFFIS Safety Critical Systems, Escherweg 2, 26121 Oldenburg - Germany
  Email: peikenkamp@offis.de
  Homepage http://www.offis.de

Exploitable result 10

Description:

- The FTA Manager used in combination with SCADE is a platform to support designers and safety engineers in the system development process. The platform supports development and analyses for any type of system, e.g. controlling, built in test and automation. The FTA Manager presents a way to introduce basic failure events, which are accessible from a failure mode library, and to specify sets of basic failure events (related to common cause events).
  The analysis technique offers a way to investigate both the correctness of the design as well as failure influence on a specified safety requirement. There is a variety of analysis options, e.g. to calculate a list of all minimal cut sets – which can consist of a mixture of basic failure events and common cause events – with a specified maximum number of events or to calculate a detailed counter example illustrating how a requirement is violated. It is also possible to impose temporal restrictions to the occurrence of failures, e.g. duration, delays and number of occurrences. Some special analyses offered are: “importance of ordering” and “cut set implication of requirement violation”.
  To each minimal cut set there is a counter example, which is presented by a model viewer making it possible for the user to explore the causes for a requirement violation.

Market application:

- The SCADE/FTA Manager platform can be applied in most technical applications. It has already been applied to the avionics, railway and automotive sectors.

Stage of development:

- Current version is a prototype but still user-friendly enough to be used as it is for many types of analysis. Depending on future special industrial needs it may require further adjustments to become a product.

Collaboration sought or offered:

- Collaboration sought with SP (Swedish National Testing and Research Institute) to have a joint project for the energy industry.
- Collaboration investigated with Airbus to further explore the use of model checking techniques as well as analyses offered by the FTA Manager.

Collaboration details: N/A

Intellectual property rights granted or published:

- None

Contact details:

- Ove Åkerlund
  Prover Technology AB
  E-mail: ove.akerlund@prover.com
  Homepage http://www.prover.com

6. Conclusion

The results reached in the project can be applied in the industrial activities to support the “Safety“ and “Design“ Processes involved in the Safety Assessment Analysis required by the Standards, in a more integrated and efficient manner.

It would be possible to identify design points of investigation and design alternatives with timeliness, then quickly verify them again using the ISAAC Framework.

Moreover, some ISAAC results have the potential for further development in future research activities and for application in large context, like the Joint Technology Initiatives environment (Clean Sky, Single European Sky ATM Research - SESAR).