THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par *l'Université Toulouse 3 – Paul Sabatier*
Discipline ou spécialité : *Informatique*

Présentée et soutenue par *Martina Ragosta*
Le 7 mai 2015

Titre : *A multi-modelS based approach for the modelling and the analysis of usable and resilient partly autonomous interactive systems*

JURY
Maria de Fátima Queiroz Vieira (rapporteur)
Francisco Javier Sáez Nieto (rapporteur)
Jean-Claude Tarby (examinateur)
Yamin Ait Ameur (examinateur)
Philippe Palanque (directeur de thèse)
Célia Martinie (co-encadrant)
Alberto Pasquini (co-encadrant)

Ecole doctorale: *Mathématiques Informatique et Télécommunications de Toulouse (MITT)*
Unité de recherche: *IRIT – UMR 5505*
Directeur(s) de Thèse: *Philippe Palanque et Célia Martinie*
Rapporteurs: *Maria de Fátima Queiroz Vieira et Francisco Javier Sáez Nieto*

Acknowledgements

Broadly speaking and, like as a good PhD student looking at the other theses trying to make a state of the art for this section, there seems to be an order for listing the persons who is needed to acknowledge: supervisors (main and second), academic and technical staff in the department, colleagues, institutions and organization, companies, and finally family and friends. Even if I tried with all my might and main, I cannot be sure that the suggested sequence has been properly followed for two main reasons. The former is that most of the persons have played more than one role making difficult the distinction between professional and personal life. The latter is that each person has been so important that list them, all of them (giving a priority), it is quite hard. So, I would like to start saying *"Merci beaucoup", "Muy obrigada", "Grazie mille", "Muchas gracias", "Vielen Danke", "Tusen takk"* to **ALL!**

I would like to express my gratitude to Philippe as supervisor. He would have never accepted anything less than my best efforts, and for that, I thank him. As well as inspiring me on an academic level, I have learnt so many other things from Philippe! I really feel lucky to have had a supervisor that I can also call a friend, someone who I was able to laugh with even when the thesis and everything else around me was driving me crazy. Thanks to him, each day that I spent in Toulouse and in the ICS team has been a beautiful day ("che bella giornata!").

I would like to express my deepest appreciation to my co-supervisor, Célia, for her excellent guidance, caring, patience, and providing me with an excellent atmosphere for doing research. We have shared so many things… I don't know if my high regard for Célia is more for her as a precious friend or for her as an inestimable co-supervisor. I would just to say "Merci de tout coeur" to Célia for being simply Célia with me.

Both, Philippe and Célia, have shown the attitude and the substance of a genius: they continually and persuasively conveyed a spirit of adventure in regard to research and an excitement in regard to teaching. Without their supervision and constant help this dissertation would not have been possible.

I would like to express my gratitude to everyone who supported me throughout the course of this PhD thesis. I am thankful for their aspiring guidance, invaluably constructive criticism and friendly advice during these years. I am sincerely grateful to them for sharing their truthful and illuminating views on a number of issues related (or not) to this research activity.

The ICS team: Eric Barboni, David Navarre, Racim, Arnaud Hamon, Caio, José, Cédric Bach, Adrienne Tankeu-Choitat, Christelle Farenc. In particular, Marco and Camille who are very good friends. They were always willing to help and give their best suggestions. All of you are really ICS women and men, my super heroes!

The Ruido team: Regina Bernhaupt, Thomas Mirlacher and the "BB" guys. In particular, Michael Pirker, my Geppi.

The Hala! Network: Paco for each beer drunk together, for each smile and hug, for putting his trust in me. I always felt his support and friendship. Eduardo for his availability, his presence, and his courage. For all the moments (and places) in which our lives have crossed and for his positive attitude in looking at the daily difficulties. The other PhD students…in particular, my German and "E"Spanish friends. We have shared working and personal worries and, thanks to this reciprocal support, we have grown up. I will forever remember in my heart the stories and the moments that all of you have shared with me during these years.

The All Deep Blue colleagues: Simone and Luca who saw something in me I didn't see in myself, my darling Paola, Sara who has been a disciple of Mr. Miyagi and we know what this means, Alberto who, before being my boss, is one of my favourite traveling and adventure companions. To All DBluers, I would like to express my warm thanks to and for their support and understanding.

I would also like to thank my parents. They were always supporting me and encouraging me with their best wishes. I always knew that you believed in me and wanted the best for me.

Finally, I would like to thank my friends that were always there cheering me up and stood by me through the good and bad times. Thank you for teaching me that my job in life was to learn, to be happy, and to know and understand myself; only then could I know and understand others. It is impossible to list all of you, but you are the bright part of my soul.

This PhD has been one of the most difficult challenges, but also, one of the best and unforgettable experiences in my life ***only thanks to all of you***!

Table of content

List of figures

List of tables

Introduction

Socio-Technical Systems (STS) (Emery & Trist, 1960) are complex systems encompassing three main components: the **system** (usually a computer-based one) the **human** (usually a trained operator with validated qualification for operating the system) and the **organisation** (usually a large entity composed of several organizational layers ranging from the local organisation where the operator is located to higher level ones such a regulatory entities provided high-level rules for the Socio-Technical System).

Dealing with such large scale systems require multi-disciplinary approaches such as electronics and computer science (for the system part), psychology and human factors (for the human part) and organization sciences (for the organisation part of the STS). Of course as the operator is interacting with the computing system, Human-Computer Interaction (HCI) issues arise and have to be addressed adequately. Beyond that, these STS are most of the time large scale distributed systems where groups of operators interact altogether to achieve the required functions of the STS. This adds all the disciplines related to collaborative aspects of Computer Supported Cooperative Work (CSCW) including distributed systems and Human-Computer Interaction (at coordination, production and communication levels). Due to the deep interleaving of the three parts of the STS, their evolution can sometimes be (e.g. under adverse conditions) non-linear making its overall performance hardly predictable.

Usually researchers take a very limited view at the development of STS focussing on some specific aspects and hiding away other ones. This allows providing local solutions to the development issues but their integration is usually claimed to be beyond the scope of the research. This thesis proposes an integrated approach addressing all these aspects in a common framework that is described later on. In order to reduce the field of investigation, the thesis focusses on Air Traffic Management (ATM) Socio-Technical System systems that are considered complex as they encompass interactions involving multiple kinds of operators, various dedicated computing systems, multiple regulatory authorities and their performance is deeply influenced by environmental aspects (i.e. weather, organizational variability (e.g. strikes) or system failures).

Above all, the current European ATM System needs to be improved for coping with the growth in air traffic forecasted for next years. Traffic is also likely to become more heterogeneous integrating Remotely Piloted Aircraft Systems (RPAS) and possibly civil aircraft with reduced flying crew (also known as single pilot operations). These two aspects require evolutions to increase ATM capacity and to guarantee that safety level is not reduced. One way towards these objectives is to increase automation in the computing part of the ATM Socio-Technical System. In that case, automation has to be considered as an enabler to improve capacity (and possibly safety) of the ATM System, considering this improvement not from any partial view point, but from an overall System performance perspective. As automation should provide support to increase capacity by empowering the operators and then improving the overall performance, the partly autonomous interactive systems require to be *usable*. As higher levels of automation should not reduce safety level, the partly autonomous interactive systems within their associated STS have to be *resilient*.

Automation then is not seen to replace operators but to empower them and to improve the overall performance of ATM as clearly defined by the research network on Higher Automation Levels in

Aviation (HALA!) (HALA! SESAR Research Network, 2012). This approach is similar to the one followed in the early 80's when flying crew for large civil aircraft was reduced from three to two by adding sophisticated Flight Management Systems (Wise, Hopkin, & Garland, 2009). However, it is important to design this automation very carefully taking into account the three parts of the STS. As clearly pointed out by Lisanne Bainbridge (Bainbridge, 1983) automation malfunctions end up most of the time in the hands of operators that were precisely supported with automation as their tasks were too complex or too resource consuming. Moreover, introducing higher level of automation requires (beyond design issues) an evaluation of the impact which the new technology may have on each STS part (operator, computing system and organisation) such as tasks migration and/or functions allocation (Jordan, 1963). Such function allocation (as illustrated in Figure 1), concerns the ground side of the ATM System i.e. the Air Traffic Control Centres (ATCs). On the top left-right side of Figure 1, Air Traffic Controllers communicate with pilots via data link or transfer aircraft interacting on the electronic labels of the aircraft on the radar screen instead of using paper strips and communicating by voice using VHF medium. Similarly, on the airborne side (lower part of Figure 1), glass cockpits (Sweet, 1995) provide a means for integrating information to support pilots activities while this information was previously distributed amongst multiple displays throughout the cockpit. In both cases, task migration and/or functions allocation require that humans improve their knowledge, learn how to interact and collaborate with the new technology for accomplishing tasks.

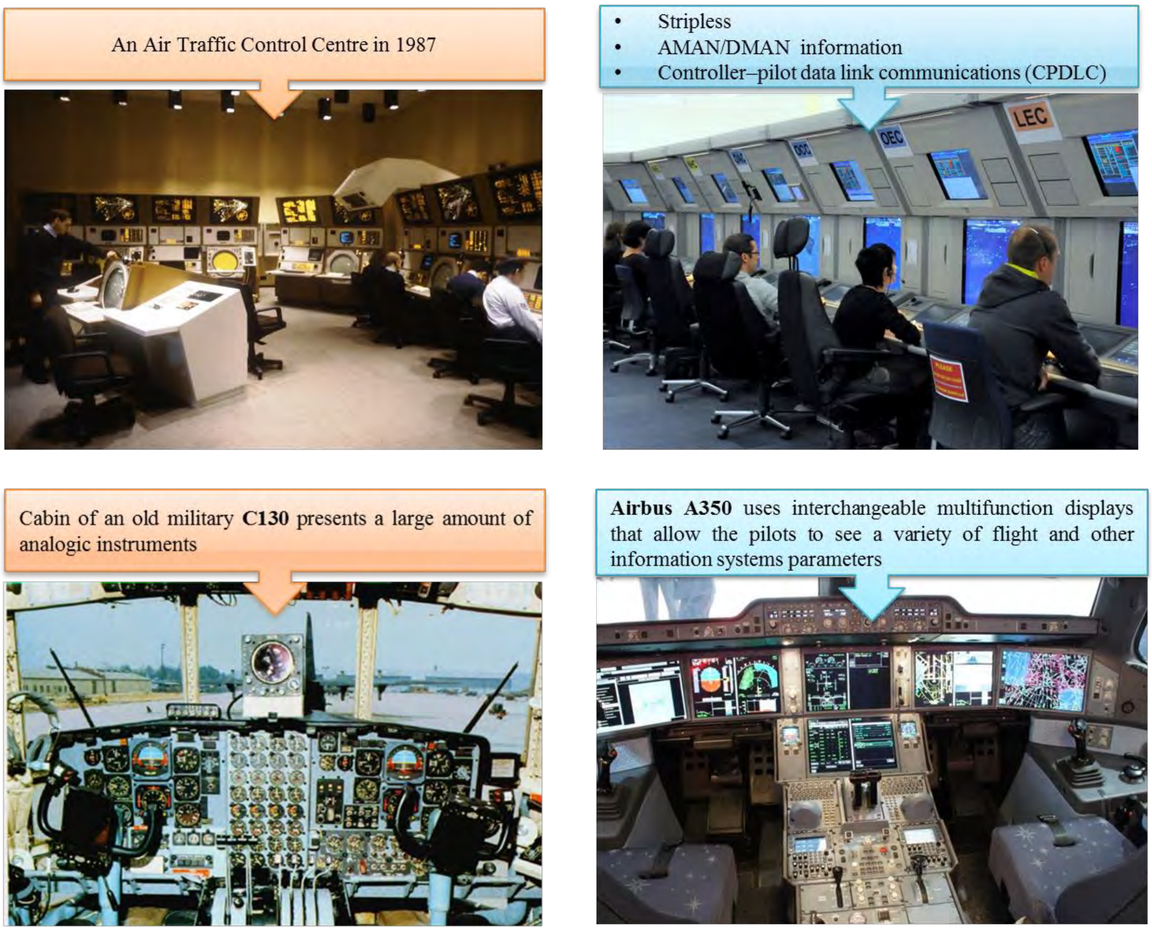


Figure 1: The enhancement of integrated automation support in ATCs and cockpits over time

In automated systems, function allocation (Older, Waterson, & Clegg, 1997) between human and machine has always been a point of controversy. In the context of automation, "functions allocation" means that the actor (either human or machine) that is best suited (based on some continuum of parameters) should perform the function. The basis for selection and grading of such parameters is at the heart of the issue of function allocation and has been subject to much investigation over the years, from Fitts with his MABA-MABA list (Fitts, 1954), through Sheridan et al. (Sheridan & Verplank, 1978) who have constructed stepwise function allocation models of automation using two main dimensions – proper task and proper level –, until Parasuraman's et al. (Parasuraman, Sheridan, & Wickens, 2000) information processing model.

Tasks usually performed by human operators, may be partially or fully automated. Most of the time automation is only partial keeping the operators in the loop so that they can forecast what will happen next and interfere with automation in case of adverse events or automation malfunction. The design of this cooperation requires understanding how to balance automation and interactivity and specify how a task can be performed by assigning the generic functions to the operator and the system in terms of function allocation. "Function allocation cannot be based on a consideration of the tasks only, but must consider the total equilibrium of a work situation—corresponding to a notion of balanced work. The concept of equilibrium emphasises the fact that a change in function allocation disturbs the established equilibrium. This will have consequences for the system as a whole, and one result may be that a new equilibrium is established which differs significantly from the previous one." (Bye, Hollnagel, & Brendeford, 1999, p. 292)

Previous work on automation can be divided according to three different perspectives: 1) the design perspective which focuses on how to engineer the computing systems (offering automation) and more precisely its user interface (Bindewald, Miller, & Peterson, 2014), 2) the evaluation perspective which focuses on how to assess the operational aspects of automation including performance impact of automation on operations (Parasuraman & Riley, 1997) (Kaber & Endsley, 2004), and 3) the human perspective which focuses on how to understand the role of the operators who deal with a new technology or a different level of automation (Parasuraman, Sheridan, & Wickens, 2000) (Save & Feuerberg, 2012). While this research work has been mostly conducted in separate fields, as the increase of automation might come along with an increase of performance variability of the whole ATM System especially in case of automation degradation, there is a need to provide an integrated view on these disjoint research activities.

Contribution

To address all the issues introduced above, this thesis proposes a multi-models based approach for the modelling and the analysis of partly-autonomous interactive systems within a STS for assessing their resilience and usability.

The design driver of the contribution was twofold:

- Address all the aspects of the STS in a single integrated framework
- Address the entire development process of STS from design to evaluation with a special emphasis on automation dependability as this is key in Lisanne Bainbridge (Bainbridge, 1983) view on ironies of automation.

Addressing all the aspects of the STS would be clearly unachievable without tackling it at the right level of abstraction. Indeed, it would be very easy to get lost too early in low-level details or to miss important aspects by focussing too much on a specific part. For this reason one of the main hypotheses of this thesis is that using an adequate model will allow us to describe the STS at the adequate level both for design and analysis purposes. The other hypothesis is that due to the very different nature of the three parts of the STSs one model will not fit all but different models have to be used and, of course, that these various models will have to be integrated to describe the entire relevant elements of the STS. While the thesis focuses on ATM systems we will separate the definition of the models and the process for using them from the ATM case studies. This will demonstrate that the approach is not dedicated to ATM but is also suitable to other large scale STSs (even though some adjustments might be required to manage their idiosyncrasies).

Thus, the contribution of this thesis is first based on the identification of a set of requirements needed being able to model and analyse each of the STS elements. Some of these requirements were met by existing modelling techniques, others were reachable by extending and refining existing ones. This thesis proposes an approach which integrates three modelling techniques: FRAM (focused on organisational functions), HAMSTERS (centred on operator goals and activities) and ICO (dedicated to the modelling of interactive systems and supporting the development of dependable computing systems). The integration of models has been done at a detailed level through a dedicated process defining in a stepwise manner how to go from one model to another one but also how information represented in one model is reused and possibly refined in another one.

The principle of the multi-models approach is illustrated on an example for carefully showing the extensions proposed to the selected modelling techniques and how they integrate together. A more complex case study from the ATM World is then presented to demonstrate the scalability of the approach. This case study, dealing with aircraft route change due to bad weather conditions, highlights the ability of the integration of models to cope with performance variability of the various parts of the STS.

Structure

Chapter 1, Chapter 2 and Chapter 3 form Part 1 of this thesis which presents an overview of the state of the art relating to the various domains relevant to this research, including tasks, systems and organisationals modelling techniques and associated methods for respectively addressing human activities, systems and related user interactions, and organisational aspects focusing on usability and resilience analysis of a partly interactive autonomous system within a STS.

Chapter 4 begins Part 2 of the thesis, the Contributions one, which introduces the first contribution regarding one of the introduced modelling techniques called HAMSTERS. The subsequent sections present the extensions that have been added to the notation in order to allow the representation of elements of declarative knowledge and in order to enable their integration within the procedural knowledge description of HAMSTERS 2.0 (please refer to Section 1). The main changes with regard to HAMSTERS 1.0 correspond to the explicit handling of declarative and procedural knowledge elements within a concept map and to the explicit representation of these elements in the task model (please refer to Section 2).

Chapter 5 presents the second contribution of this thesis, the multi-models based approach which is the process for integrating the three selected modelling techniques in order to model and to analyse a partly autonomous interactive system within a Socio-Technical System and to assess its resilience and usability properties. Furthermore, this approach provides support for examining the system under analysis at different levels of granularity.

Both contributions are carefully explained via an illustrative example.

Chapter 6 concludes Part 2. This chapter proposes a more complex case study taken from the ATM World to demonstrate the scalability of the approach. This case study, dealing with aircraft route change due to bad weather conditions, highlights the ability of the integration of models to cope with performance variability of the various parts of the STS.

Finally, the last Chapters present the “Conclusions” and offer “Perspectives” regarding the application of the developed approach and a list of selected “Personal publications”.

PART I – State Of the Art

Chapter 1 – Addressing human activities through task modelling techniques and associated methods

This chapter describes the current state of the art of task modelling techniques and methods used to address the description of human activities which are needed to interact with a partly autonomous interactive system. The scope of this chapter is to:

- Explain why task analysis and modelling are needed to analyse and assess usability and resilience of partly autonomous interactive systems
- Describe and compare the existing task modelling techniques and methods
- Examine what is still missing in existing task modelling techniques and methods to provide support for the modelling and the analysis of usable and resilient partly autonomous interactive systems with regard to human activities which are needed to interact with this kind of systems. Moreover, it proposes a set of requirements for improving these task modelling techniques and methods in order to integrate knowledge representation in task models
- Make a comparison of task modelling techniques and methods according to the identified requirements.

The first section illustrates how task analysis is at the core of most work in human-computer interaction because it is concerned with the performance of work (Diaper, 2004).

The second section offers a state of the art on task modelling techniques and methods. It aims at describing and comparing the existing techniques and it also includes a deeper description of the HAMSTERS task modelling technique which has been developed by the ICS team and which is currently being used in research projects related to the ATM domain to investigate the human behaviour. HAMSTERS stands for Human-centered Assessment and Modelling to Support Task Engineering for Resilient Systems and is tagged version 1.0 (HAMSTERS 1.0) in part 1 of this manuscript. HAMSTERS version 2.0 integrates the contribution proposed by this thesis and is presented in part 2 of this manuscript.

The third section highlights what is still missing in existing task modelling techniques and methods and identifies requirements for addressing human activities, which are needed to interact with a partly autonomous interactive system, through task modelling techniques and methods for the modelling and the analysis of usable and resilient partly autonomous interactive systems.

The fourth section offers a comparison of task modelling techniques and methods according to the identified requirements.

1 Task analysis and modelling to support analysis and assessment of usability and resilience of partly autonomous interactive systems

As stated by Annett (Annett, 2004) “Analysis is not just a matter of listing the actions or the physical and cognitive processes involved in carrying out a task, although it is likely to refer to either or both. Analysis, as opposed to description, is a procedure aimed at identifying performance problems and proposing solutions." Task analysis produces one or more models of the world and these models describe the world and how work is performed in it (Diaper, 2004) .

Task models provide abstract and refined descriptions of human activities. They contain:

1. a description of the work that has to be performed by an operator and the goals s/he has to accomplish,
2. a refinement of these goals into human actions.

There are many reasons for developing task models. Task model of human activities required to operate an existing system can be created in order to better understand the underlying design, to analyse its potential limitations and how to overcome them (Paternò, 2004). Task models can also be recorded and used in a complementary way with other data collection methods (Law, et al., 2009) such as interviews and scenario based approaches (Rosson & Carroll, 2001). Task analysis and modelling techniques provide a unique way of understanding and analysing in systematic way users' roles and activities. They can be useful for supporting analysis and assessment of usability (Greenberg, 2004) (Law, et al., 2009) and resilience (Hollnagel, 2006) of partly autonomous interactive systems. As illustrated in Figure 2, the complexity of the real work can be analysed through task analysis. The complexity of the real work is filtered out through a purple filter that represents a task analysis. This "task analysis" filter allows focusing on specific pieces of information about the real work by using task analysis notation elements to describe real work activities. The aim of this filtering out operation is to obtain a description of the assumed real work. The description of the real work relies on the chosen notation, i.e. on aspects that are relevant to the analyst. The analyst may then choose between several task modelling techniques to record this description in task models. Each task modelling technique can provide a different way to represent the task. This point is illustrated in Figure 2 where there are 2 purple glasses for representing different task modelling techniques and their specific notations such as Task Architect (Stuart & Penn, 2004) in the upper part, or HAMSTERS 1.0 (Martinie, Palanque, & Winckler, 2011) in the lower part (Section 2.1 presents a state of the art on existing notations). Depending on the choice of notation, the task modelling operation will result in a different model of the activities since each type of model highlights particular aspects of the assumed real work. For this reason, it is important to choose the most suitable task modelling technique. Thanks to the selected task modelling technique, the analyst can highlight the aspects that are relevant to the goals of his/her analysis.

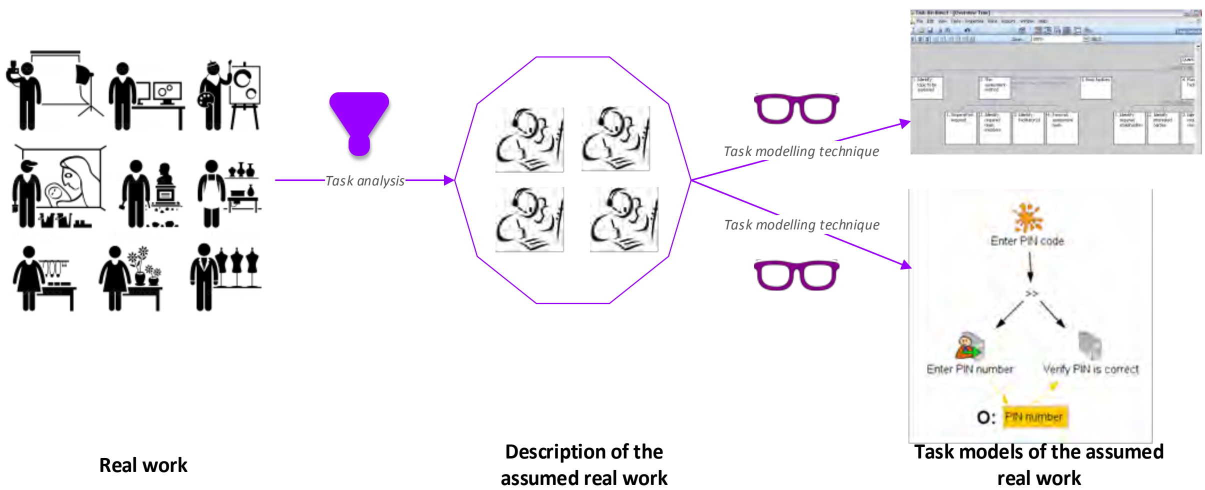


Figure 2: Task analysis and modelling to represent the real work

On one hand, the process of filtering out activities allows to take into consideration the aspects of the real work that are important for the analysis, and, on the other hand, it allows to define the boundaries of analysis. For example, in the presented work we concentrate on human tasks related to the use of the partly autonomous interactive system. Therefore, our task modelling approach focuses on the activities carried out by human interacting with this type of systems in order to understand what are his/her tasks when the system properly works and what are his/her tasks in case of a system automation degradation. Our goal is to understand, through task analysis and modelling, how tasks can be distributed between the human and the system in order to find a good balance for ensuring and improving the usability and the resilience of the system.

2 State of the art on task modelling techniques and associated methods

Task modelling techniques provide support to structure and store information gathered from task analysis through their own characteristics elements of representation such as hierarchical trees, flow charts, etc. The resulting models are able to capture and represent some particular aspects of the description of the assumed real work which are relevant for the analyst who selects the most suitable task modelling technique for her/his needs.

To accomplish tasks, users may need to manipulate objects or information about current situation of the system and its environment, and knowledge about which actions to perform and how to perform them. When designing partly autonomous interactive systems, the phase of tasks analysis usually focuses on: identifying goals which should be reached, grouping activities that have to be accomplished, understanding execution order of these activities and identifying objects required to perform the tasks. Existing analysis techniques and notations do not provide full support for explicitly and distinctively describing concepts related to the notions of object, knowledge and its different types (declarative, situational, procedural and strategic) and information. Although some of them provide support for describing manipulated objects, most of these notations are focused on representation of procedures and methods for reaching a goal rather than the knowledge, information and objects involved.

This section proposes:

- a detailed analysis of a number of established task modelling techniques and associated methods including the description of their main elements of notation and the description of their supporting tools,
- a focus on Human-centered Assessment and Modelling to Support Task Engineering for Resilient Systems version 1.0 (HAMSTERS 1.0) because, thanks to a structured process, it provides support for modelling complex human activities and for representing information and objects. Furthermore, it is inspired from Concur Task Trees (CTT) (Paternò, 2004) and has thus been intended to remain compatible with it (at the user level). In addition, HAMSTERS 1.0 involves notation elements such as conditions associated to task executions, data flow across task models etc. extending its expression power beyond the one of CTT. HAMSTERS is publicly available, featuring a task simulator and providing a dedicated API for observing editing and simulation events[1].

[1] http://www.irit.fr/recherches/ICS/softwares/hamsters/index.html

2.1 An overview of task modelling techniques and associated methods

Task analysis has foundations going back to the end of the last century. One of the reasons of its longevity is that task modelling techniques and methods have been adapted to the evolution of socio-technical systems they are meant to describe. This has given rise to over one hundred methods and techniques that are called task analysis as claimed in the introduction of the "Handbook on task analysis" (Diaper & Stanton, 2004). The scientific community has actively developed and improved the expressive power of task analysis and modelling techniques as put forward in (Caffiau, Scapin, Girard, Baron, & Jambon, 2010). This expressive power increases enables for instance to better represent and analyse human activities while interacting with new technologies as in (Jourde, Laurillau, & Nigay, 2010) or to explicitly represent information used for performing tasks as in (Villaren, Coppin, & Leal, 2012). We provide here after an overview of representative task modelling techniques and their related methods when available. They are presented in chronological order.

Hierarchical Task Analysis (HTA) (Meyer, Annett, & Duncan, 1967) (Anett, 2004) models can be represented textually or graphically with structured numbering corresponding to the various goals and decomposed actions. Plans can then be drawn up indicating the order in which tasks are to be performed. The numbering of goals and sub-goals allows adding sequence information. It is able to describe only procedural knowledge. The modelling activity is supported by Task Architect (Stuart & Penn, 2004).

The Goals, Operators, Methods and Selection rules (GOMS) (Card, Newell, & Moran, 1983) model was developed as a model for predicting human performance while interacting with a system. GOMS reduces a user's interaction with a computer to its elementary actions that can be physical, cognitive or perceptual. Using these elementary actions as a framework, an interface can be studied. There are several support tools, but the most recent one is CAT-HCI (Williams K. , 2005).

Task Knowledge Structure (TKS) (Johnson & Johnson, Knowledge Analysis of Task; Tast Analysis and Specification for Human-Computer Systems, 1989) is based on the assumption that people possess knowledge structures in their memory that relate to tasks. The approach represents declarative and procedural knowledge to describe the roles, goals, plans and procedures composed of actions and objects. Even though knowledge is part of the acronym, it is not explicitly represented in models which can be edited using the ADEPT tool (Johnson, Wilson, Markopoulos, & Pycock, 1993).

Méthode Analytique de Description de tâches (MAD) (Scapin & Pierret-Golbreich, 1989) combines structured interviewing and modelling techniques focusing explicitly on the hierarchical relation of tasks. The main concepts of MAD include tasks, actions and structures. In last versions, it can also describe objects explicitly (and even in a formal way) and conditions related to tasks. The environment K-MADe supports this method (Scapin D. L., 2007).

User Action Notation (UAN) (Hix & Harston, 1993) was designed to formalize the communication between the User Interface (UI) designers and the development team when specifying and analysing details of technology. The central concept is the abstraction of task, including hierarchical structure, and sequencing. User actions, interface feedback and state changes are used to build up the task description, which may subsequently be used as an action at higher levels of abstraction. This rating primarily provides support to the formal modelling of interactive tasks. Even though very popular in the 90's the close connection between tasks and user interfaces made it very difficult to cope with large systems.

Groupware Task Analysis (GTA) (Van Der Veer, Lenting, & Bergevoet, 1996) was developed for the modelling of complex tasks in a co-operative environment. The foundations of the GTA approach are a combination of both ethnography and activity theory adopting a clear distinction between tasks and actions. Thus, it introduces the concept of role to the person in charge of a task. It describes only procedural knowledge. Euterpe is the support tool for modelling activity (van Welie, van der Veer, & Eliëns, 1998).

Diane+ (Tarby & Barthet, 1996) models a task with three concepts: operation, sequencing and decomposition. DIANE+ like MAD employs a rich graphical notation to represent the decomposition of tasks as well as temporal and logical relationships between the tasks. It is supported by several tools, such as TAMOT (Paris, Tarby, & Vander Linden, A Flexible Environment for Building Task Models, 2001) and Isolde (Paris, Vander Linden, & Lu, 1992) but there is no user's knowledge representation.

Visual Task Modelling Language (VTML) (Brown & Leveson, 1998) is a graphical notation designed in order to analyse formally the actions of a human operator in relation to safety issues. It proposes to record the actions of the operators in the form of flow charts. There is no indication on knowledge representation. VTML is supported by SpecTRM (Lee, Howard, & Anderson, 2002).

Cognitive Work Analysis (CWA) (Vicente, 1999) is a work-centered conceptual framework to guide the design of technology for use in the work place. It consists of 5 phases with a dedicated approach for each one: 1) Work Domain Analysis (WDA) with Abstraction and Decomposition table, 2) Control Task Analysis (CTA) supported by decision ladder, 3) strategies analysis with information flow maps, 4) social organization and cooperation analysis in which you can use all the above approaches, 5) worker competencies analysis based on Skill, Rules, Knowledge taxonomy (Rasmussen, 1983) (Rasmussen, 1985). CWA is not supported by a tool.

Task Analysis for Knowledge Descriptions (TAKD) (Diaper, 2001) is a method which is first used to generate descriptions of tasks, and then to re-express the descriptions in terms of knowledge. The resulting knowledge descriptions consist of action/object pairs that when combined represent the knowledge content of tasks. This method is related to TKS. It is a summary representation of the different types of knowledge that are recruited and used in task behaviour (Johnson, Johnson, Waddington, & Shouls, 1989), and it is supported by LUTAKD (Diaper, 2001).

Concurrent Task Tree (CTT) (Paternò, 2004) a graphical notation with a dedicated support tool, CTT Environment (CTTe) (Paternò, Mancini, & Meniconi, 1997) have enabled the distinction of abstract, user, interaction and application tasks as well as the possibility to model temporal aspects of activities when specifying a model. A set of operators, mainly taken from the LOTOS formalism (ISO, 1989) is used to indicate the temporal relationships among tasks such as iteration, sequentially, concurrency, disabling, and recursion. It was one of the first notation embedding in procedural knowledge concurrency and interruptions that are however clear characteristics of human procedural knowledge. It provides support for the description of cooperative tasks and user role. Scheduling temporal operators apply only between tasks and how to decompose tasks can lead to ambiguities in the model interpretation.

AMBOSS (Giese, Mistrzik, Pfau, Szwillus, & von Detten, 2008) task modelling environment takes into account the special needs for safety-critical socio-technical systems. The authors do not provide an accurate description of the underlying notation; however it seems to use the same concepts of hierarchical structure and temporal MAD. AMBOSS can also combine information related to human error and safety in task models (barriers, risk factor, and criticality of the task). Furthermore, it takes into account the concept of role. However, concepts of declarative knowledge are not handled.

Situation Awareness Modelling & ANalysis for Transition Amelioration (SAMANTA) (Villaren, Coppin, & Leal, 2012) is a methodology for design of Human-Computer Interfaces in dynamic systems taking into consideration the situation elements constituting operators' activity. It consists of 4 phases: 1) task modelling based on CTT notation (Paternò, 1999) and situation modelling phase based on DSA (Stanton, et al., 2006), 2) Task/Situation elements Association in which each task of the task model is associated with a set of Situational Awareness (SA) elements from the situation model, 3) Transition Analysis, tasks are compared two by two in order to categorize the transitions that link them, 4) Impact on HCI for interface validation integrated with design experts recommendations. It is the only notation in which there is a tentative to propose the production of a specific model of knowledge by means of a model called context. It thus identifies clearly knowledge required for performing the tasks but does not separate it into several categories.

2.2 Focus on the Human-centered Assessment and Modelling to Support Task Engineering for Resilient Systems version 1.0 (HAMSTERS 1.0)

Human-centered Assessment and Modelling to Support Task Engineering for Resilient Systems version 0.1 (HAMSTERS 1.0) (Martinie, Palanque, & Winckler, 2011) is a notation to describe the decomposition of human goals into activities. It is inspired from Concur Task Trees (CTT) (Paternò, 2004) and has thus been intended to remain compatible (at the users level) with it. Indeed both can be considered as hierarchical and graphical models representation relationship between tasks by means of operators. However, HAMSTERS 1.0 involves extensions such as conditions associated to task executions, data flow across task models etc. extending its expression power beyond the one of CTT. Additionally, it is publicly available, featuring a task simulator and providing a dedicated API for observing editing and simulation events[2].

It is a tool-supported graphical task modelling notation aiming at representing human activities in a hierarchical and ordered way. Goals can be decomposed into sub-goals, which can in turn be decomposed into activities, and output of this decomposition is a graphical tree of nodes. Nodes can be tasks or temporal operators.

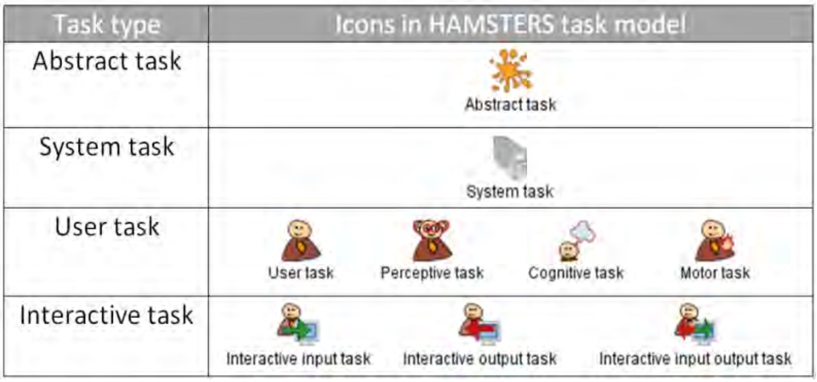


Figure 3: High-level task types in HAMSTERS 1.0

[2] http://www.irit.fr/recherches/ICS/softwares/hamsters/index.html

As illustrated in Figure 3, tasks can be of several types and contain information such as a name, information details, critical level and so on. Due to space constraints, only the high-level task type is presented but they are further refined (for instance the cognitive tasks can be refined in Analysis and Decision tasks (Martinie, Palanque, & Winckler, 2011)). For the procedural description concepts are very close to the ones of CTT such as the task types and the operators.

As detailed in Table 1, there are several operators type listed in the first column. Each operator type is represented by a symbol which describes a temporal relationship between sub-goals and between activities. For example, the operator type "Enable" is represented by the symbol ">>". It means that the temporal relationship between T1 and T2 can be expressed as "T1>>T2" that is T2 is executed after T1.

TABLE 1: TEMPORAL ORDERING OPERATORS IN HAMSTERS 1.0

Operator type	Symbol	Description
Enable	T1>>T2	T2 is executed after T1
Concurrent	T1|||T2	T1 and T2 are executed at the same time
Choice	T1[]T2	T1 is executed OR T2 is executed
Disable	T1[>T2	Execution of T2 interrupts the execution of T1
Suspend-resume	T1|>T2	Execution of T2 interrupts the execution of T1, T1 execution is resumed after T2
Order Independent	T1|=|T2	T1 is executed then T2 OR T2 is executed then T1

Tasks can also be tagged by temporal properties to indicate whether or not they are iterative, optional or both. As explained above; composition and structuration mechanisms have been introduced in order to provide support for description of large amounts of activities (Martinie, Palanque, & Winckler, 2011). One main element of these mechanisms is subroutine. A subroutine is a group of activities that a user performs several times possibly in different contexts and which might exhibit different types of information flows. A subroutine can be represented as a task model and a task model can use a subroutine to refer to a set of activities. This element of notation enables the distribution of large amount of tasks across different task models and factorization of the number of tasks.

HAMSTERS 1.0 also provides support for representing how particular objects (data, information...) are related to particular tasks.

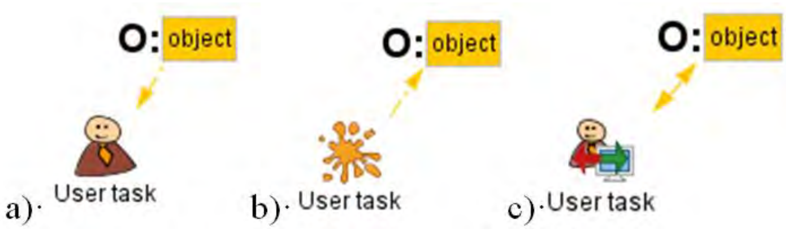


Figure 4: Relationships between tasks and objects in HAMSTERS 1.0

The previous image illustrates the three relationships (input, output or both) between objects and tasks that can be expressed with HAMSTERS 1.0 notation. Object (data, information…) can be needed as an input to accomplish a particular task (as illustrated in Figure 4a) by the incoming arrow. Particular tasks may generate an object or modify it (as illustrated in Figure 4b). The values of these objects can be used as preconditions for the tasks (as shown in Figure 5). Indeed, to accomplish the abstract goal "Enter PIN code" is required that the user "enter PIN number" and then the system "verify PIN is correct". When the user "Enter PIN number" this generates an object "PIN Number" which is used by the system in "Verify Pin is correct".

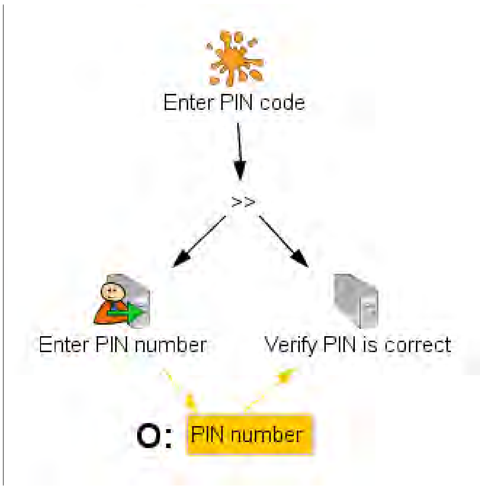


Figure 5: Relationship with objects in HAMSTERS 1.0

HAMSTERS 1.0 thanks to its hierarchical decomposition and subroutine mechanisms can be used to describe procedural knowledge. Additionally, thanks to the explicit representation of objects HAMSTERS 1.0 can be integrated in the task model by means of preconditions on tasks.

3 Identify requirements for existing task modelling techniques and associated methods: what is still missing

The description of existing task modelling techniques and methods has demonstrated that there is limited support in existing notations for explicitly representing the various types of information manipulated by the users. In widely used notations such as HTA or CTT, information or objects manipulated by the users while performing the tasks does not receive a similar treatment as the sequencing of tasks. Moreover, the presented techniques offer a limited support for integrating all these data (objects and information) into the usual procedural-centred description of tasks. This procedural-centred description offers a partial view about just one type of knowledge (de Jong & Ferguson-Hessler, 1996), the procedural knowledge, that is required to perform a task. As a consequence, task models produced using these techniques only capture partial information about the tasks. This may lead to an increase of the design effort (for example, an increased number of prototyping-evaluation cycles to be able to deliver a usable application).

The latest task modelling techniques, such as SAMANTHA, confirm the interest of the scientific community in integrating the representation of different types of knowledge in task models. To be able to integrate different types of knowledge, an extension of the current task modelling notations and techniques is required.

According to the cognitive psychology, there are two main types of knowledge: declarative and procedural (Anderson, et al., 1995). Declarative knowledge is factual knowledge what is true or false.

It describes objects by specifying the properties which characterize them. This type of knowledge does not pay attention to the actions needed to obtain a result, but only to their properties (Turban & Aronson, 1988). Alternatively, procedural knowledge corresponds to the knowledge exercised in the performance a task. It is directly applied to a task and is acquired while performing activities (Koedinger & Corbett, 2006). Declarative knowledge corresponds to the "I know that" while procedural knowledge corresponds to "I know how". They have different modes of mental representation and involve different memory processes and brain areas (Berge & Van Hezewijk, 1999). Beyond this declarative/procedural distinction, there are other two types of knowledge: situational and strategic (de Jong & Ferguson-Hessler, 1996). Situational knowledge is related to case-based reasoning and contains domain specific information, while strategic knowledge is associated to a plan, a parallel checking and an analysis of possible choices. They have to be considered as refinements of the two main types (declarative and procedural). Each type of knowledge has its own distinctive mental representation and organization.

With regard to the declarative knowledge is represented by mental images and/or symbolic forms such as words (Sternberg, 2008). The image is analogue to the object that it represents (e.g. a physical object or an information in the real world) and is made up of concrete attributes and features that can be concurrently captured. Conversely, the word is a symbolic representation and the relationship between this symbolic form and the object is arbitrary.

A concept provides a mean for understanding the world and is a mental organization of knowledge. It is the fundamental unit of declarative knowledge, and often a single concept may be captured in a single word. Concepts have different characteristics which ensure a wide flexibility in using them: such as (a) concepts can contain other concepts, (b) concepts can include typical and generic events, (c) concepts can present different abstraction degrees and (d) they can comprehend information about relationships (Komatsu, 1992).

Semantic relations between concepts are represented through a semantic network that is as a web of interconnected elements made up of nodes and labels (Collins & Quillian, 1969). Nodes represent concepts and labels denote relationships which may involve category membership, attributes, or other semantic relations. These labels connect concepts in memory for allowing persons to link different nodes through their meaning. Semantic networks are given different names according to their scope and use. For instance, they are called associative networks when focusing on links and concept maps when focusing on concepts. Concept maps are also used as a direct method for assessing knowledge (Shavelson, Ruiz-Primo, & Wiley, 2005) and can be used for representing the transition from novice to expert performance (Royer, Cisero, & Carlo, 1993) (Williams C. G., 1998). As claimed by Sowa (Sowa, 1992) all these semantic networks share a graphical representation that can be used either to represent knowledge or to support automated systems for reasoning about the represented knowledge.

Concept maps are graphical representations of knowledge that are comprised of concepts and the relationships between them. We define a concept as a perceived regularity in events or objects, or a record of events or objects, designated by a label. Concepts are usually enclosed in circles or boxes, and relationships between concepts are indicated by connecting lines that link them together. Sometimes these are called semantic units, or units of meaning (Coffey, et al., 2003). Concept maps are aimed at creating a knowledge model (Cañas, Hill, & Lott, 2004) that is a set of concept maps and associated resources about a particular domain of knowledge (Novak & Cañas, 2008).

The structure of a concept map is dependent on its context. Consequently, maps having similar concepts can vary from one context to another and are highly idiosyncratic. The strength of concept maps lies in their ability to measure a particular person's knowledge about a given topic in a specific context. Therefore, concept maps constructed by different persons on the same topic are necessarily different, as each represents its creator's personal knowledge (as for modelling activity where the model represents the analyst point of view and her/his interest as illustrated in Figure 2). Similarly, we cannot refer to the correct concept map about a particular topic, as there can be many different representations of the topic that are correct. There are no restrictions on what words can be used to form concepts. Usually they tend to be nouns and consist of as few words as possible (Cañas, et al., 2005).

Task modelling notations should provide support to explicitly represent different types of knowledge (declarative, situational, procedural and strategic), as well as objects (manipulated by the user) and information. Therefore, to properly represent declarative knowledge through task modelling techniques and methods there is the need to a set of requirements (REQ_TM) derived from the presented cognitive psychology excursus:

1) REQ_TM_1: Concepts should be used to represent explicitly declarative knowledge.
2) REQ_TM_2: Concepts can be refined into objects and into information and these two entities should be made explicit.
3) REQ_TM_3: Semantic networks may be used to represent the concepts and their relationships.
4) REQ_TM_4: As concepts relate to each other, these connections should be made explicit to structure and group declarative knowledge.
5) REQ_TM_5: Declarative knowledge can be refined into strategic and situational and these two types of knowledge should be represented explicitly.

As for the declarative also the procedural knowledge has its own distinctive mental representation and organization from which we derive the related requirements. Mental representation of procedural knowledge has always been considered hard to identify and of course to prove and most of the results currently available derive from computer simulations. As claimed by Schank and Abelson (Schank & Abelson, 1995) such knowledge could be represented as scripts which is a structure that describes sequences of events in a particular context. The structure is an interconnected net made of slots influencing each other.

Procedural knowledge can be structured as a set of rules governing a production which is the entire set of rules to perform a task or use a skill (Young, 2001). These rules are organized into routines that are sets of instructions for accomplishing a task. They can be refined in subroutines that are sets of instructions to accomplish a subtask. These routines are typically organized according to basic control structures in computer science such as sequence, alternatives and iteration (Meyer, Annett, & Duncan, 1967).

Due to the structuring in routines and subroutines it is possible to adopt a hierarchical representation to describe them (Sowa, 1992). In such hierarchical networks information known at a higher level in the hierarchy is known at lower levels.

Therefore, to properly represent procedural knowledge through task modelling techniques and methods there is the need to a set of requirements (REQ) derived from the presented cognitive psychology excursus:

6) REQ_TM_6: Routines and subroutines should be used to represent explicitly procedural knowledge.

7) REQ_TM_7: As routines and subroutines are related to each other, these relationships should be made explicit in order to structure and group procedural knowledge.

8) REQ_TM_8: Hierarchical structures (trees) may be used to represent both routines and sub-routines and their relationships.

9) REQ_TM_9: Procedural knowledge can be refined into strategic and situational and these two types of knowledge should be represented explicitly.

Even though we presented declarative and procedural knowledge in an independent way, both knowledge types corresponds to different aspects of the same concept. Indeed, cognitive psychology research contributions aiming at proposing integrative models for representing declarative and procedural knowledge such as the Adaptive Character of Thought—Rational (ACT-R) and the Parallel Distributed Processing (PDP). The former integrates a network representation for declarative knowledge and a production system representation for procedural knowledge. On one hand, declarative network includes storing and retrieving of information (called compilation process) and, on the other hand, procedural system implements the processes of automation or proceduralization (Anderson, et al., 1995). According to ACT-R (Anderson, Matessa, & Douglass, 1995), all knowledge begins as declarative information. Procedural knowledge is learned by making inferences from already existing factual knowledge. New production rules are formed by the conjunction or disjunction of existing production rules. This demonstrates the importance of dealing with declarative knowledge in an explicit and exhaustive way. The latter suggests that humans are able to carry out a large number of cognitive operations simultaneously by the use of countless numbers of neural processors located in the brain (Rumelhart, Hinton, & McClelland, 1987). In neural networks the configurations, also called patterns, carry more information than single units generating knowledge representation.

The integrative approaches suggest that declarative and procedural knowledge can be structured, grouped and connected together as they are closely intertwined while users perform tasks. These provide another requirement needed for knowledge modelling representation:

10) REQ_TM_10: All the information above should be integrated in a single model as they are closely intertwined while users perform tasks.

11) REQ_TM_11: These task modelling techniques and methods should allow the integration with other methods and modelling techniques ensuring compatibility at model and tool level

12) REQ_TM_12: These task modelling techniques and methods should be supported by a tool in order to deal with the amount of data and to ensure data analysis

It is important to note that this section has focussed on the data manipulated by users when performing tasks either being of knowledge, information or objects nature. Cognitive process manipulating this data (analysis and decision) or detailed interaction with the real world (motor and perception) have not be detailed in this related work as they are already rather extensively covered by notations for task models.

The identified requirements can guarantee a support for representing in a complete and systematic way information, objects and knowledge users need when accomplishing tasks while using partly autonomous interactive systems.

4 Comparison of task modelling techniques and associated methods according to the identified requirements

In this chapter we have offered an overview of the exiting task modelling techniques and methods. Moreover, we have identified a set of requirements needed for representing in a complete and systematic way information, objects and knowledge users in task modelling techniques and methods. In this section we make a comparison of task modelling techniques and methods according to the identified requirements.

Indeed, the following table which summarizes for each main type of knowledge (declarative and procedural, column 1), the requirements which have been expressed above (column 2). As the tools associated to the notation sometimes cover only part of the notation and sometimes go beyond the notation (with the help of simulation or analysis for instance), last row to "Tool support" for each task modelling notation. For all requirements, there are six possible values:

- "Yes" (green box with YES) means that the task modelling technique satisfies the corresponding requirement,
- "No" (white box with NO) means that the task modelling technique does not satisfy the requirement,
- "Partially" (grey box) indicates that the task modelling technique partially satisfies the requirements i.e. it does not deal with knowledge and information,
- "Partially with objects" (grey box with P-O) indicates that the task modelling technique partially satisfies the requirements i.e. it does not deal with knowledge and information,
- "Through precondition" (grey box with T-P) indicate that the task modelling technique has attempted to integrate all the requirements via precondition,
- "Through constraint" (grey box with T-C) indicates that the task modelling technique has attempted to integrate all the requirements via constraint.

Table 2: Comparison of task modelling notations and associated methods according to suggested requirements (taken and adapted from (Martinie, Palanque, & Ragosta, 2013))

	Requirements /Notations	HTA	GOMS	TKS	MAD	UAN	GTA	Diane+	VTML	CWA	TAKD	CTT	AMBOSS	SAMANTA	HAMSTERS 1.0
Declarative Knowledge (DK)	**REQ_TM_1.** Concepts used to represent DK	No	No	Yes	No	No	No	No	No	Yes		No	No	Yes	No
	REQ_TM_2. Refinement of concepts into objects and information	No	No	No	P-O (explicit)	No	No	No	No	No	P-O	P-O	No	P-O	P-O
	REQ_TM_3. Concepts connections should be made explicit	No	No	Yes	No	No	No	No	No	Yes	No	No	No	Yes	No
	REQ_TM_4. Semantic networks to represent the concepts and their relationships	No	No	No	No	No	No	No	No	No	No	No	No	Yes	No
	REQ_TM_5. Refinement into strategic and situational knowledge	No	No	No	No	No	No	No	No	No	No	No	No	No	No
Procedural Knowledge (PK)	**REQ_TM_6.** Routines and subroutines should be used to represent explicitly PK	No	No	No	No	No	No	No	No	No	No	No	No	No	Yes
	REQ_TM_7. Routines and subroutines relationships should be made explicit in order to structure and group PK	No	No	No	No	No	No	No	No	No	No	No	No	No	Yes
	REQ_TM_8. Hierarchical structures to represent routines, sub-routines and their interrelationships	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	No	No	Yes	Yes	Yes	Yes
	REQ_TM_9. Refinement into strategic and situational knowledge	No	No	No	No	No	No	No	No	No	No	No	No	No	No
Integration	**REQ_TM_10.** All the information above should be integrated in a single model	No	TP	TP	TP	No	TP	TP	TP	TC	TP	TP	TP	TP	TP
	REQ_TM_11. Integration with other methods and modelling techniques ensuring compatibility at model and tool level	No	No		No	No	No	No	No	No			No		Yes
Tool support	**REQ_TM_12.** Supported by a tool	Task-Architect	CAT-HCI and others	ADEPT	K-MADe	Quantum	GTA and EUTERPE	TAMOT and Isolde	SpecTRM tool Suite	(No tool support)	LUTAKD	CTTe	AMBOSS	SAMANTA	HAMSTERS 1.0

Yes	The task modelling technique satisfies the requirement
No	The task modelling technique doesn't satisfy the requirement
	The task modelling technique partially satisfies the requirement (with P-O via object; with T-P via precondition; with T-C via constraint)

As shown in the table, most of these techniques are focused on procedural representation of knowledge than knowledge per sè. To overcome the identified issues, we have selected a particular task modelling technique, HAMSTERS, because, despite it does not present all the identified requirements, it provides support for modelling complex human activities and for representing information and objects thanks to a structured process. Furthermore, it involves notation elements such as conditions associated to task executions, data flow across task models etc. extending its expression power beyond the one of CTT, HTA, MAD... HAMSTERS is publicly available, featuring a task simulator and providing a dedicated API for observing editing and simulation events[3]. Moreover, this PhD thesis focuses on human activities which are required to interact with a partly autonomous interactive system, and HAMSTERS offer the opportunity to be integrated with ICO (a notation for the description of interactive systems). This integration is done at the model level, thanks to the iterative construction of system and task models, as well as at the tool level exploiting PetShop environment for the system side and HAMSTERS for the task side.

In addition, as one of the contributions of this PhD thesis, Chapter 4 presents the extensions to HAMSTERS 2.0 task modelling technique in order to allow the representation of all the elements of declarative knowledge and their integration within the procedural knowledge description of HAMSTERS 1.0.

5 Conclusion, lessons learnt and way forward

In this chapter we have provided an overview of the current state of the art in task analysis and modelling to support the analysis and assessment of usability and resilience of partly autonomous interactive systems. The description of existing task modelling techniques and methods has demonstrated the limited support for explicitly representing the various types of information, objects manipulated by the users and different types of knowledge required to interact with a partly autonomous interactive system.

The highlighted limitations are a justified need for improving task modelling techniques and methods for which many other considerations much be taken into account to ensure efficient and successful human computer interaction particularly for partly autonomous interactive systems. In order to overcome these limitations, a set of requirements has been proposed.

The identified requirements can guarantee a support for representing in a complete and systematic way information, objects and knowledge users need when accomplishing tasks while using partly autonomous interactive systems.

According to these requirements, a comparison of the exiting task modelling techniques and methods has been made. This has highlighted that most of these techniques are focused on procedural representation of knowledge than knowledge per sè. Moreover, the comparison allowed us in selecting one of the presented task modelling techniques, HAMSTERS 1.0, because, despite it does not satisfy all the identified requirements, it is promising as a basis due to several reasons. This task modelling technique can:

- support the representation of human activities in a hierarchical and ordered way. Goals can be decomposed into sub-goals, which can in turn be decomposed into activities, and output of this decomposition is a graphical tree of nodes. Nodes can be tasks (which can be of several

[3] http://www.irit.fr/recherches/ICS/softwares/hamsters/index.html

types and contain information) or temporal operators. Thanks to those, it is possible to describe temporal relationships between sub-goals and between activities,

- offer the opportunity to be integrated with ICO (a notation for the description of interactive systems), both at model and also a tool level, according to the focus of this PhD thesis on human activities needed to interact with a partly autonomous interactive system,
- be extended.

Indeed, as one of the contributions of this PhD thesis, Chapter 4 presents the extensions to HAMSTERS 2.0 task modelling technique in order to allow the representation of all the elements of declarative knowledge and their integration within the procedural knowledge description of HAMSTERS 1.0.

The extended version of HAMSTERS task modelling technique and its integration with ICO, it will be a starting point for developing a multi-models based for the modelling and the analysis of usable and resilient partly autonomous interactive systems as a whole. This multi-models based approach is the second PhD thesis contribution and it is presented in Chapter 5.

Chapter 2 – Addressing partly autonomous interactive systems and user interactions through modelling techniques and associated methods

This chapter describes the current state of the art of system modelling techniques and methods used to address partly autonomous interactive systems and user interactions. The scope of this chapter is to:

- Define and characterise a partly autonomous interactive system. Moreover, it explains why system analysis and modelling are needed to analyse and assess usability and resilience of partly autonomous interactive systems
- Describe and compare the existing system modelling techniques and methods
- Examine what is still missing in existing system modelling techniques and methods to provide support for the modelling and the analysis of usable and resilient partly autonomous interactive systems. Moreover, it proposes a set of requirements for the analysis and modelling of usable and resilient partly autonomous interactive systems
- Make a comparison of system modelling techniques and methods according to the identified requirements.

The first section offers a description in terms of definition and characteristics of a partly autonomous interactive system. Additionally, it explains why system analysis and modelling are needed to analyse and assess usability and resilience of partly autonomous interactive systems

The second section offers a state of the art on system modelling techniques and methods aiming at describing and comparing the existing ones. It also includes a more detailed description of the ICO system modelling techniques and its associated CASE tool PetShop. This tool supported technique is currently used in ATM domain to investigate issues about human computer interactions.

The third section highlights what is still missing in existing system modelling techniques and methods and offers a set of requirements for addressing partly autonomous interactive systems and users' interactions.

The fourth section offers a comparison of system modelling techniques and methods according to the identified requirements. Moreover, it highlights a synergistic use of task and system modelling techniques for the analysis and modelling of usable and resilient partly autonomous interactive systems.

1 System analysis and modelling to support analysis and assessment of usability and resilience of partly autonomous interactive systems

Automation comes in many different forms and there is a wide range of system characteristics and capabilities that may be classified as automation. Automation can be thought of as the process and the result of allocating activities to a machine or system to perform which have been performed by humans in the recent past. Thus, the widely accepted Wickens' et al. (Wickens, Mavor, Parasuraman,

& McGee, 1998) definition of automation is used, in which automation is referred to as "a device or system that accomplishes (partially or fully) a function that was previously carried out (partially or fully) by a human operator". Note that in some cases, the introduction of automation does not only imply partial or full replacement of the former activity of a human operator. Alternatively the automation, as a new team player, can be in charge of completely new tasks (e.g. such as in the case of airborne automation performing self-separation manoeuvres).

The level of automation taxonomy is useful to analyse and to compare automation by understanding its nature in detail. In principle, automation is not either "all or nothing" and in this context, ten different levels of automation may be identified, as illustrated in Table 3. These levels help to identify in which degree a task is automated, from a low level (i.e. no system assistance) to a high level (i.e. completely automated system).

TABLE 3: LOA IN MAN-COMPUTER DECISION-MAKING (TAKEN FROM (SHERIDAN & VERPLANK, HUMAN AND COMPUTER CONTROL OF UNDERSEA TELEOPERATORS, 1978))

HIGH	10. The computer decides everything, acts autonomously, ignoring the human.
	9. Informs the human only if it, the computer, decides to
	8. informs the human only if asked, or
	7. executes automatically, then necessarily informs the human, and
	6. allows the human a restricted time to veto before automatic execution, or
	5. executes that suggestion if the human approves, or
	4. suggests one alternative
	3. narrows the selection down to a few, or
	2. the computer offers a complete set of decision/action alternatives, or
LOW	1. the computer offers no assistance: human must take all decisions and actions

Tasks usually performed by human operators, may be partially or fully automated. The automation may become in charge of either all or parts of the tasks previously attributed to humans or part of new tasks (that were previously not performed by the human). The automation level determines the type human and automation cooperate. This cooperation requires understanding how to balance automation and interactivity and specify how a task can be performed by assigning the generic functions to the human and the machine operator in terms of function allocation. "Function allocation cannot be based on a consideration of the tasks only, but must consider the total equilibrium of a work situation—corresponding to a notion of balanced work. The concept of equilibrium emphasises the fact that a change in function allocation disturbs the established equilibrium. This will have consequences for the system as a whole, and one result may be that a new equilibrium is established which differs significantly from the previous one." (Bye, Hollnagel, & Brendeford, 1999, p. 292)

As illustrated in the following picture, the number of functions can vary according to the levels of automation. In case of low level of automation, there are few functions which can be carried out by the user "manually". When the level of automation grows up, the number of functions increases but, to be carried out, the user can be supported by automation. This support can be described in terms of re-allocation or migration of functions from user side to system side (which is represented by the top of the distribution curve in Figure 6). When the level of automation becomes higher, the number of functions, which can be carried out by the user, decreases because the system can execute most of them "autonomously".

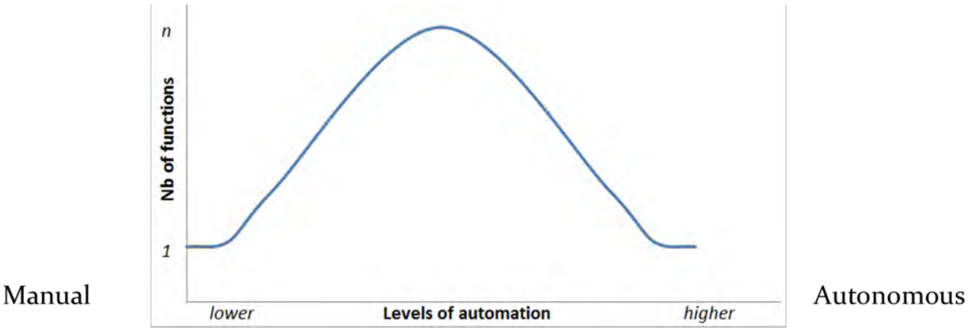


Figure 6: Distribution curve of number of functions related to levels of automation

This relationship between the number of functions and the levels of automation is crucial when we are dealing with partly autonomous interactive systems. For example, one of the tasks, which is continuously performed by pilots, is to check weather conditions. This task implies, as first activity, to check if the weather radar is properly works. Test weather radar application is a repetitive task which does not improve the pilots' situational awareness but it just requires time and a number of manual functions to be accomplished (Harrison, Johnson, & Wright, 2002). According to the distribution curve, this case fits in the lower levels of automation and several manual functions should be carried out by the pilot (as illustrated in Figure 7). Increasing the level of automation, the weather radar can execute automatically this test. This migration of functions reduces the number of functions previously performed by the pilot.

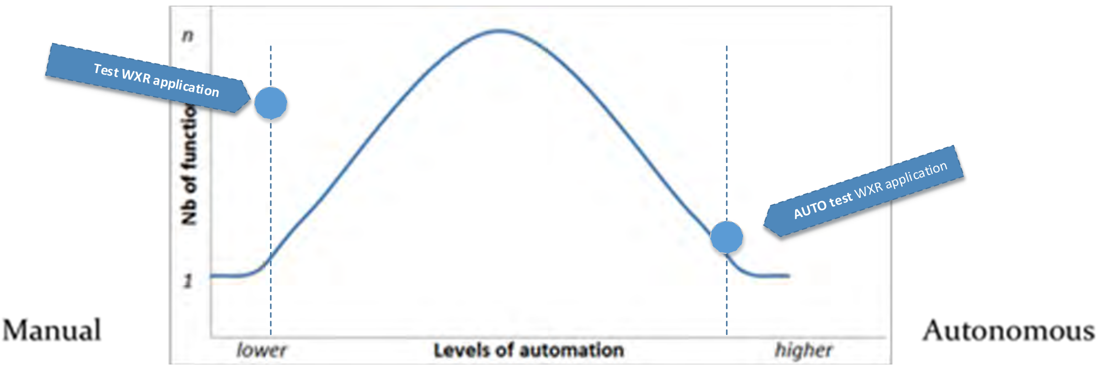


Figure 7: Distribution curve of test and auto test of weather radar application

To achieve a good balance between the functions allocated on human and on system, it is important to assess tasks complexity through task analysis and using task modelling techniques and methods to represent the human activities (as explained in Chapter 1). At the same time, it is important to assess system complexity through system analysis and using system modelling techniques and methods to represent the system and the user interactions. Both modelling techniques should be able to represent this migration of functions in terms of an evolution of corresponding models.

According to this introduction, we can define a system in terms of levels of automation and functions allocation. Moreover, it should be kept in mind that the description of each automation level follows the reasoning that automation is addressed in relation to human performance, i.e. the automation being analysed is not just a technical improvement but has an impact on how the human is supported in his/her task accomplishment. So in this PhD thesis, we consider the system under analysis also interactive.

An interactive system is a reactive computerised system which takes into account, in the execution, the information provided by the users or the system, and produces, during the execution time, an observable representation of its internal state. Figure 8 presents the general architecture of an interactive system and helps to show the main components of this type of system. The user interacts with the system via an interface consists of devices (hardware, electronics), device drivers (software), and a dialogue controller (software). The dialogue controller manages scheduling of the input / output operations and consistency. This concept is based on the models of Seeheim and Arch/Slinky (Bass, et al., 1991).

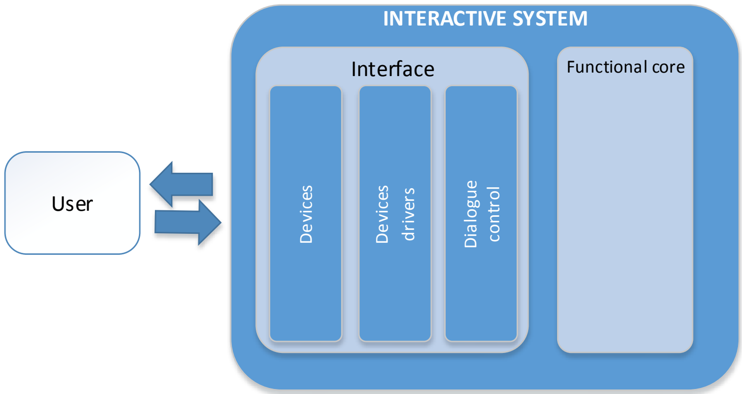


Figure 8: General architecture of an interactive system (adapted and taken from (Martinie, 2011))

This architecture representation of an interactive system can highlight the various physical, electronic and software components that should be taken into account during the design and development of this type of system (Martinie, 2011).

As explained in Chapter 1 with regard to task analysis and task modelling techniques, we can assume the same reasoning with regards to system analysis and system modelling techniques. In Figure 9, the complexity of real systems is filtered out through a yellow filter that represents a system analysis. The aim of this filtering out is to obtain a description of real systems focused on all that information belongs to the system analysis. Once obtained this description, the analyst can choose between several system modelling techniques and methods to create a system model.

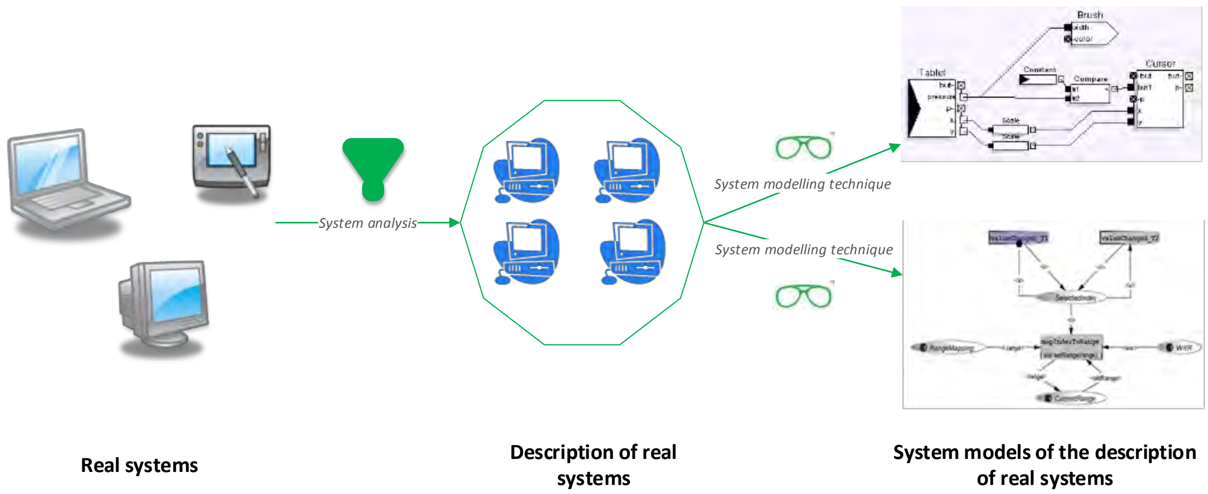


Figure 9: System analysis and modelling to represent real systems

Each system modelling technique and associated method can provide a different way to represent the system. As illustrated in Figure 9, there are 2 green glasses for representing different system modelling techniques and their specific notations as Input CONfigurator (ICON) (Dragicevic & Fekete, 2001) in the upper part and ICO (Navarre, Palanque, & Bastide, 2001) in the lower part. Depending on the choice of notation, the system modelling operation will result in a different model of the system since each type of model highlights particular aspects of the system (Section 2.1 presents a state of the art on existing notations). For this reason it is important to choose the most suitable system modelling technique. Thanks to the selected system modelling technique and associated method, the analyst can highlight the aspects that are relevant to the goals of his/her analysis.

In the current work we concentrate on partly autonomous interactive system and how users interact with them. Therefore, our system modelling approach focuses on partly autonomous interactive systems in order to understand how tasks can be allocated between human and system and also what will be the impact of an automation degradation of the system on this allocation between human tasks and system functions. Our goal is to understand, through system analysis and modelling, how tasks can be distributed between the human and the system to find a good balance for ensuring and improving the usability and the resilience.

2 State of the art on system modelling techniques and associated methods

A number of formal notations have been developed to assist the analysis and modelling of partly autonomous interactive systems. This section offers an overview of these notations. It is therefore concerned with the extent to which these notations support the usability and the resilience of partly autonomous interactive systems. The notations considered on the one hand describe the capabilities and resources of users in relation to a specific system, and on the other, those aspects of an interactive system that must be analysed from a user perspective in order to ensure and improve those characteristics. This section proposes:

- a detailed analysis of a number of established system modelling techniques and associated methods including the description of their main elements of notation and the description of their supporting tools.
- a focus on the Interactive Cooperative Objects (ICOs) and its associated CASE tool PetShop because it enables to describe, in a formal way, all the components of highly interactive applications, the structural or static aspects of systems, and uses high-level Petri nets (Genrich, 1991) to describe their dynamic or behavioural aspects. An ICO specification fully describes the potential interactions that users may have with the application. Furthermore, it has a genuine capability of being connected to task models (please refer to Section 4).

2.1 An overview of exiting system modelling techniques and associated methods

Since the late sixties a lot of work has been devoted to the definition of notations and tools to support interactive systems analysis and modelling. As the complexity of interactive application was growing, the expressive power of the underlying formalisms has also been increased. They are presented in chronological order.

ATN (Parnas, 1969) in which the user interacts with a sort of terminal connected to a system. At any point in time that terminal is some specific terminal state which is characterised by the set of possible input messages and their interpretation. The author proposes a model with finite states of system where, although a state has a consistent response to a given input, the number of states would be so large as to make analysis of the system impossible for all practical purposes (Haugeneder & Gehrke, 1986)

Automata (Jacob R. J., A specification language for direct manipulation user interfaces, 1986) focus on direct manipulation user interfaces which involve a set of visual representations on a screen and a standard repertoire of manipulations that can be performed on any of them. This means that the user has no command language to remember beyond the standard set of manipulations, few changes of mode, and continuous reminder on the display of the available objects and their states. Direct manipulations represent a powerful paradigm for designing user interfaces, but such interfaces have been difficult to specify and program conveniently. Jacob et al. (Jacob, Deligiannidis, & Morrison, 1999) have proposed a new version of their description technique in order to provide support for describing non WIMP interface.

Tools foR an Interactive Development EnvironmeNT (TRIDENT) (Vanderdonckt & Bodart, 1993) consists of a methodology and support environment for developing highly interactive business oriented applications. This supports a continuum from task analysis to a first prototype without disruption. The main steps involve a task analysis performed as a hierarchical decomposition of the interactive task into a sub-task, a specification of the functional requirements and its integration with task analysis results, a writing of an activity chaining graph which graphically depicts information and function flow within the task, the selection of an interaction style, the definition of presentation units, the selection of abstract interaction objects, their transformation in concrete objects to be placed before generating the first prototype. Unfortunately, there is not a maintained tool support for this environment.

Language Of Temporal Ordering Specification (LOTOS) (Coutaz, Paterno, Faconti, & Nigay, 1993) is a formal specification language based on temporal ordering used for protocol specification in ISO OSI standards. It is an algebraic language that consists of two parts: a process algebraic part based on Milner's Calculus of Communicating Systems (CCS) and on Hoare's Communicating Sequential Processes (CSP), and a data algebraic part based on the abstract data type language (ACT ONE). These two aspects of LOTOS are complementary and independent. Indeed, the process algebra is used to model dynamic behaviours of systems, and ACT ONE is used to model data structures and value expression. It is mathematically well-defined and expressive: it allows the description of concurrency, nondeterminism, synchronous and asynchronous communications. It supports various levels of abstraction and provides several specification styles. Several tools, such as the EUCALYPTUS toolset, exist to support specification, verification and code generation.

Z interactor (Duke & Harrison, Technical Report SM/WP29, MRC-APU, 1994) is a notation that allows the predicates on interface state can be expressed logically. Interactors are components in the description of an interactive system that encapsulate a state, the events that manipulate the state and the means by which the state is made perceivable to users of the system (the presentation) (Duke, Faconti, Harrison, & Paternò, 1994).

Whizz'ed (Esteban, Chatty, & Palanque, 1995) is a visual programming tool for highly interactive interfaces. Whizz'Ed is based on a data-flow model of Whizz. Whizz'Ed allows describing the behaviour of graphical objects in a visual way through the use of a direct manipulation editor, with a building block metaphor.

Hybrid High-levels Nets (HyNet) (Wieting, 1996) is a methodology for modelling and simulation of hybrid systems. It integrates three established modelling approaches into one language that are a) High-level Petri Nets representing the basic discrete framework, b) differential algebraic equations (DAEs) using to describe continuous system behaviour and c) object-oriented concepts improving the expressiveness and compactness of models.

Flownets (Willans & Harrison, 2001) are descriptions of developed specifically of virtual environments. The discrete components of the interactions are described using traditional place-transition Petri-nets and the continuous components are described using constructs based on systems dynamic literature. The continuous constructs have a clear mapping to the external environment. Within the continuous description, thresholds enable the firing of transitions between discrete states and these states and transitions can enable flow control valves which activate continuous flow. This formalism has been successfully used to model the behaviour of a variety of interaction techniques. It also been used to model the behaviour of world objects within virtual environments with which the user perceive and interacts via interaction techniques.

Input CONfigurator (ICON) (Dragicevic & Fekete, 2001) is an editor designed to configure a set of input devices and connect them to actions into a graphical interactive application. It works with Java Swing and requires applications to describe their interactions styles in terms of ICON modules. It allows physically challenged users to connect alternative input devices and configure their interaction technique according to their needs. Unfortunately ICON doesn’t allow run-time created modules or recursion.

Started from the early work of Parnas (Parnas, 1969) based on finite state automaton which was only able to model application featuring a small number of states, we have reached more expressive notations like Petri nets (Petri, 1962) in order to deal with large scale applications. Over the years, other notations based on Petri nets have been proposed but it seems to already provide a power led to the realisation and the modelling activity for a partly autonomous interactive system.

2.2 Focus on the Interactive Cooperative Objects (ICOs) and its associated CASE tool PetShop

This section presents the Interactive Cooperative Objects (ICOs) (Navarre, Palanque, & Bastide, 2001) formalism that has been used for 20 years in Research and Technology projects in the fields of ATM, Aerospace and Aeronautics. ICO enables to describe, in a formal way, all the components of highly interactive applications, including post-WIMP (Windows Icons Menus and Pointers) applications. The ICOs formalism is a formal description technique dedicated to the specification of interactive systems (Navarre et al. 2003). It uses concepts borrowed from the object-oriented approach (dynamic instantiation, classification, encapsulation, inheritance, client/server relationship) to describe the structural or static aspects of systems, and uses high-level Petri nets (Genrich, 1991) to describe their dynamic or behavioural aspects.

An ICO specification fully describes the potential interactions that users may have with the application. The specification encompasses both the "input" aspects of the interaction (i.e. how user actions impact on the inner state of the application, and which actions are enabled at any given time) and its "output" aspects (i.e. when and how the application displays information relevant to the user).

Time-out transitions are special transitions that do not belong to the categories above. They are associated with a timer that automatically triggers the transition when a dedicated amount of time has elapsed. When included in a system model such transition is considered as a system transition. They can also be included in a user model representing spontaneous user's activity. An ICO specification is fully executable, which gives the possibility to prototype and test an application before it is fully implemented (Bastide, Navarre, Palanque, Schyn, & Dragicevic, 2004). The specification can also be validated using analysis and proof tools developed within the Petri nets community.

ICOs are dedicated to the modelling and the implementation of event-driven interfaces, using several communicating objects to model the system, where both behaviour of objects and communication protocol between objects are described by Petri nets. The formalism made up of both the description technique for the communicating objects and the communication protocol is called the Cooperative Objects formalism (CO).

In the ICO formalism, an object is an entity featuring four components (Basnyat, 2006):

1. Cooperative Object (CO): a cooperative object models the behaviour of an ICO. It states how the object reacts to external stimuli according to its inner state. This behaviour, called the Object Control Structure (ObCS) is described by means of high-level Petri net. A CO offers two kinds of services to its environment. The first one, described with Java software interfaces, concerns the services (in the programming language terminology) offered to other objects in the environment. The second one, called user services, provides a description of the elementary actions offered to a user, but for which availability depends on the internal state of the cooperative object (this state is represented by the distribution and the value of the tokens (called marking) in the places of the ObCS).
2. Presentation part: the Presentation of an object states its external appearance. This Presentation is a structured set of widgets organized in a set of windows. Each widget may be a way to interact with the interactive system (user → system interaction) and/or a way to display information from this interactive system (system → user interaction).
3. Activation function: the user → system interaction (inputs) only takes place through widgets. Each user action on a widget may trigger one of the ICO's user services. The relation between user services and widgets is fully stated by the activation function that associates to each couple (widget, user action) the user service to be triggered.
4. Rendering function: the system → user interaction (outputs) aims at presenting to the user the state changes that occurs in the system. The rendering function maintains the consistency between the internal state of the system and its external appearance by reflecting system states changes.

ICOs are used to provide a formal description of the dynamic behaviour of an interactive application. An ICO specification fully describes the potential interactions that users may have with the application. The specification encompasses both the "input" aspects of the interaction (i.e., how user actions impact on the inner state of the application, and which actions are enabled at any given time) and its "output" aspects (i.e., when and how the application displays information relevant to the user). Time-out transitions are special transitions that do not belong to the categories above. They are associated with a timer that automatically triggers the transition when a dedicated amount of time has elapsed. When included in a system model such transition is considered as a system transition. They can also be included in a user model representing spontaneous user's activity.

An ICO specification is fully executable, which gives the possibility to prototype and test an application before it is fully implemented (Navarre, Palanque, & Bastide, 2004). The specification can also be validated using analysis and proof tools developed within the Petri nets community and extended in order to take into account the specificities of the Petri net dialect used in the ICO formal description technique.

PetShop[4], is the CASE tool associated CASE to the ICO formalism. It allows editing models and their execution.

3 Identify requirements for existing system modelling techniques and associated methods: what is still missing

The wide range of existing system modelling techniques and associated methods highlights the scientific community's effort and its recurring desire of increasing the bandwidth between the interactive system and users. For this reason, more sophisticated interaction techniques are continuously being proposed. The presented ones have been selected either because they provide a unique and important contribution or because they played a pioneering role in the area of User Interface Description Languages (UIDLs).

According to Hamon et al. (Hamon, Palanque, Silva, Deleris, & Barboni, 2013), if the users' tasks are complex, requiring, for instance, the execution of multiple commands in a short period of time or the manipulation of large data sets (e.g. pilot checking the weather radar implemented in the cockpit), it is likely that the new interaction techniques will significantly improve the overall performance of the operators. However, in such cases, the development process will at the minimum be more difficult (resources consumption will increase throughout the design, development and evaluation stages) or even be impossible if tools and techniques available for the development do not bring the required level of quality in the final product. This quality is assessed by the dependability level of the interactive system which must be compliant with the requirements set by the certification authorities. The same authors present a set of requirements which are an extension of a previous work presented by Navarre et al. (Navarre, Palanque, Ladry, & Barboni, 2009). They propose twelve different properties of the language that characterise the expressiveness of the UIDL.

The first three characteristics deal with data and data description. In particular:

1) REQ_SM_1: the description of objects and values in the language should be represented explicitly (this is named "Data Description")
2) REQ_SM_2: the description of states should be represented explicitly ("State Representation") and
3) REQ_SM_3: the description of events should be represented explicitly ("Event Representation")

Moreover, time is an important characteristic for behavioural description of interactive applications. This can be:

[4] http://www.irit.fr/recherches/ICS/softwares/petshop/

4) REQ_SM_4: Qualitative time between two consecutive model elements should be made explicit for representing ordering of actions such as precedence, succession, and simultaneity
5) REQ_SM_5: Quantitative time between two consecutive model elements should represent behavioural temporal evolutions related to a given amount of time (usually expressed in milliseconds)
6) REQ_SM_6: Quantitative time over non-consecutive elements for multimice double and fusion double click interactions should be explicitly represented. This requirement derives from Palanque et al. (Palanque, Barboni, Martinie De Almeida, Navarre, & Winckler, 2011).

Another requirement is the concurrent behaviour (REQ_SM_7) which is necessary when the interactive systems feature multimodal interactions or can be used simultaneously by several users.

A dynamic instantiation of interactive objects is a characteristic required for the description of interfaces where objects are not available at the creation of the interface as, for instance, in desktop-like interfaces where new icons are created according to user actions. Supporting explicit representation of dynamic instantiation requires the UIDL to be able to explicitly represent an unbounded number of states, as the newly created objects will by definition represent a new state for the system. Most of the time, this characteristic is handled by means of code and remains outside the UIDLs. Only Petri-nets-based UIDLs can represent explicitly such a characteristic, provided they handle high-level data structures, or objects. So, the description language must be able to receive dynamically created objects. This dynamicity has also to be addressed at operation time (i.e. when the system is currently in use). For instance, to cope with potential hardware failure reconfigurations of the interaction techniques might be required. Moreover, in order to ensure the availability of every system commands and maintain a high level of usability, the mapping between interaction techniques and commands shall be resolved during run-time. Thus, dynamic instantiation is related to widgets and devices. The corresponding set of requirements is presented hereafter:

8) REQ_SM_8: dynamic instantiation of widgets should be represented explicitly
9) REQ_SM_9: dynamic instantiation of devices should be represented explicitly
10) REQ_SM_10: reconfiguration of interaction technique should be represented explicitly
11) REQ_SM_11: reconfiguration of low level events should be represented explicitly

Furthermore, an added value feature in system modelling techniques and associated methods is the formal analysis capability. This capability enables to reduce time and resources spent on user studies as the analysis can be accomplished in an automated way with the models as input (Dix, 1995). It requires a formal description of the system and of the interactions and can be separated into three groups addressed by different types of analysis techniques: a) validation accomplished by interactive simulation (step by step), invariant, structural and reachability/coverability graph analysis; b) verification, accomplished by invariant, structural and reachability/coverability graph analysis; c) performance analysis, accomplished by simulation (Paternò & Santoro, 2000). The results of the analysis aim at:

- detecting impreciseness or potential problems through the analysis of formal description (Dix, 1995).
- validating the existence of required properties (Paternò & Santoro, 2000) (initially proposed to validate the existence of sub-properties belonging to usability and could be applied to validate the existence of sub-properties belonging to resilience),
- studying the performance of the proposed interaction techniques (Palanque, Barboni, Martinie De Almeida, Navarre, & Winckler, 2011).

This formal analysis capability is the 12th requirement (REQ_SM_12).

Moreover, these system modelling techniques and associated methods should allow the integration with other methods and modelling techniques ensuring compatibility at model and tool level (REQ_SM_13).

Finally, in order to deal with system modelling and analysis, these techniques and associated methods need to be supported by a tool (REQ_SM_14).

4 Comparison of system modelling techniques and associated methods according to the identified requirements

In this chapter we have offered an overview of the existing system modelling techniques and associated methods. Moreover, we have described a set of requirements needed for representing in a complete and systematic way system behaviour and user interactions through system modelling techniques and associated methods. In this section we make a comparison of system modelling techniques and associated methods according to the identified requirements.

In Hamon et al. (Hamon, Palanque, Silva, Deleris, & Barboni, 2013), the authors offer a detailed comparison of several system modelling techniques and associated methods focused on multi-touch and multi-modal interactions. According to the scope of this section and this PhD thesis, we propose a reviewed version of their comparison including the presented system modelling techniques and associated methods in the previous section.

Indeed, the following table reports in the rows the described requirements and in the columns the system modelling techniques and associated methods. For all requirements, there are three possible values:

- “Yes” (green box with YES) means that characteristic is explicitly handled by the UIDL,
- “No” (white box with NO) means that the characteristic is not explicitly handled,
- “Partially” (grey box) means that the property is partly present.

As shown in the table, most of these techniques deal with data. These are described in many UIDLs such as ICON (Dragicevic & Fekete, 2001), which allows modelling data emission and reception from an output port of a device of the model to the input port of another device. Some UIDLs can also represent states of the system, such as ICON (Dragicevic & Fekete, 2001), which represents the states with nodes in the models. Events are also sometimes explicitly represented as in Wizz’ed (Esteban, Chatty, & Palanque, 1995) where connections between bricks represent event flows.

While all of the reported system modelling techniques and methods explicitly handle with the qualitative time, only few of them handle with both quantitative time requirements.

Table 4: Comparison of system modelling techniques and associated methods according to suggested requirements for UIDL expressiveness (taken and adapted from (Hamon, Palanque, Silva, Deleris, & Barboni, 2013))

	Requirements / Notations	ATN	AUTOMATA	TRIDENT	LOTOS	Z Interactor	Wizz'ed	HyNet	Flownet	ICON	ICO
Data	**REQ_SM_1**. Data Description				Yes	Yes	Yes	Yes	Yes	Yes	Yes
	REQ_SM_2. State representation	Yes	Yes		No	Yes	No	Yes	Yes	No	Yes
	REQ_SM_3. Event representation	Yes	Yes		Yes	Yes	Yes	Yes	Yes	Yes	Yes
Time	**REQ_SM_4**. Qualitative between two consecutive model elements	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	REQ_SM_5. Quantitative between two consecutive model elements	No	No	No	No	No	No	No	No	No	Yes
	REQ_SM_6. Quantitative over non consecutive elements	No	No	No	No	No	No	No	No	No	Yes
Behavior	**REQ_SM_7**. Concurrent Behavior	No	No		Yes	No	Yes	Yes	Yes	Yes	Yes
Dynamic Instantiation	**REQ_SM_8**. Dynamic instantiation of widgets	No	No		No	No	No	No	No	No	Yes
	REQ_SM_9. Dynamic instantiation of input devices	No	No	No	No	No	No	No	No	No	Yes
	REQ_SM_10. Reconfiguration of Interaction technique	No	No	No		No	No	No	No	No	Yes
	REQ_SM_11. Reconfiguration of low level events	No	No	No	No	No	No	No	No	No	Yes
Analysis	**REQ_SM_12**. Formal analysis capability	No	No			No	No			No	
Integration	**REQ_SM_13.** Integration with other methods and modelling techniques ensuring compatibility at model and tool level	No	No	No	No	No	No	No	No	No	Yes
Tool support	**REQ_SM_14:** Supported by a tool	APE	VRED	TRIDENT	EUCALYPTUS	No tool support	Wizz'ed	No tool support	Marigold	Icon toolkit	PetShop

Yes	The system modelling technique handles with the requirement
No	The system modelling technique doesn't handle with the requirement
	The system modelling technique partially handles with the requirement

Regarding to the representation of concurrent behaviour, this can be made explicit in the models like in data flow notations, as in ICon (Dragicevic & Fekete, 2001) or Whizz'Ed (Esteban, Chatty, & Palanque, 1995) and in all the notations based on Petri nets (last four columns of Table 4). These are the only ones able to explicitly represent dynamic instantiations. In addition, just few of them can deal with multimodality and analysis requirements.

As reported in the table, ICO is the notation which explicitly handles the all requirements. In addition, ICO has a genuine capability of being connected to task models.

5 Conclusion, lessons learnt and way forward

In this chapter we have provided an overview of the current state of the art in system analysis and modelling to support the analysis and assessment of usability and resilience of partly autonomous interactive systems.

According to automation levels, which determine the type human and automation interaction, it is needed to understand how to balance automation and interactivity and specify how a task can be performed by assigning the generic functions to the human and the machine operator in terms of function allocation.

The wide range of existing system modelling techniques and methods highlights the scientific community's effort and its recurring desire of increasing the bandwidth between the interactive system and users. For this reason, more sophisticated interaction techniques are continuously being proposed.

These have been compared through a set of identified requirements which are needed for representing in a complete and systematic way system behaviour and user interactions through system modelling techniques and methods. The comparison has highlighted that the most suitable system modelling technique is ICO. This notation has been carefully chosen for two reasons:

- ICO explicitly handles the all requirements
- it has a genuine capability of being connected to task models.

Indeed, system models provide a complete description of system behaviour while task models gather information related to users goals and activities. As far as interactive systems are concerned, such description must make explicit all the possible states of the system and, for each state, which actions are available to the user on the interface. On the rendering side, the system model must describe, according to any state change how this state change is presented to the user. As the system model describes the actions available to the user and as the task model describes the actions that have to be performed by the user in order to reach a goal, these two models provide two different views on the same elements (Navarre, Palanque, Barboni, & Mistrzyk, 2007).

For these reasons, we focus on the possible articulations of task models and system models. As illustrated in Figure 10, this integration is done at the model level (thanks to the iterative construction of system and task models which provide a preliminary version of both models) as well as at the tool level exploiting PetShop environment for the system side and HAMSTERS for the task side. Since we are interested in specifying the behaviour of interactive systems, and more particularly partly autonomous interactive systems which require complete, concise and non-ambiguous descriptions, and thus we have focused our attention on ICOs which is the most appropriate formalism allowing us to specify the most relevant information.

Moreover, thanks to PetShop, we are able to model this type of system and at the same time to integrate the task model to represent human activities. This synergistic use allows us to "Check performance and complexity objectives" (Figure 10) and, if needed, to redefine them or to obtain a proper model "Towards User testing, Training design and operation" (Figure 10).

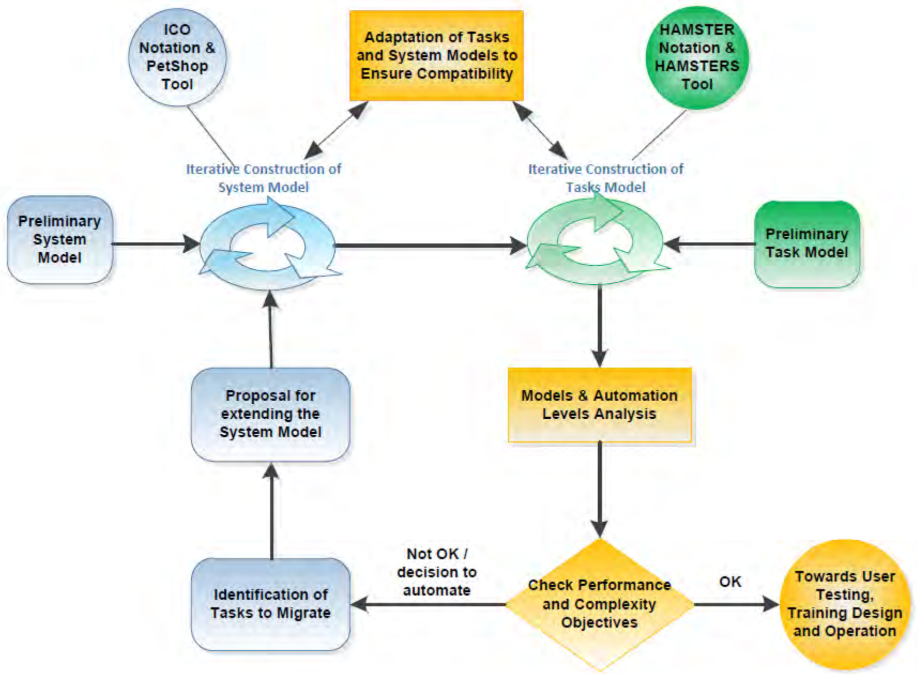


Figure 10: The iterative model-based design life cycle using both tasks and system models (taken from (Martinie, et al., 2011))

This synergistic use allows us in representing all the components of a partly autonomous interactive system and, in addition, in model interactions and how these can evolve over time due to system change or task migration. This iterative multi models approach can support automation design (Martinie, et al., 2011). Indeed, at each variation on the system model, which impacts on human tasks, corresponds a change on the task model and vice versa.

This approach enables to handle large amounts of increasingly complex information between users and systems. Furthermore, it also improves the current research work in the field of HCI which promotes the development of new interaction and visualisation techniques in order to increase the bandwidth between the users and the systems (Navarre, Palanque, Barboni, Ladry, & Martinie, 2011).

The presented synergistic use of these two types of models is the first step toward the federation of several types of models and then towards the second contribution of this PhD thesis with regard to a process for analysing and modelling a partly autonomous interactive system as a whole. This multi-models based approach is presented in Chapter 5.

Chapter 3 – Addressing organizational aspects through modelling techniques and associated methods

This chapter describes the current state of the art of methods and modelling techniques used to address the organisational aspects for the analysis and modelling of resilient partly autonomous interactive systems. The scope of this chapter is to:

- Explain why resilience analysis and modelling are needed to analyse and assess usability and resilience of partly autonomous interactive systems
- Describe and compare the existing resilience modelling techniques and methods
- Examine what is still missing in existing methods and modelling techniques to provide support for the modelling and the analysis of resilient partly autonomous interactive systems with regard to organisational aspects. Moreover, it proposes a set of requirements for comparing these methods and modelling techniques
- Make a comparison of methods and modelling techniques according to the identified requirements.

The first section offers a description in terms of definition and characteristics of resilience. Additionally, it explains why resilience analysis and modelling are needed to analyse and assess usability and resilience of partly autonomous interactive systems.

The second section offers a state of the art on resilience methods and modelling techniques aiming at describing the existing ones. It also includes a deeper description of one of the exiting methods which has been developed by Erick Hollnagel and currently used in ATM domain to investigate the resilience aspects in accidents and incidents analysis. This method is the Functional Resonance Analysis Method or FRAM (Hollnagel, 2012).

The third section highlights what is still missing and identifies requirements in addressing organisational aspects through methods and modelling techniques for the analysis and modelling of resilient partly autonomous interactive systems.

The fourth section offers a comparison of methods and modelling techniques according to the identified requirements.

1 Resilience analysis and modelling to support analysis and assessment of usability and resilience of partly autonomous interactive systems

The term resilience is used with different meanings in different domains of application. We tend to accept the definition provided by Hollnagel in his landmark books, that is "resilience is often defined in terms of the ability to continue operations or recover a stable state after a major mishap or event. This definition focuses on the reactive nature of resilience and the ability to recover after an upset. […] We use a more general definition that includes preventions of upsets. In our conception, resilience is the ability of the systems to prevent or adapt to changing conditions in order to maintain (control over) a system property" (Hollnagel, Woods, & Leveson, 2005, p. 94-95). However, the main remaining problem we see with that definition is that it remains at an abstract philosophic level and does not provide any direct indication to support its evaluation. In this definition, the term "system" used by the authors refers to the concept of Socio-Technical System. A Socio-Technical System (STS) is a term coined by Emery and Trist (Emery & Trist, 1960) to describe systems that involve a complex interaction between humans, machines and the environmental aspects of the work system. Nowadays, this interaction is true for most the enterprise systems and organisations. The corollary of this definition is that all of these factors—people, machines and organisations—need to be considered when developing, modelling and analysing such STS (Baxter & Sommerville, 2011). In order to avoid term confusion, in this chapter each time that the term is used with the same meaning of Hollnagel et al., we adopt "Socio-Technical System" or its acronym "STS".

As Hollnagel and Woods (Hollnagel & Woods, 2006) argue, the potential for resilience can be measured, but resilience itself cannot be measured. Since resilience refers to a capability of a Socio-Technical System, rather than to something a Socio-Technical System delivers (such as a product), it cannot be measured by counting specific STS outcomes such as accidents or incidents. Again, we tend to agree with the Hollnagel point of view and its approach to consider the role of resilience: "It focuses on the four main abilities that together constitute resilience: the ability to respond, the ability to monitor, the ability to anticipate and the ability to learn. These abilities can be assessed by means of a number of questions and the answers can be represented by an easily comprehensible graphical form. This can be used to compare consecutive measurements, and thereby as a way to support the management of a Socio-Technical System's resilience" (Hollnagel, 2011). The following picture, in Figure 11, illustrates the four main abilities of resilience.

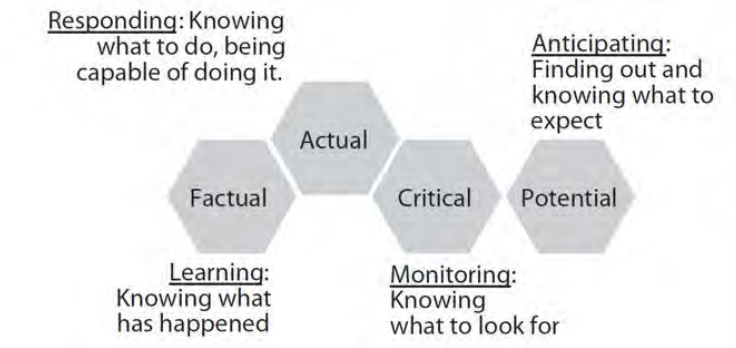


Figure 11: Resilience four main abilities (taken from (Hollnagel, 2011))

As illustrated in Figure 11, the four main abilities of a resilient Socio-Technical System can offer a more concrete view on resilience definition:

- Respond: knowing what to do, or being able to respond to regular and irregular variability, disturbances, and opportunities either by adjusting the way things are done or by activating ready-made responses. This is the capability to address the actual.
- Monitor: knowing what to look for, or being able to monitor that which changes, or may change, so much in the near term that it will require a response. The monitoring must cover the Socio-Technical System's own performance as well as changes in the environment. This is the capability to address the critical.

- Anticipate: knowing what to expect, or being able to anticipate developments, threats, and opportunities further into the future, such as potential disruptions or changing operating conditions. This is the capability to address the potential.
- Learn: knowing what has happened, or being able to learn from experience, in particular to learn the right lessons from the right experience. This is the capability to address the factual.

These abilities are mutually dependent and can be developed to different degrees. In this PhD thesis we focus on the ability to monitor resilience. This ability can enable a Socio-Technical System or an organization to manage and adjust its capacity of adaptation when facing disturbances (these can impact on the STS and generate variability – please refer to Section 2.2 where it is explained in detail). This ability provides support for recognising and interpreting evidences of new vulnerabilities. By identifying components of the Socio-Technical System that may be vulnerable (e.g. the human, the system or the organizational aspects), it provides opportunities to improve these components. The concept of functional resonance, developed within the resilience engineering paradigm (Hollnagel, 2011), describes accidents as the detectable signal that emerge from the unintended interaction of everyday operations. So, differently from the classical view of resilience in which the focus is on the reactive nature of resilience and the ability to recover after an upset or analysed once the accidents have happened on existing STS without any support to deal with its analysis at design time, in this thesis, we propose a resilience perspective based on its integration at the early stage of development process in order to identify and prevent resilience issues rather than analyse them in a post-hoc accident analysis. The functional resonance is based on the principle of equivalence of successes and failures and the principle of approximate adjustments (please refer to Chapter 3 – Section 2.2). Performance is therefore in practice always variable (Hollnagel, 2011). The performance variability of upstream functions may affect the performance variability of downstream functions, and thereby lead to non-linear effects. An upstream function is defined as a function that – in a given instantiation – happens before other functions, and which therefore may affect them. Functions that – in the instantiation – happen after other functions, and which therefore may be affected by them, are called downstream functions.

Our approach (please refer to Chapter 5) rests on modelling and analysing STS and using the information gained in designing the socio-technical system, in evaluating both planned responses to events and to prevent adverse ones, and to detect changes and analyse them in case of task migrations. To be useful, such modelling and analysis must be able to handle complex, tightly coupled systems with distributed human and automated control, advanced technology and software-intensive systems, and the organizational and social aspects of systems (Leveson, et al., 2007)

To apply this early resilience analysis and modelling in a partly autonomous interactive system development process, models-based and scenarios-based approaches are required (as explained in Chapter 1) as well as complementary experts' support to envision the possible consequences. These aspects of the proposed approach are described in the contribution chapter 5 and in the case study chapter (please refer to the Chapter 6).

As explained in Chapter 1 and in Chapter 2 with regard to task and system analysis and their modelling techniques, we can assume the same reasoning with regards to resilience analysis and modelling techniques.

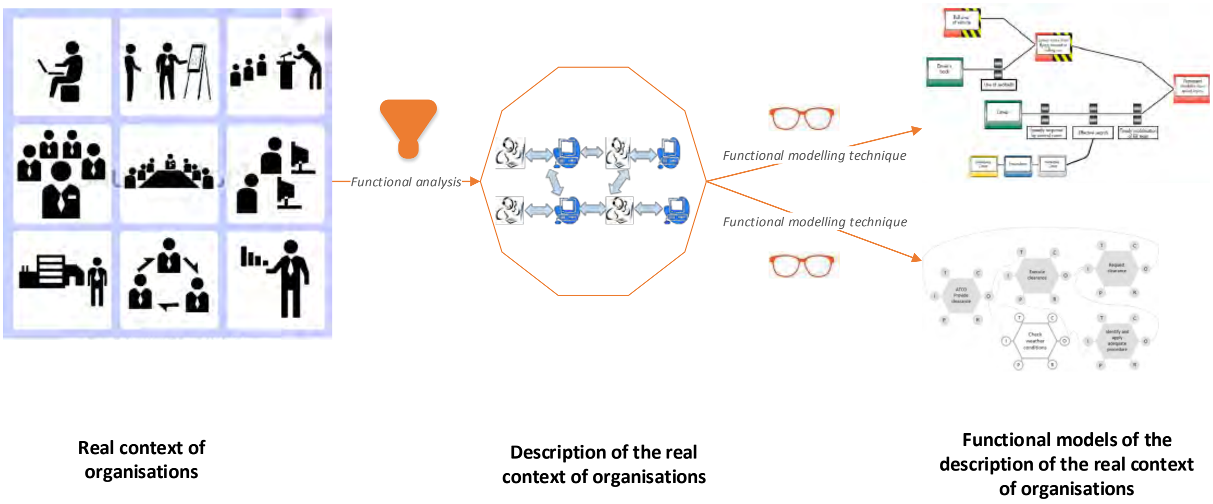


Figure 12: Functional methods and modelling techniques to represent the real context of organisations

In Figure 12, the complexity of the real context of organisations is filtered out through an orange filter that represents functional analysis. This "functional analysis" filter allows focusing on specific pieces of information about the real context of organisations by using functional analysis notation elements to describe real context of organisations. The aim of this filtering out operation is to obtain a description of the real context of organisations. The description of the real context of organisations relies on the chosen notation, i.e. on aspects that are relevant to the analyst. The analyst may then choose between several functional modelling techniques to record this description in functional models. Each functional modelling technique and associated methods can provide a different way to represent the organisation. This point is illustrated in Figure 12 where there are 2 orange glasses for representing different functional modelling techniques and their specific notations such as TRIPOD (Groeneweg, 2002) in the upper part, or FRAM (Hollnagel, 2004) in the lower part (Section 3.1 presents a state of the art on existing notations). Depending on the choice of notation, the functional modelling operation will result in a different model of the organisation since each type of model highlights particular aspects of the assumed real context of organisations. For this reason, it is important to choose the most suitable functional modelling technique. Thanks to the selected functional modelling technique, the analyst can highlight the aspects that are relevant to the goals of his/her analysis.

2 State of the art on modelling techniques and associated methods for resilience analysis

The evolution of partly autonomous interactive systems is continuous and rapidly driven by a combination of technological innovation, commercial considerations, and users' demands while the risk assessment and resilience analysis methods and modelling techniques develop at a far more moderate pace. It means that they are not able to model and address the current complexity of these Socio-Technical Systems because, in general, they focus on a specific, salient factor of an event such as an accident or incident cause (Hollnagel & Speziali, 2008). For this reason, a number of methods and techniques have been improved and/or developed to support the analysis of resilience. Some of them have been developed and evaluated by the scientific community in cooperation with the industries, such as the ones studied in project funded by SESAR[5] and by the European Commission (RESIST[6], SPAD[7] and SCALES[8])

This section proposes:

- a detailed analysis of a number of established resilience modelling techniques and associated methods including their notations and supporting tools,
- a focus on the Functional Resonance Analysis Method (FRAM) because it supports both accident investigation and functional resonance assessment processes (Hollnagel, 2004). It is based on a set of principles related to complex socio-technical systems structure and dynamic. Furthermore, it provides support for analysing functional variability in terms of non-linear dynamic dependencies among functions rather than simple or complex cause effect relations. These features offer us the possibility to focus on, model and analyse partly autonomous interactive systems

2.1 An overview of modelling techniques and associated methods for resilience analysis

If it is acknowledged that risks and accidents can arise from unexpected combinations of normal everyday activities, then the assumption of causality must be partly abandoned. If risks and accidents cannot always be linked to failures and malfunctions of components, then methods should not be restricted to causal explanations (Hollnagel, 2014). The alternative is to develop methods and modelling techniques that describe Socio-Technical System functions rather than components or structures, as STS functions can account for the non-linear propagation of events. Resilience analysis can be achieved by using functional resonance instead of causality, and by using normal performance variability instead of malfunctioning (Hollnagel, 2008).

In this section, we offer an overview of representative methods and modelling techniques for resilience analysis. This section presents the evolution and recent advances about these methods and techniques. They are presented in chronological order.

Fault Tree Analysis (FTA) was originally developed in 1962 at Bell Laboratories by H.A. Watson, under a U.S. Air Force Ballistics Systems Division contract to evaluate the Minuteman I Intercontinental Ballistic Missile (ICBM) Launch Control System (Ericson, 1999). And then, engineers at Boeing further developed and refined the procedures and they argue that this method was well suited for performing safety analysis of complex electromechanical systems (International Electrotechnical Commission (IEC), 2006)[9]. This method is a top down, deductive failure analysis in which an undesired state of a system is analysed using Boolean logic to combine a series of lower-

[5] The Single European Sky ATM Research aims at developing a modernized air traffic management system for Europe, which will prevent crippling congestion of the European sky and reduce the environmental impact of air transport.

[6] http://www.resist-noe.org/

[7] http://www.irit.fr/recherches/ICS/projects/spad/

[8] http://www.hala-sesar.net/scales

[9] The IEC Standard IEC 61025 is intended for cross-industry use and has been adopted as European Norm EN 61025. However, FTA methodology is described in several industry and government standards, including NRC NUREG–0492 for the nuclear power industry, an aerospace-oriented revision to NUREG–0492 for use by NASA, SAE ARP4761 for civil aerospace, MIL–HDBK–338 for military systems.

level events. This methodology is mainly used in the fields of safety engineering and reliability engineering to understand how systems can fail, to identify the best ways to reduce risk or to determine event rates of a safety accident or a particular system level (functional) failure. A fault tree consists of fault events, branches and tree gates. Events are failure situations resulting from the logical interaction of primary failures or of failures of interest. Branches connect two events or a tree gate and an event. Gates are Boolean logic symbols that relate the input to its output. A system is represented by a series of these components making a fault tree. FTA can be supported by several tools. Classic programs include the Electric Power Research Institute's named Computer Aided Fault Tree Analysis System (CAFTA)[10] software, which is used by many of the US nuclear power plants and by a majority of US and international aerospace manufacturers, and the Idaho National Laboratory's SAPHIRE[11], which is used by the U.S. Government to evaluate the safety and reliability of nuclear reactors, the Space Shuttle, and the International Space Station. Outside the US, the software RiskSpectrum[12] is a tool for Fault Tree and Event Tree analysis and is licensed for use at almost half of the world's nuclear power plants for Probabilistic Safety Assessment.

Failure Mode, Effects and Criticality Analysis (FMECA) (U.S. Department of Defense, 1980) is an extension of Failure Mode and Effects Analysis (FMEA) (U.S. Department of Defense, 1949). FMEA is a bottom-up, inductive analytical method which may be performed at either the functional or piece-part level. FMECA extends FMEA by including a criticality analysis, which is used to chart the probability of failure modes against the severity of their consequences. The result highlights failure modes with relatively high probability and severity of consequences, allowing remedial effort to be directed where it will produce the greatest value (Bouti & Kadi, 1994). FMECA tends to be preferred over FMEA in space and North Atlantic Treaty Organization (NATO) military applications (i.e. ECSS-Q-30-02A standard which is part of a series of ECSS Standards belonging to the ECSS-Q-30 "Space product assurance – Dependability"[13]), while various forms of FMEA predominate in other industries.

Swiss cheese model (SCM) (Reason J. T., 1990) where an organization's defences against failure are modelled as a series of barriers, represented as slices of Swiss cheese. The holes in the cheese slices represent individual weaknesses in individual parts of the Socio-Technical System, and are continually varying in size and position in all slices. The Socio-Technical System as a whole produces failures when all of the holes in each of the slices momentarily align, permitting "a trajectory of accident opportunity", so that a hazard passes through all of the holes in all of the defences, leading to a failure. The TRIPOD (Groeneweg, 2002) concept and set of methods can be considered in a sense as the origin of the SCM. The idea behind TRIPOD is that organisational failures are the main factors in accident causation. These factors are more "latent" and, when contributing to an accident, are always

---

[10] It is publicly available: http://www.epri.com/abstracts/Pages/ProductAbstract.aspx?ProductId=000000003002000020

[11] The latest version of SAPHIRE is distributed through the SAPHIRE Users Group (https://saphire.inl.gov/)

[12] It is distributed by Lloyd's Register Consulting (http://www.riskspectrum.com/en/risk/Meny_2/RiskSpectrum_FTA/)

[13] This Standard defines the principles and requirements that shall be adhered to with regard to FMECA implementations in all elements of space projects in order to meet the mission performance requirements as well as the dependability and safety objectives, taking into account the environmental conditions. It defines requirements and procedures for performing a FMECA to systematically evaluate and document the potential impact of each failure on product operation and mission success, personnel and product safety, maintainability and maintenance requirements. Recommended forms and formats are identified in this Standard.

followed by a number of technical and human errors. HFACS (Human Factors Analysis and Classification System) is used by the Federal Aviation Agency (US).

Root Cause Analysis (RCA) (Wilson, Dell, & Anderson, 1993) identifies underlying deficiencies in a safety management system that, if corrected, would prevent the same and similar accidents from occurring. RCA analysis is a systematic process that uses the facts and results from the core analytic techniques to determine the most important reasons for the accident. The modelling activity is supported by TapRoot® (Paradies & Unger, 2000). Moreover, there are other methods related to RCA: The Human Performance Evaluation System (HPES) and HINT which is a recent development of J-HPES, the Japanese version of the HPES.

Människa-Teknologi-Organisation or Man-Technology-Organisation (MTO) (Rollenhagen, 1995) is based on HPES and the basis for the MTO-analysis is that human, organisational, and technical factors should be focused equally in an accident investigation. The MTO method has been extensively used by the Swedish NPPs. The principle is also widely used in other domains, such as traffic safety and aviation. The MTO methods has many features common with other methods (Swiss cheese, HPES), but distinguishes itself from the single-factor methods.

Cognitive Reliability and Error Assessment Method (CREAM) (Hollnagel, 1998) is a so-called second generation HRA methods, but differs from other methods of the same type (ATHEANA, MERMOS) by being explicitly developed for both accident investigation and risk assessment. CREAM was developed to be used both predictively and retrospectively. CREAM uses the Contextual Control Model (COCOM) as a basis for defining four different control modes (strategic, tactical, opportunistic, and scrambled). It is assumed that a lower degree of control corresponds to less reliable performance. The level of control is mainly determined by the common performance conditions (CPC). The retrospective use (accident analysis) is based on a clear distinction between that which can be observed (called phenotypes) and that which must be inferred (called genotypes). The genotypes used in CREAM are divided into three categories: individual, technological and organisational, corresponding to the MTO triplet. The method does not look for specific causes, but rather for the operational conditions that can lead to a loss of control, hence accidents. It is grounded in cognitive systems engineering. Similar to other second generation methods it rejects consider human error as a meaningful causal category. The basis for the analysis is the event as it happened, rather than preconceived causal factors.

Accident Evolution and Barrier Analysis (AEB) (Svensson, 2001) provides a method for the analysis of incidents and accidents that models the evolution towards an incident/accident as a series of interactions between human and technical systems. The interaction consists of failures, malfunctions or errors that could lead to or have resulted in an accident. The method forces analysts to integrate human and technical systems simultaneously when performing an accident analysis.

A HAZard and OPerability study (HAZOP) (British Standards Institution, 2002) [14] is a structured and systematic examination of a planned or existing process or operation in order to identify and evaluate problems that may represent risks to personnel or equipment, or prevent efficient operation; it is carried out by a suitably experienced multi-disciplinary team (HAZOP team) during a set of meetings. The HAZOP technique is qualitative, and aims to stimulate the imagination of participants to identify potential hazards and operability problems; structure and completeness are given by using guideword prompts. The Standard (British Standards Institution, 2002) calls for team members to display

[14] This British Standard reproduces verbatim IEC 61882:2001 and implements it as the UK national standard

"intuition and good judgement' and for the meetings to be held in 'a climate of positive thinking and frank discussion". The HAZOP technique was initially developed to analyse chemical process systems and mining operation process but has later been extended to other types of systems and also to complex operations such as nuclear power plant operation and to use software to record the deviation and consequence.

Human Error in European Air Traffic Management (HERA)[15] (Isaac, Shorrock, & Kirwan, 2002) is a method to identify and quantify the impact of the human factor in incident/accident investigation, safety management and prediction of potential new forms of errors arising from new technology. Human error is seen as a potential weak link in the ATM system and measures must therefore be taken to prevent errors and their impact, and to maximise other human qualities such as error detection and recovery. HERA is predicated on the notion that human error is the primary contributor to accidents and incidents. Three related tools are available to predict the possibility of human error in the Socio-Technical System (HERA-PREDICT), observe errors and their management (HERA-OBSERVE), and to support the management of safety (HERA-SMART). This method is related to the Technique for Retrospective Analysis of Cognitive Errors (TRACEr) which has tried to harmonise active and reactive approaches to Human Error Analyses in the realm of ATM and Predictive Analysis of Cognitive Errors (EUROCONTROL, 2004).

System-theoretic model of accidents (STAMP) (Leveson N. G., 2004) which is based on the hypothesis that system theory is a useful way to analyse accidents, particularly system accidents. Accidents occur when external disturbances, component failures, or dysfunctional interactions among system components are not adequately handled by the control system. Safety is viewed as a control problem, and is managed via constraints by a control structure embedded in an adaptive socio-technical system. Understanding why an accident occurred requires determining why the control structure was ineffective. Preventing future accidents requires designing a control structure that will enforce the necessary constraints. Systems are viewed as interrelated components that are kept in a state of dynamic equilibrium by feedback loops of information and control. STAMP claims to be general method for explanation of mishaps with teleological systems. STAMP uses a specific causal model, i.e., a feedback control system. The basic principle is that an accident occurs when operational constraints have been violated. STAMP investigates systems involved, especially human organisational subsystems, to identify missing or inappropriate features (those which fail to maintain the constraints). It proceeds through analysing feedback & control (F&C) operations, which replaces the traditional chain-of-events model. The model includes software, organizations, management, human decision-making, and migration of systems over time to states of heightened risk.

ARKTRANS (Natvig & Westerheim, 2008) is an architectural notation which is based on the Enterprise Architecture approach. This is typically applied on complex environments or systems. In the standards specifying the approach (Hilliard, 2000) (ISO, 2011), the term system is simply used as a placeholder such that the scope and focus of the architecture could be a business enterprise, a software system or a system-of-software systems. An enterprise architecture typically prescribes a holistic approach where the technology is not isolated from the human factor; both aspects are treated equally important, and there is often an integration of technical system components and human stakeholders (often specified as roles with instantiations of human actors) and application of separation of concern through the use of perspectives and viewpoints. The current standard (ISO, 2011) prescribes how to develop enterprise architecture frameworks and the manner in which architecture descriptions of

---

[15] http://www.eurocontrol.int/services/human-error-atm-hera

systems are organized and expressed. Further, it specifies architecture frameworks defined as conventions, principles and practices for the description of architectures established within a specific domain of application and/or community of stakeholders and architecture viewpoints defined as a work product expressing the architecture of a system from the perspective of specific system concerns for use in architecture descriptions. So, a view is governed by its viewpoint: the viewpoint establishes the conventions for constructing, interpreting and analysing the view to address concerns framed by that viewpoint. The separation of the whole system of interest into different viewpoints enables a focus on specific concerns while hiding other possibly irrelevant details. As such the user (e.g. a software developer, an analyst or end-user) can zoom in on specific aspects while disregard other aspects. And, as the different viewpoints in union compose a more complete system the user can zoom out and get a more holistic view on the system. Different viewpoints may use different notation, but often UML is used in some variant. It is the case of ARKTRANS which uses UML consistently as its architectural notation. The roles and the functional viewpoint prescribe the use of UML Use Case notation; the process viewpoint uses UML activity diagrams; and the information viewpoint applies UML class diagrams as the notation for describing information elements. This methodology has over a number of years been used in the development of enterprise architectures related to the transport domain. It prescribes a top-down approach, starting with conceptual aspects at the top, working its way down to logical aspects and finally down to technological aspects. In a recent work, ARKTRANS has been improved using BPMN to model processes whereas ARKTRANS uses UML Activity diagrams. The notation is pretty similar, BPMN is more expressive with its extended use of stereotypes and especially the possibility to express different activity types (e.g. in BPMN one can clearly state through symbols and tagged values that an activity is performed by a human or a technical system). This new framework (Herrera, Pasquini, Ragosta, & Vennesland, 2014) developed in SCALES[16] project is also supported by Enterprise Architect tool (Sparx Systems, 2010).

2.2 Focus on the Functional Resonance Analysis Method (FRAM)

FRAM is a method that aims at supporting both accident investigation and functional resonance assessment processes (Hollnagel, 2004). It is based on four of principles related to complex socio-technical systems structure and dynamic:

1. the equivalence of success and failures because these last ones represent the adaptions necessary to cope with the under specification found in complex real-world classifications,
2. the principle of approximate adjustments because to get anything done people must adjust their performance to the current conditions and the resources and time are finite, such adjustment will inevitably be approximate,
3. the principle of emergence, both failures and normal performance are emergent phenomena: neither can be attributed to or explained simply by referring to the (mal)functions of specific components or parts,
4. the principle of functional resonance substitutes the traditional cause-effect relationship. The resonance explains how disproportionate large consequences can arise from seemingly small variations in performance and conditions.

Application of FRAM method is based on functional models where functions are describes with six aspects (as illustrated in Figure 13):

- Input (I): that which the function processes or transforms or that which starts the function,

[16] http://www.hala-sesar.net/scales

- Output (O): that which is the result of the function, either an entity or a state change,
- Preconditions (P): conditions that must be exist before a function can be carried out,
- Resources (R): needed by the function when it is performed (Execution Condition) or consumed to produce the Output,
- Time (T): temporal constraints affecting the function (i.e. starting time, finishing time, or duration), and
- Control (C): how the function is monitored or controlled.

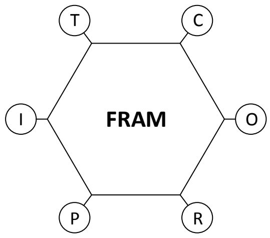


Figure 13: A FRAM function with its 6 aspects (taken and adapted from (Hollnagel, 2004))

FRAM method is structured in four main phases:

1. Identify the essential functions that are necessary (and sufficient) for the intended performance to occur (when 'things go right'). The functions can be assigned to either the set of foreground functions or the set of background functions. Characterise using the six basic aspects (Input, Output, Pre-conditions, Resources, Time, and Control). Taken together, the functions are sufficient to describe what should happen (i.e., the everyday or successful performance of a task or an activity).
2. Characterise the variability, first as the potential of the functions described by the model, and then as the (possible) actual variability for a set of instantiations of the model. Consider whether the actual variability will be what one should expect ('normal') or whether it will be unusually large ('abnormal').
3. Identify the dynamic couplings (functional resonance) that likely will play a role during an event. These comprise an instantiation of the model which can be used to predict how an event will develop and whether control can be lost. In relation to the traditional risk assessment, this instantiation provides an explanation of what may happen, although it does not necessarily identify unique or specific outcomes. The explanation will be based on the couplings of the variability of everyday performance, rather than failures and malfunctions.
4. Propose ways to monitor and dampen performance variability (indicators, barriers, design/modification, etc.) In the case of unexpected positive outcome, one should look for ways to amplify, in a controlled manner, the variability rather than for ways to dampen it.

The definition of functional variability in FRAM is based on the principle that the variability of the output of a function depends of the composition of three sources of variability: endogenous variability, exogenous variability and coupling variability. Variability of output can be described with a set of dimensions such as timing, precision, distance, speed, direction, force, magnitude, object, sequence or quantity. Endogenous source of variability is related to the internal variability of the Socio-Technical System (automation, human, group or organisation) that performs the function. Exogenous source of variability is related to the variability of the environment of execution of the function (working conditions, culture, etc.). Coupling source of variability is related to the variability of functions that are coupled with the studied function. This means that the model will make it possible to understand how degradation of a purely technological function may affect the overall performance of the Socio-Technical System (Macchi, Hollnagel, & Leonhard, Resilience Engineering Approach to Safety Assessment: An Application of FRAM for the MSAW system, 2009) (EUROCONTROL, 2009).

For partly autonomous complex systems, the degradation of functions can be explained in terms of non-linear dynamic dependencies among functions rather than simple or complex cause effect relations. The dilemma that experts and safety managers are facing is that the situation cannot be improved by eliminating performance variability, since performance variability is essential to ensure safety, productivity, and quality. The solution is instead to manage performance variability, typically by trying to dampen the variability in order to weaken resonance effects (Hollnagel, 2012). For dampening, the effective management of a Socio-Technical System or a set of processes not only requires that the current state is known, but also that the means for effective intervention in the process are available. The means should enable those in charge of the Socio-Technical System to make sure that developments go in the right direction and progress at the right speed. In the language of process control this means that the variability must be managed, specifically that it must be dampened when it looks like it will get out of hand and lead to adverse outcomes. Another solution for weaken resonance effects is to monitor and individuate performance indicators. Control in a socio-technical system, such as safety or quality management, requires that appropriate targets can be set, that ongoing processes and developments can be monitored, and that activities can be regulated to keep the processes on track so that both short-term and long-term goals can be attained. The selection of which indicators to monitor often involves a compromise among several criteria and interests – a trade-off between efficiency and thoroughness.

A FRAM model describes the functions of a Socio-Technical System and how they are mutually coupled. A FRAM model can in particular be used to identify the conditions where developments potentially may get out of control, for instance by identifying couplings that may lead to an increase in performance variability. A FRAM model can therefore be used as a basis for proposing performance indicators, and then as a basis for monitoring performance.

FRAM is a modelling approach that has been put forward as a novel way of describing and understanding the behaviour of complex Socio-Technical Systems. It provides a framework and a method for systematically describing and evaluating functions and performance variability. It has been applied in accident/incident analysis (Macchi, 2010) and, in these applications; it can offer an additional perspective and improve the knowledge about what has happened.

3 Identify requirements for existing resilience modelling techniques and associated methods: what is still missing

Resilience Engineering (RE) aims at improving the ability of partly interactive autonomous systems to continue operations under expected and unexpected conditions. It expands the focus of analysis of proactive Safety Management Systems including data about what goes wrong (defined as Safety I) as well as what goes right (defined as Safety II) (Hollnagel, 2014). In this perspective, it is essential to monitor the Socio-Technical System performance looking at what goes right, as well as on what should have gone right. Consequently, performance variability is both normal and necessary and is the source of positive and negative outcomes – successes and failures – alike. While some adverse events can be attributed to component malfunctions such as automation degradations, others arise from unexpected combinations and/or non-linear interactions among normal performance variability of everyday performance. In order to prevent negative outcomes, it is needed to identify the situations where normal performance variability may combine to create unwanted effects and to monitor continuously how the system functions in order to intervene and "dampen" performance variability that threatens to get out of control (Hollnagel, 2008). This requires a flexible and iterative approach in order to identify areas where an action is needed for improving the resilience.

Resilience modelling notations should provide support to explicitly represent the STS as a whole taking into account possible expected and unexpected changes. Due to the fact that these changes may impact, in a different way, on one or more elements of the STS, each element shall have its own representation. This helps in identifying areas where an action is needed for improving the resilience. So, the first three requirements deal with monitor activity, elements representation and identification of areas for improving resilience

1) REQ_OM_1: The Socio-Technical System under analysis should be represented explicitly in terms of Socio-Technical System's own performance and possible expected and unexpected changes
2) REQ_OM_2: Each element (human, system and their interrelations) including in the Socio-Technical System under analysis should be represented explicitly
3) REQ_OM_3: Identification of areas where an action is needed for improving the resilience (human, system and their interrelations) should be made explicitly

In addition, the comparison of existing methods and modelling techniques has demonstrated the limited support at design time for explicitly representing the resilience and the potential functional resonance effects. This means that to be useful, such modelling and analysis must be able to handle complex, tightly coupled systems with distributed human and automated control, advanced technology and software-intensive systems, and the organizational and social aspects of systems (Leveson, et al., 2007). The modelling notation should provide support for representing this complexity in terms of non-linear dynamic dependencies and no restricted to causal explanations (Hollnagel, 2014) between the STS elements:

4) REQ_OM_4: Non-linear dynamic dependencies should be represented explicitly
5) REQ_OM_5: These methods and modelling techniques should be able to handle combinations of faults

Most of the existing notations are focused on incident and accident analysis (Hollnagel & Speziali, 2008). This means that they are particularly adapted to analyse, from a high-level point of view, systems that are already deployed and in use. They thus do not particularly provide support for describing in details different point of views, which can be achieved by integrating several modelling methods and techniques. But to be able to model and analyse new systems, techniques that provide support for analysing systems under development are needed, and as different views are required, capability to be integrated with other techniques is also required. And this integration has to guarantee compatibility at model level and also at tool level. These provide additional requirements that should be satisfied by methods and modelling techniques:

6) REQ_OM_6: These methods and modelling techniques should provide support for analysing systems under design and development

As proposed by Hollnagel (Hollnagel, 1998), there is the need of an analytic capability. This can ensure a qualitative and quantitative evaluation of performance variability that is the 8th requirement:

7) REQ_OM_7: These methods and modelling techniques should support qualitative and quantitative performance variability analysis

As previously presented in the state of the art on resilience engineering modelling techniques, many conceptual papers about functional methods and modelling techniques have been published. However, there is still a need of practical tools that support the modelling activities. The tool support ensures the analysis and the integration with other methods and modelling techniques:

8) REQ_OM_8: These modelling techniques and methods should allow the integration with other methods and modelling techniques ensuring compatibility at model and tool level)
9) REQ_OM_9 These modelling techniques and methods should be supported by tool

The identified requirements can guarantee a support for representing and analysing resilience in a complete and systematic way taking into account both the STS as a whole and its elements with their dependencies.

4 Comparison of modelling techniques and associated methods for resilience analysis according to the identified requirements

In this chapter we have offered an overview of the existing modelling techniques and methods for resilience analysis. Moreover, we have described a set of requirements needed for representing in a complete and systematic way organisational aspects through these modelling techniques and methods. In this section we make a comparison of modelling techniques and methods for resilience analysis according to the identified requirements.

Indeed, the following table reports in the rows the described requirements and in the columns the modelling techniques and methods. For all requirements, there are three possible values:

- "Yes" (green box with YES) means that that characteristic is explicitly handled by the method or techniques,
- "No" (white box with No) means that the characteristic is not explicitly handled,
- "Some" (grey box) means that the property is partly present.

TABLE 5: COMPARISON OF MODELLING TECHNIQUES AND ASSOCIATED METHODS FOR RESILIENCE ANALYSIS ACCORDING TO SUGGESTED REQUIREMENTS

	Requirements /Notations	FTA	FMECA	SCM (TRIPOD)	RCA	MTO	CREAM	AEB	HAZOP	HERA	STAMP	ARKTRANS	FRAM
Data	**REQ_OM_1.** Representation of the STS in terms of STS's own performance and changes	No	No	No	No	No							Yes
	REQ_OM_2. Representation of each element including in the STS	No	Yes	Yes	Yes	Yes	Yes	Yes	No		Yes	Yes	Yes
Analysis	**REQ_OM_3**. Identification of a specific area where an action is needed for improving the resilience	No	No	No	No	No	No	No	No	No	No		Yes
	REQ_OM_4. Representation of non-linear dynamic dependencies	No			No	No			No				Yes
	REQ_OM_5. Handling combinations of faults				No	No			No	No	No		Yes
	REQ_OM_6. Predictive capability	No	No		No	No			No		No	No	
	REQ_OM_7. Analytic capability of performance variability qualitatively and quantitatively		No		No	No	No	No	No	No			
Integration	**REQ_OM_8.** Integration with other methods and modelling techniques ensuring compatibility at model and tool level		No	No		No	No	No	No		No	Yes	No
Tool support	**REQ_OM_9**. Tool support	CAFTA/SAPHIRE/rISKspECTRUM	Table-based	No tool support	TapRoot®	No tool support	No tool support	No tool support	Table-based	HERA-PREDICT/OBSERVE/SMART	No tool support	Enterprise Architect	No tool support

Yes	The modelling technique handles with the requirement
No	The modelling technique doesn't handle with the requirement
	The modelling technique partially handles with the requirement

As shown in the table, while all of the reported methods and modelling techniques fulfil representation elements requirements (REQ_OM_2), only few of them fulfil the other requirements such as the ability to monitor the STS and the identification of areas where an action is needed for improving the resilience.

Moreover, due to the fact that most of them are focused on accident and incident analysis, they are just able to represent causal relationship instead of non-linear dynamic dependencies. FTA and FMECA are able to handle combinations of faults. FTA is a deductive, top-down method aimed at analysing the effects of initiating faults and events on a complex system. This contrasts with FMECA, which is an inductive, bottom-up analysis method aimed at analysing the effects of single component or function failure on equipment or subsystem. FTA provides support for showing how resistant a system is to single or multiple initiating faults. It does not provide support for finding all possible initiating faults. FMECA provides support for exhaustively cataloguing initiating faults, and identifying their local effects. It does not provide support for examining multiple failures or their effects at a system level.

Few methods and modelling techniques are tool supported. In the resilience engineering community, there have been several tentatives to develop a FRAM computer aided software engineering tool but no usable and functional one is available yet. This implies that most of the presented methods and modelling techniques can be integrated at the model level but not at the tool level.

As reported in the table, FRAM is the method which satisfies most of the requirements. But it needs to be integrated with other methods in order to move from an incident analysis perspective to a preventive and proactive approach. Furthermore, at the current stage of development of FRAM, the variability is characterised qualitatively, i.e., as a quality rather than quantitatively.

5 Conclusion, lessons learnt and way forward

In this chapter we have provided an overview of the current state of the art in resilience analysis and modelling to support the analysis and assessment of usability and resilience of partly autonomous interactive systems. The state of the art section has demonstrated that existing modelling techniques and methods provide limited support for explicitly analysing the resilience of a Socio-Technical System and, in particular, the functional variability of a partly autonomous interactive system. In order to overcome the highlighted limitations, a set of requirements has been proposed.

The identified requirements aim at providing support for representing in a complete and systematic way organisational aspects of a STS. According to these requirements, a comparison of the exiting modelling techniques and methods has been made. This has highlighted that most of these techniques are focused on accident and incident analysis and they are not able to deal with predictive capability and variability analysis.

The purpose of the required representation elements is to achieve resilience analysis, not only for accident and incident analysis through causal relationships, but also for developing partly-autonomous interactive systems within a proactive approach encompassing non-linear dynamic dependencies.

Moreover, the comparison allowed us in selecting one of the presented methods and modelling techniques: FRAM. Despite it does not satisfy all the identified requirements, it is promising as a basis due to several reasons, as it provides support for:

- representing each element of the Socio-Technical System under analysis;

- monitoring the Socio-Technical System under analysis in terms of Socio-Technical System's own performance and possible expected and unexpected changes;
- identifying areas where an action is needed for improving resilience (human, system and both);
- representing non-linear dynamic dependencies in a proactive approach;
- analysing performance variability (qualitatively at the moment)

In addition, this method can be extended and integrated with the others modelling techniques that have been selected in the previous chapters in order to analyse and model a partly autonomous interactive system within a STS. This multi-models based approach is the second contribution of this PhD thesis and is presented in Chapter 5.

PART II – Contributions

Chapter 4 – The HAMSTERS extensions

This chapter presents how the HAMSTERS task modelling notation has been extended in order to provide support for complete and systematic description of required knowledge when a user interacts with a partly autonomous interactive system. According to the state of the art and the comparison between several task modelling techniques and methods presented in Chapter 1, we have selected HAMSTERS version 1.0. This notation can be used as a basis but the proposed extensions are also compatible with other procedural-centred notations such as CTT for instance.

As explained in Chapter 1 Section 3, what is still missing in task modelling techniques is the representation of the various types of data manipulated by the users such as information, objects, and knowledge of several types that are needed by a user to achieve a task while interacting with a partly autonomous interactive system. These data have been systematically taken into account with a set of extensions to the HAMSTERS notation. These extensions provide support for analysing system usability properties such as learning curve, task complexity, and information workload. We demonstrate the application of the approach on a simple but meaningful example of a simple two player's game which is used to highlight the connection between the task models produced using the extended version of the notation and the user interfaces of the game.

The HAMSTERS notation is supported by a software tool which enabled to create, edit and modify task models.

This extended version of the notation is named HAMSTERS 2.0 (version 2.0). The scope of this chapter is to:

- Introduce all the proposed extensions regarding to information, object and knowledge representation according to the identified requirements described in Chapter 1 – Section 3
- Explain how the use of concept map can support the description and structuring of the information within a task modelling technique
- Provide an illustrative example in order to highlight the HAMSTERS extensions.

The first section describes HAMSTERS 2.0 and, in particular, all the extensions that have been added to enable the representation of knowledge and information required to perform a task.

The second section illustrates how the use of concept maps can support the description and structuring of the information within a task modelling technique. Moreover, it suggests some guidelines and recommended terminology to build a concept map associated to a task model.

The third section shows how these extensions can be used in practice through an illustrative but meaningful example.

1 Representation of Knowledge and Information inside HAMSTERS 2.0 task models

This section presents the extensions that have been added to the HAMSTERS notation in order to allow the representation of elements of declarative knowledge and in order to enable their integration within the procedural knowledge description of HAMSTERS 2.0. The main changes with regard to

HAMSTERS 1.0 correspond to the explicit handling of declarative and procedural knowledge elements within a concept map and to the explicit representation of these elements in the task model.

Declarative knowledge, according to the identified requirements (please refer to Chapter 1 Section 3), is represented by a "DK" label followed by a violet rectangle or box containing the title of the knowledge (Figure 14 a)). It can be refined in strategic (Figure 14 b)) or situational knowledge (Figure 14 c)). Relationships between the represented knowledge and the tasks can be described using input/output relationships (represented with arcs as for the objects in Figure 4).

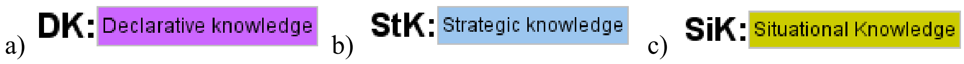


Figure 14: Representation of declarative knowledge a) which can be further refined into strategic b) and situational c)

Concepts which are of information type can also be represented in a non-ambiguous way using the Information box illustrated in Figure 15. As for an object, it is possible to represent the relationships between a task and information (input, output and input-output, as illustrated in Figure 4).

Figure 15: Representation of a concept of information type

Procedural knowledge is already partly described in task models, as a task model describes a procedure required to be accomplish in order to reach a goal. However, according to the identified requirements (please refer to Chapter 1 Section 3), extensions are required to explicitly distinguish strategic and situational procedural knowledge**.** Representative distinctions between strategic and situational procedure can be made using two new types of arcs illustrated in Figure 16. An ordered set of actions related to a strategy the user can apply will be highlighted with the blue "St" tagged arcs (Figure 16a)). An ordered set of actions the user can execute in a given situation will be highlighted with the green "Si" tagged arcs (Figure 16b)).

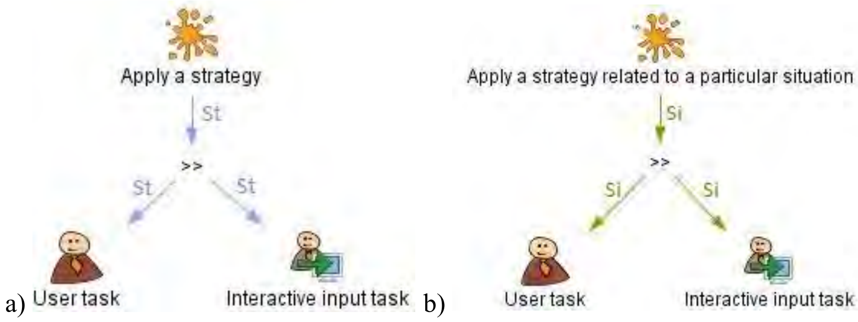


Figure 16: Representation of procedural knowledge refined into strategic a) and situational b)

2 Representation of Knowledge, Information and Objects linked to HAMSTERS 2.0 task models

As presented in the previous section, HAMSTERS 2.0 provides support to describe various data types, such as knowledge, required to accomplish tasks. However, in task models, data is linked to tasks and the HAMSTERS task model view does not provide a unified description of all the concepts (information, objects, knowledge and their relationships). This kind of representation can be achieved through the building of concept maps. A concept provides a mean for understanding the world and is a mental organization of knowledge. Despite the many differences among theories of knowledge organization (please refer to Chapter 1 Section 3), they share a fundamental assumption that knowledge can be modelled in terms of a set of components and their relationships. Concept mapping is a method for externalizing such a structure in an individual, making concepts and relationships explicit (Leake, Maguitman, & Reichherzer, 2004). Moreover, they can be used either to represent knowledge or to support automated systems for reasoning about the represented knowledge.

To build a concept map, the following steps are required (Novak & Cañas, 2008):

1. Create a context: identify a segment of a text, a laboratory or field activity, or a particular problem or question that one is trying to understand. This creates a context that will help to determine the hierarchical structure of the concept map. It is also helpful to select a limited domain of knowledge for the first concept maps.
   a. A good way to define the context for a concept map is to construct a Focus Question, that is, a question that clearly specifies the problem or issue the concept map should help to resolve.
2. Identify the key concepts: these could be listed, and then from this list a rank ordered list should be established from the most general, most inclusive concept, for this particular problem or situation at the top of the list, to the most specific, least general concept at the bottom of the list. This list is called parking lot.
3. Construct a preliminary concept map by writing all of the concepts or using a computer software program. It is important to recognize that a concept map is never finished. After a preliminary map is constructed, it is always necessary to revise this map. Other concepts can be added. Good maps usually result from three to many revisions.
4. Once the preliminary map is built, cross-links should be sought: these are links between concepts in different segments or domains of knowledge on the map that help to illustrate how these domains are related to one another.

According to the four steps of the guidelines for building a concept map, once created the context there is the need to identify the key concepts. In particular, we propose a recommended terminology to build a concept map for representing concepts related to the use of a partly autonomous interactive system. To identify these concepts and their definition, we browsed several documents and repositories, we performed an extensive review of the literature, and we discussed them with subject matter experts in both domains, ATM and HMI (please refer to PART I – State Of the Art). At the end of these iterations, we ended up with a selection of five key concepts that are:

- **Strategy**: a usual or customary action or proceeding. (They could be best practices or procedures which can be adopted. They define the actions which can be performed by a person)
- **Rule**: an authoritative regulation or direction concerning method or procedure, as for a court of law, legislative body, game, or other human institution or activity. (They define how to apply a best practices or a strategy)
- **Role**: the part played by a person in a particular situation. (Different roles can be played by different persons that are defined actors. An actor performs a strategy conforming to rule and taking into account the available objects)
- **Object**: a tangible and visible thing (that towards which cognition is directed, as contrasted with the thinking subject; anything regarded as external to the mind, in the external world)
- **Device**: a machine or tool used for a specific task. A computer hardware that is designed for a specific function. When an actor is performing a specific task, s/he could provide an input to the device which produces an output. This input should be provided through an interface.
- **Domain:** a field or scope of knowledge or activity.

According to the guidelines, these are the most general, most inclusive concepts which can be further detailed into more specific and less general concepts for a particular situation.

Albeit with some variations, the same concepts have been applied to the illustrative case study which is presented in the following section.

3 An illustrative example: The Game of 15

To illustrate how the HAMSTERS extensions data can be used to represent information, objects and knowledge in a systematic way during task modelling activities, we propose a simple but meaningful example.

The Game of 15 is a two player traditional game. Each player takes his/her turn to choose a remaining number (a token) ranging from 1 to 9. The first player whose sum of selected number is exactly 15 wins the game. In this example, we analyse the tasks performed by the players using the interactive application depicted in Figure 17.

Figure 17: Example of a user interface for playing Game of 15

3.1 Task models of Game of 15

This section presents task models of the Game of 15 and illustrates the HAMSTERS extensions. This illustrative example highlights the integration of both procedural and declarative knowledge in HAMSTERS 2.0 task models as well as the integration of the representation of information and objects required to perform the tasks.

Figure 18 and Figure 19 present HAMSTERS 2.0 models of tasks for playing Game of 15 using the interactive application presented in Figure 17. Indeed, Figure 18: HAMSTERS 2.0 model of Game of 15 – main task to play the game presents the main HAMSTERS 2.0 model with top-level task named "Play Game of 15". The hierarchical and temporally ordered sub-trees detail in a procedural way the various tasks that have to be known and performed by the players. First they have to agree on who will start the game. Then they play the game in an iterative way (round-shaped arc at the top right of the "Play the game" abstract task) until they commonly agree to stop (if one player won or if there are no token left or if they just want to stop). The "Play the game" sub-task is a sequence of the following activities: turn taking (for one player), selecting a token (for current player, this task is a subroutine described in detail in Figure 19), processing combination of tokens (for each player), and giving turn to the other player.

In Figure 18, procedural knowledge required to play the Game of 15 is described via hierarchical decomposition and temporally ordering of nodes. Declarative knowledge (refined in objects and rules in the concept map in the next section) is represented by mean of violet boxes connected to particular tasks with input arcs. For example, in Figure 18, the declarative knowledge "One token can be taken once "is required to be able to perform the task "Select a token". In the same way, information required to perform tasks, and/or produced when the task is complete, are depicted by mean of orange boxes connected to particular tasks with input and/or output arcs. An input/output arc between an information and a task means that the task is consuming and modifying the information. For example, in Figure 18, the information "One of the player wins" is required to perform the collaborative task for agreeing that "One of the player won" (input arc from the information to the task) and the tasks "Player 1 processes combination of tokens" produces the information "One of the player wins" (output arc from the task to the information)

In Figure 19, procedural knowledge required to accomplish the "Select of token" subroutine is also described via hierarchical decomposition and temporally ordering of nodes. It describes in a hierarchical way the activities one player has to perform for selecting a token. The sequence of tasks to be performed is the following: the interactive application interface is first displaying remaining tokens. The player then reads the set of available numbers, has to remember existing strategies and to adopt a particular strategy to select a number (the sub-trees detail possible choices of strategies which may be chosen according to the selected strategy). Once the player has selected a token, s/he inputs the number in the interface and the interface displays the token that has been selected. . Declarative knowledge (further detailed in the next section through the concept map) is represented by mean of violet boxes connected to particular tasks with input arcs are they described required knowledge to accomplish a task. Strategic knowledge is represented both:

- in a procedural way using blue arcs in sub-trees dedicated to strategic knowledge below "Adopt a Strategy" task
- and in a declarative way blue boxes labelled StK (for **St**rategic **K**nowledge) connected to tasks related to particular strategies, also coming from the concept strategies in the concept map (please refer to the next section)

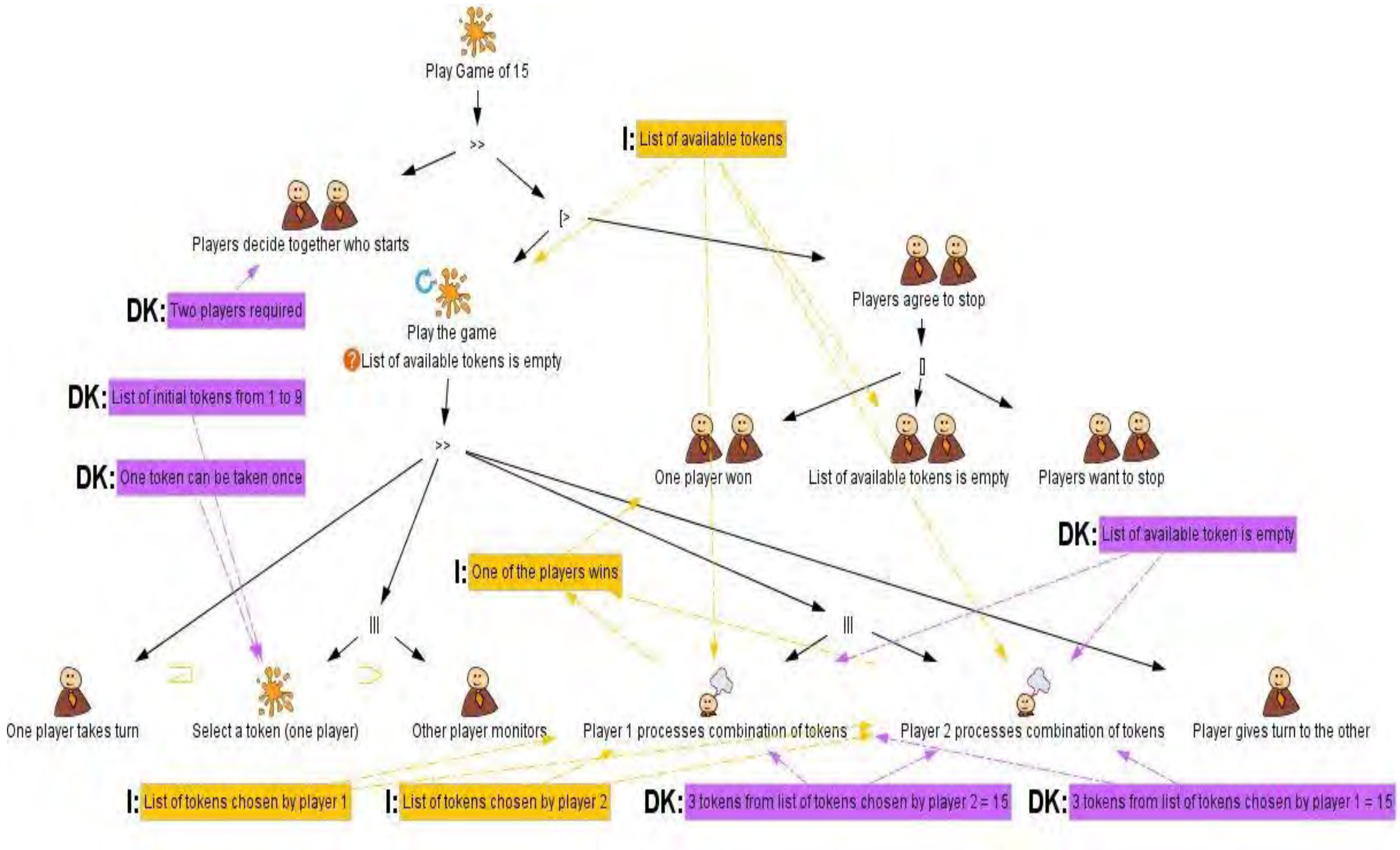


Figure 18: HAMSTERS 2.0 model of Game of 15 – main task to play the game

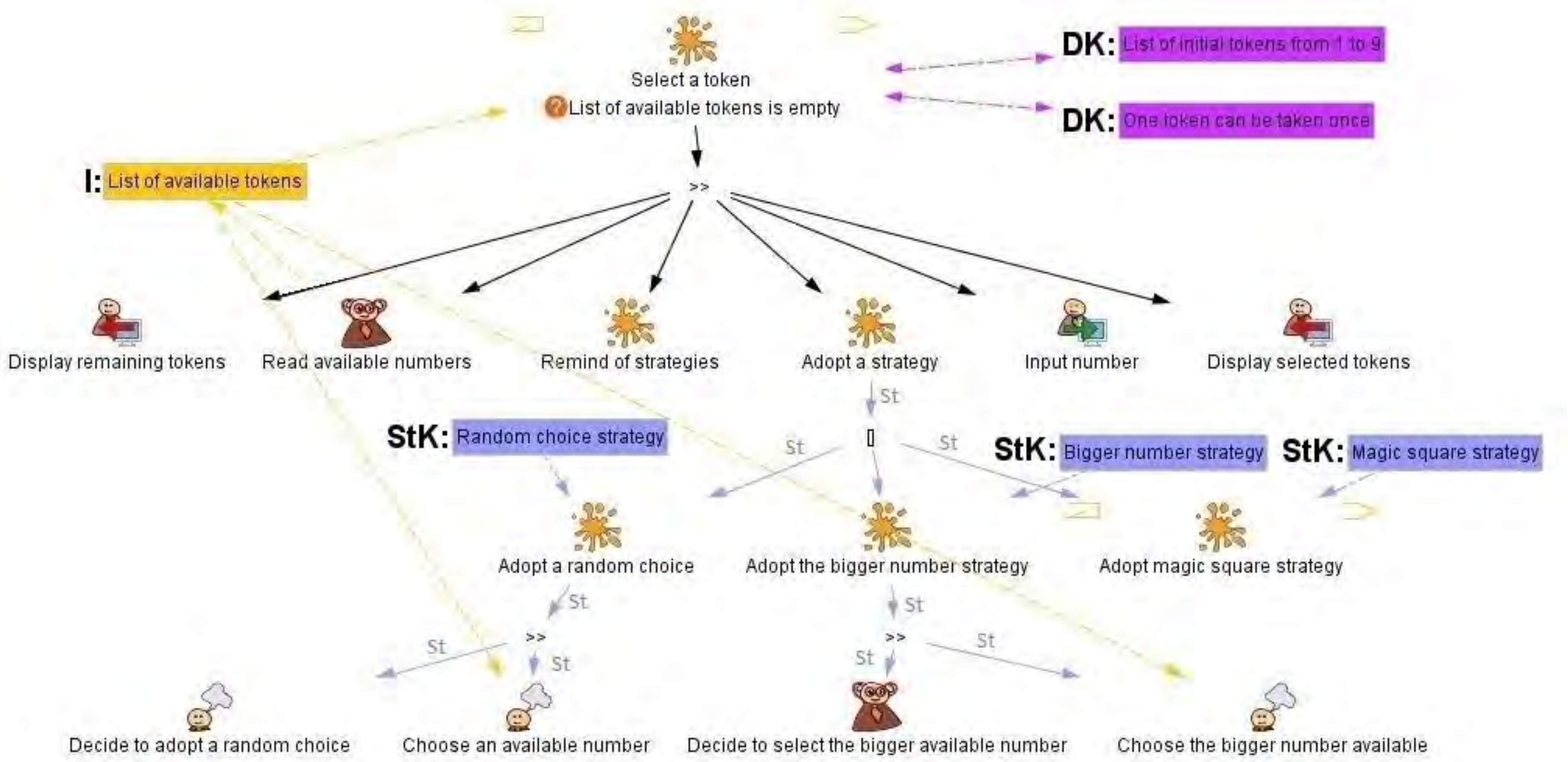


Figure 19: HAMSTERS 2.0 model of Game of 15- subroutine corresponding to the task Select token in Figure 18

Required information is depicted by mean of orange boxes connected to particular tasks. For example, in Figure 19, the information "List of available tokens" is required to perform the task "Choose an available number" (input arc from the information to the task) and the task "Read available number" produces the information "List of available tokens" (output arc from the task to the information).

Example of declarative situational knowledge description is not depicted in this example but would be represented in the same way as strategic knowledge (with corresponding colour and arc label as described in Figure 16b).

3.2 Concept map of Game of 15

This section presents the concept map for the Game of 15 interactive application and illustrates the connections between the task models and the concept map.

According to the extensions associated to HAMSTERS 2.0 presented in Chapter 4 – Section 2, Figure 20 illustrates the concept map associated to HAMSTERS 2.0 task models for representing knowledge, information and objects related to the Game of 15.

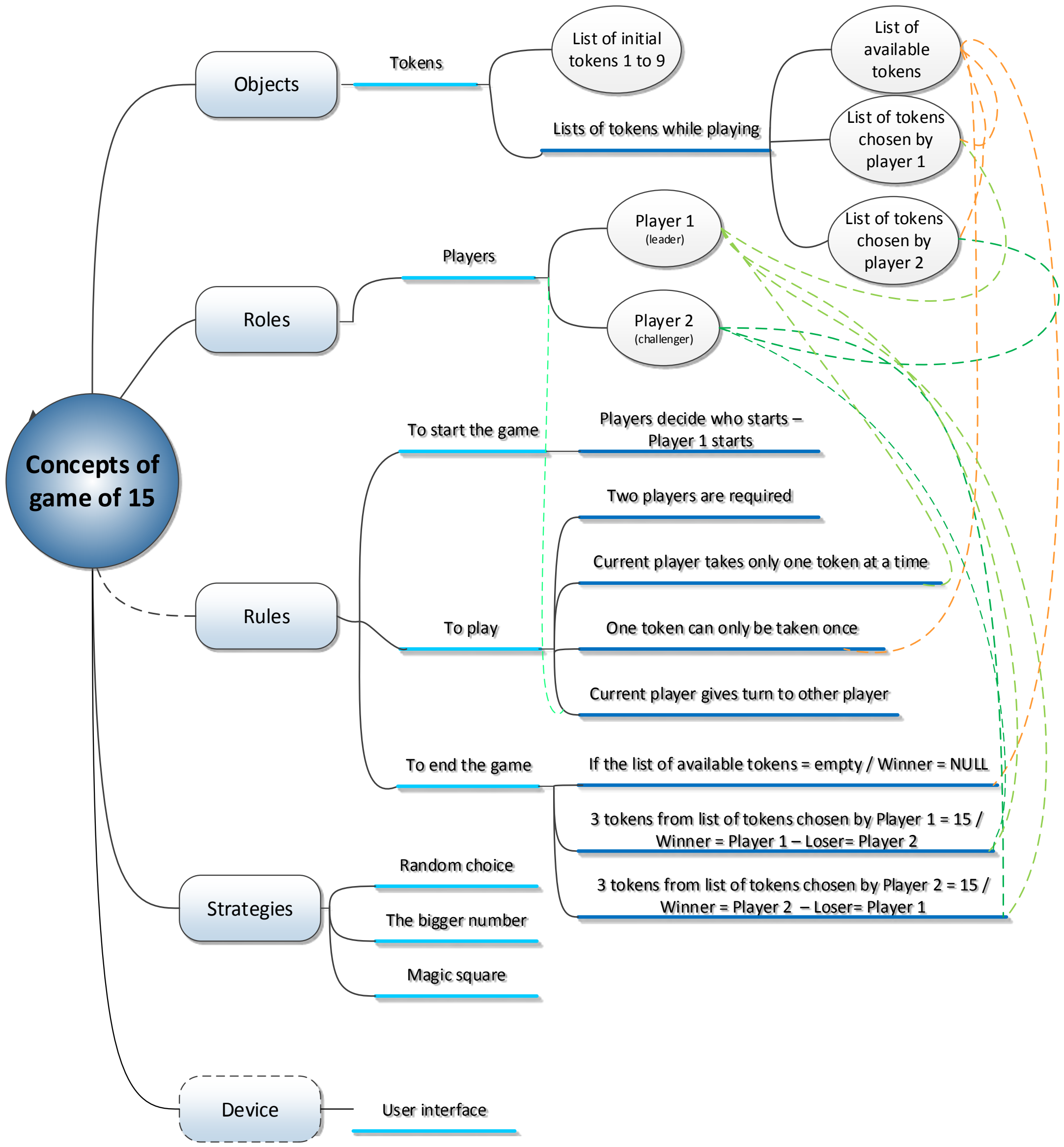


Figure 20: Concept map of Game of 15

"Figure 20: Concept map of Game of 15" represents required declarative knowledge necessary to be able to play the game. The concepts which are necessary or useful to perform actions during the game are grouped and linked through a semantic network. It is made up of 5 key concepts or nodes using the recommended terminology explained in Section 2 of Chapter 4. These key concepts are Objects, Roles, Rules, Strategies and Device. In our illustrative example, these concepts are further detailed and applied to the specific context of the game of 15. Structure from left to right shows the refinement of concepts, from abstract to concrete, as well as the instantiation. For example, the concept of "Players" is instantiated twice: the "Player 1 (Leader)" instance and the "Player 2 (challenger)" instance. Lastly, relationships between refined concepts can also be represented in an explicit way, using links. For instance, the link between the concept "List of available tokens" and the concept "One token can only be taken once" makes explicit the relationship between the selection of a token and the fact that it won't be able anymore in the list of available tokens.

All the represented concepts should have a corresponding representation in one or several models produced with the task, system or organisation modelling techniques. To point out this correspondence between the Game of 15 concept map and HAMSTERS 2.0, we can observe that the declarative knowledge violet box "List of initial tokens from 1 to 9" in "Figure 19: HAMSTERS 2.0 model of Game of 15- subroutine corresponding to the task Select token in Figure 18" corresponds to the bubble with the same label in Figure 20: Concept map of Game of 15".

These correspondences enable to ensure consistency between the different views needed to properly model and analyse a partly autonomous interactive system. They are further refined in the next chapter, in which is presented the second contribution of this PhD thesis: the multi-models based approach.

4 Conclusion

This illustrative example demonstrates that the integration of both declarative and procedural knowledge in a task model provides additional useful information for making explicit the data (knowledge, information and object) required for reaching a goal, as well as the data produced while executing tasks.

These HAMSTERS extensions used in a synergistic way with concept maps can provide useful support in the various phases of the development process of an interactive system (provided that it explicitly refers to a task analysis and task modelling phases such as the one presented in (Martinie, Palanque, Navarre, & Barboni, 2012)):

- Requirements elicitation by identifying what the users' needs in terms of knowledge, information and objects
- Design of the interactive system by making explicit the information that has to be provided to the user and the type of input that the users will need to perform
- Implementation of the interactive system by providing an informal but complete description of the data that the system will have to embed in order to allow performance of the user tasks
- Evaluation of the interactive system by providing an explicit representation of the information that will be manipulated and thus identifying candidate values for user testing.
- Training of the future operators of the interactive system by making explicit what knowledge they have to master in order to use the system.

This systematic account for knowledge, information and objects required to perform tasks provides means for analysts to verify the compatibility and completeness of the concept map with respect to the task model and vice versa. For instance one could check that all the concepts in the concept map appear in the task model and are graphically visible on the user interface. For example, the presented case study, shows that all the knowledge, information and objects dealing with the concept of tokens such as: list of available tokens, tokens selected by each player... are present in both task models and concept map.

This work is targeting partly autonomous interactive systems where analysis about knowledge and information processing is of prime importance. For instance workload analysis has to be performed and decisions around tasks migration and automation can be fruitfully informed by a more complete account of declarative knowledge involved in tasks performance.

All the extensions presented to the original HAMSTERS 1.0 notation have been implemented in HAMSTERS 2.0 tool which allows both editing and simulating task models. Moreover, we have also demonstrated the possibility to connect task models with behavioural descriptions of user interfaces making co-execution and validation of models interactive.

As shown in the following table, HAMSTERS 2.0 is compared with the other modelling techniques and its previous version according to the identified requirements. For all requirements, there are six possible values:

- “Yes” (green box with Yes) means that the task modelling technique satisfies the corresponding requirement,
- “No” (white box with No) means that the task modelling technique does not satisfy the requirement,
- “Partially” (grey box) indicates that the task modelling technique partially satisfies the requirements i.e. it does not deal with knowledge and information
- “Partially with objects” (grey box with P-O) indicates that the task modelling technique partially satisfies the requirements i.e. it does not deal with knowledge and information,
- “Through precondition” or “Through constraint” (grey box with T-P or T-C respectively) indicate that the task modelling technique has attempted to integrate all the requirements via precondition or constraint,
- “Through concept map” (grey box with TCM) indicates that the task modelling technique has been integrated with concept maps

TABLE 6: COMPARISON OF TASK MODELLING NOTATIONS AND HAMSTERS 2.0

	Requirements /Notations	HTA	GOMS	TKS	MAD	UAN	GTA	Diane+	VTML	CWA	TAKD	CTT	AMBOSS	SAMANTA	HAMSTERS 1.0	**HAMSTERS 2.0**	**THESIS CONTRIBUTION**
Declarative Knowledge (DK)	**REQ_TM_1.** Concepts used to represent DK	No	No	Yes	No	No	No	No	No	Yes		No	No	Yes	No	Yes	
	REQ_TM_2. Refinement of concepts into objects and information	No	No	No	P-O (explicit)	No	No	No	No	No	P-O	P-O	No	P-O	P-O	P-O	
	REQ_TM_3. Concepts connections should be made explicit	No	No	Yes	No	No	No	No	No	Yes	No	No	No	Yes	No	TCM	
	REQ_TM_4. Semantic networks to represent the concepts and their relationships	No	No	No	No	No	No	No	No	No	No	No	No	Yes	No	TCM	
	REQ_TM_5. Refinement into strategic and situational knowledge	No	No	No	No	No	No	No	No	No	No	No	No	No	No	Yes	
Procedural Knowledge (PK)	**REQ_TM_6.** Routines and subroutines should be used to represent explicitly PK	No	No	No	No	No	No	No	No	No	No	No	No	No	Yes	Yes	
	REQ_TM_7. Routines and subroutines relationships should be made explicit in order to structure and group PK	No	No	No	No	No	No	No	No	No	No	No	No	No	Yes	Yes	
	REQ_TM_8. Hierarchical structures to represent routines, sub-routines and their interrelationships	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	No	No	Yes	Yes	Yes	Yes	Yes	
	REQ_TM_9. Refinement into strategic and situational knowledge	No	No	No	No	No	No	No	No	No	No	No	No	No	No	Yes	
Integration	**REQ_TM_10.** All the information above should be integrated in a single model	No	TP	TP	TP	No	TP	TP	TP	TC	TP	TP	TP	TP	TP	TP	
	REQ_TM_11. Integration with other methods and modelling techniques ensuring compatibility at model and tool level	No	No		No	No	No	No	No	No			No		Yes	Yes	
Tool support	**REQ_TM_12.** Supported by a tool	Task-Architect	CAT-HCI and others	ADEPT	K-MADe	Quantum	GTA and EUTERPE	TAMOT and Isolde	SpecTRM tool Suite	(No tool support)	LUTAKD	CTTe	AMBOSS	SAMANTA	HAMSTERS 1.0	**HAMSTERS 2.0**	

As reported in the table, HAMSTERS 2.0 provides support for:

- Representing explicitly declarative knowledge
- Describing all the knowledge, information and objects required while interacting with system and the relationships between these different types of data through concept maps.
- Refining the description of both declarative and procedural knowledge into strategic and into situational knowledge

The current chapter has presented how all these extensions have been used in practice through an illustrative but meaningful example. The same illustrative example is used in the next chapter for explaining the second contribution of this PhD thesis which is a process for analysing and modelling a partly autonomous interactive system within a STS. This process relies on the integration of HAMSTERS 2.0 with the other selected modelling techniques and methods which are ICO and FRAM.

Chapter 5 – The multi-models based approach for modelling and analysing a partly autonomous interactive system within Socio-Technical Systems

This chapter presents the process for integrating the three selected modelling techniques in order to model and to analyse a partly autonomous interactive system within a Socio-Technical System (STS) and to assess its resilience and usability properties. Furthermore, this approach provides support for examining the system under analysis at different levels of granularity. In the state of the art, the comparison between several modelling techniques and methods helped us to select the most suitable modelling techniques for carrying out this whole representation. These are: HAMSTERS version 2.0 (indeed we adopt the extended version presented in Chapter 4), the Interactive Cooperative Objects (ICOs) and its associated CASE tool PetShop (presented in Chapter 2 – Section 2.2) and FRAM (described in Chapter 3 – Section 2.2).

In addition, as explained in each chapter of Section 3, a common identified requirement for each selected modelling technique is the support for integration with other modelling techniques and associated methods. The full integration, at model and tool level, is currently available for two of the selected techniques which are HAMSTERS 2.0 and ICO (their synergistic use is described in Chapter 2 – Section 5). The integration with the third method, FRAM, has been done at the model level. This integration allows us to represent all the parts of a partly autonomous interactive system within a Socio-Technical System (STS) and, through this multi-models based approach, it provides support for modelling and analysing the STS under consideration as a whole.

The scope of this chapter is to:

- Introduce and describe the process and its main phases for modelling and analysing a partly autonomous interactive system within a Socio-Technical System.
- Provide an illustrative example in order to demonstrate step by step the proposed approach.

The first section describes all the phases of the approach for modelling and analysing a partly autonomous interactive system within its Socio-Technical System. In particular, it provides a detailed description of the modelling and the analysis phase. Regarding the modelling phase, the specific role and connections of each selected modelling techniques and how they integrate together are explained. Concerning the analysis phase, the section describes how this process can ensure both qualitative and qualitative variability analysis.

The second section shows how this approach can be used in practice through an illustrative but meaningful example that is the game of 15.

1 The multi-modelS based approach

This section presents the multi-modelS based approach that has been developed in order to integrate the three selected techniques for modelling and analysing a partly autonomous interactive system[17]

[17] Due to space constraints, in the following figures we use the acronym "PAIS" for indicating the Partly Autonomous Interactive System

within a STS. This approach provides support for assessing the resilience and the usability of the system under consideration and at different levels of granularity.

Figure 21 illustrates the key of the process diagram. Indeed, each box of the diagram contains the name of the phase (in capital and bold letters), one or several sub-phases (the dotted box(es) inside the phase), and one or more decision blocks.

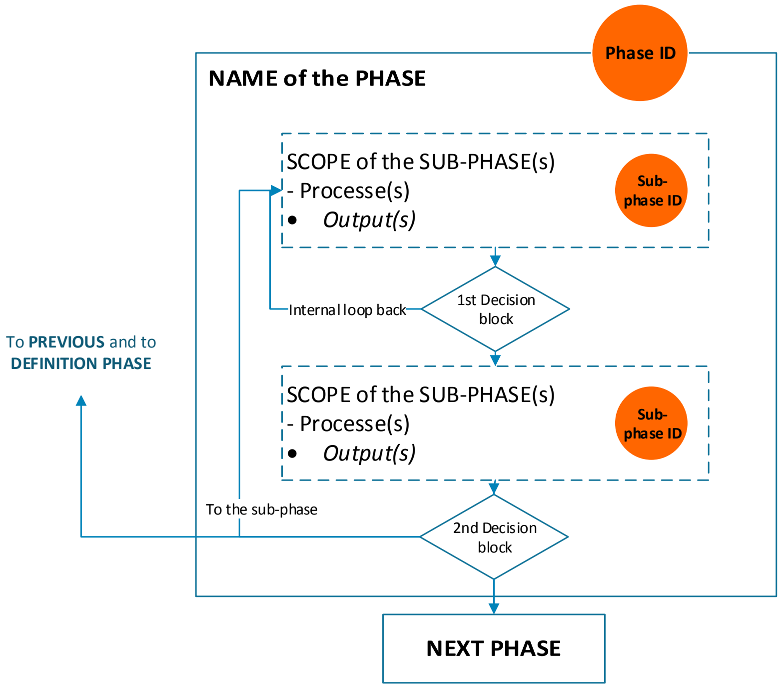


Figure 21: Key of the process diagram

For each sub-phase, the scope is written in capital letters and is associated to:

- the process(es) needed to achieve the scopes,
  - *the output(s)* that each sub-phase produces.

Moreover, the 1st Decision block (the rhombus) is associated to an internal back loop to the previous sub-phase(s).This back loop mechanism represents the fact that the sub-phases are iterative and that the analyst may re-perform some previous steps if the output of the sub-phase does not match the decision criteria described in the decision block. The 2nd Decision block (the rhombus) is associated to a back loop to the previous sub-phase, to the previous Phase(s) and to the “Definition phase”. For each decision block, if the output incoming from the previous sub-phase match the decision criteria, the analyst can go on to the next phase of the process.

Figure 22 depicts the overview of the proposed process to model and analyse a partly autonomous interactive system within a Socio-Technical System.

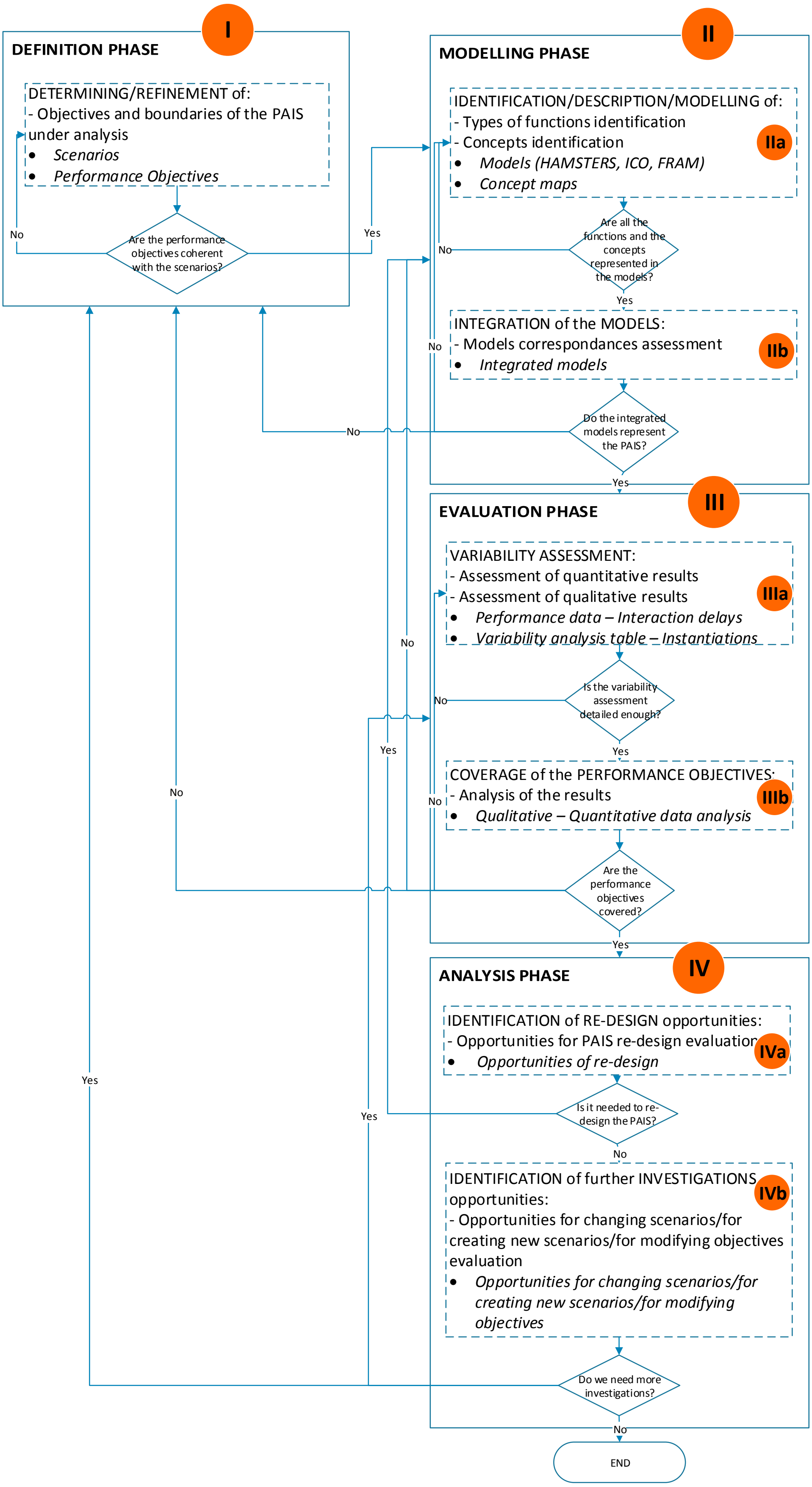


Figure 22: The process to model and analyse a partly autonomous interactive system within a Socio-Technical System

The complete process (for which an overview is presented in Figure 22) consists of four main phases (I, II, III, and IV) and contains a set of steps for modelling and analysing a partly autonomous interactive system within a Socio-Technical System:

1. The I "Definition phase" (detailed in Section 1.1) is the starting point of the process. Its scope is to define/redefine the boundaries and the objectives of the PAIS under analysis. Its output is descriptions of scenarios and of performance objectives. Before proceeding with the next phase ("Modelling phase", labelled II in Figure 22), the analyst has to verify if performance objectives are coherent with scenarios. If they are not coherent ("No" back loop in phase I of Figure 22), the analyst has to redefine them or redefine the scenarios, if they are coherent ("Yes" arrow from phase I to phase II in Figure 22), s/he can carry on to phase II.
2. The II "Modelling phase" (detailed in Section 1.2) can be further decomposed into two sub-phases:
    a. The IIa "Identification/Description/Modelling" sub-phase aims at identifying and modelling human, system and interactive functions as well as their associated concepts (required information, knowledge and objects). The outputs of this sub-phase are the models (of human tasks and functions, of system behaviour and functions, of human-system interactions) and the concept maps (of required information, knowledge and objects). In order to proceed with the next sub-phase ("Integration of the models", labelled IIb in Figure 22), the analyst has to verify if all the functions and the concepts are represented in the models and in the concept maps. If they are not ("No" back loop to the IIa sub-phase in phase II in Figure 22), the analyst has to correct the models and/or the concept maps (IIa sub-phase in Figure 22). If all the functions and concepts are represented ("Yes" arrow from sub-phase IIa to sub-phase IIb in phase II in Figure 22), s/he can carry on to phase IIb.
    b. The IIb "Integration of the Models" sub-phase aims at assessing all the correspondences between models and at integrating the models. The outputs of this sub-phase are the integrated models. In order to proceed with the next phase ("Evaluation phase", labelled III in Figure 22), the analyst has to verify if the models properly represent the PAIS within its Socio-Technical System (according to the scenarios and the objectives of the analysis). If they do not ("No" back loop in Figure 22), the analyst has to understand why the representation is not consistent with scenarios and objectives. The cause of the problem may be a modelling issue, and in this case, models have to be re-worked/modified (sub-phase IIa in Figure 22). Or, if it is needed to redefine the PAIS under analysis (with associated scenarios and/or objectives of the analysis), the analyst has to re-work on/modify the scenarios and/or objectives re-perform ("Definition phase", labelled I in Figure 22). If the integrated models represent the PAIS within the Socio-Technical System ("Yes" arrow from sub-phase IIb to phase III in Figure 22), s/he can carry on to phase III.
3. The III "Evaluation phase" (detailed in Section 1.3) can be further decomposed into two sub-phases:
    a. The IIIa "Variability assessment" sub-phase aims at assessing both qualitative and quantitative results from the analysis of the integrated models. These sub-phase outputs are performance data and variability analysis tables. In order to proceed with

the next sub-phase (“Coverage of the performance objectives”, labelled IIIb in Figure 22), the analyst has to verify if the qualitative and quantitative assessments of performance variability are detailed enough to be analysed. If they are not (“No” back loop to sub-phase IIIa in phase III in Figure 22), the analyst has to re-work on variability assessments (sub-phase IIIa in Figure 22). If variability assessments are detailed enough (“Yes” arrow to sub-phase IIIb in phase III, in Figure 22), s/he can carry on to phase IIIb.

b. The IIIb “Coverage of the performance objectives” sub-phase aims at analysing how many performance objectives are reached by the PAIS and how. This sub-phase output is a coverage estimation. To proceed with the next phase, the analyst has to verify if the performances objectives are covered. If they are not (“No” back loops to phase II and to phase I in Figure 22), the analyst has to understand why. There may be errors in the models and the analyst will have to re-work on them and to modify them (“Modelling phase”, labelled II in Figure 22). Or, the performance objectives may have to be redefined (“Definition phase”, labelled I in Figure 22). If the performance objectives are covered (“Yes” arrow to phase IV in Figure 22), s/he can carry on to phase IV.

4. The IV “Analysis phase” (detailed in Section 1.4) can be further decomposed into two sub-phases:

   a. The IVa “Identification of re-design opportunities” sub-phase aims at taking into consideration opportunities for re-designing the PAIS. If the PAIS needs to be re-designed, the analyst rework on the models (“Modelling phase”, labelled II in Figure 22) for identifying what it is needed to be improved or changed. If is the PAIS does not need to be re-designed, the analyst can proceed with the next sub-phase (“Identification of further investigations opportunities”, labelled IVb in Figure 22).

   b. The IVb “Identification of further investigations opportunities” sub-phase aims at taking into consideration several opportunities with regards to the on-going analysis. The analyst can identify opportunities for changing scenarios, for creating new scenarios, and for modifying performances objectives. If additional investigations are required, the analyst has to re-perform some previous steps. These can belong to the previous “Evaluation phase” (labelled III in Figure 22) if the analyst needs to assess another level(s) of variability. The analyst may also need to define new scenarios and/or to modify the performance objectives (“Definition phase”, labelled I in Figure 22). If further investigations are not required, the process can end.

These phases and sub-phases are ordered in a consecutive way but they can be repeated until the analyst has achieved her/his goal. So, the multi-models based approach is iterative at the generic level as well as at the specific step level. The following sections explain in detail each phase and sub-phase.

1.1 The Definition phase

As illustrated in Figure 23, the “Definition phase” starts by the definition of the purpose and the boundaries of the partly autonomous interactive system to be analysed.

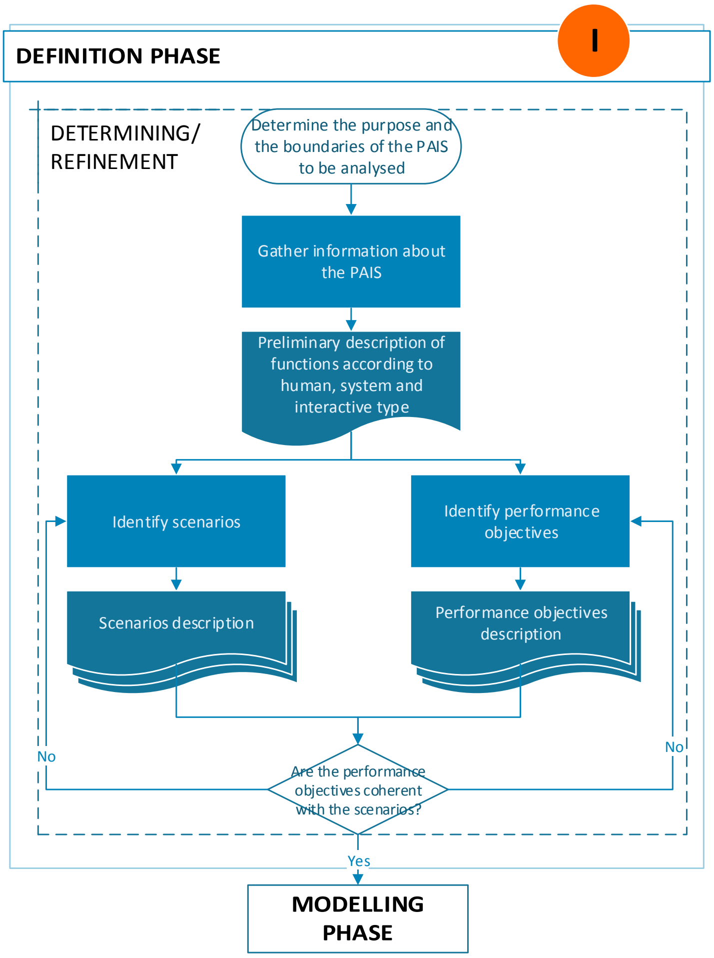


Figure 23: The Definition phase

Then, it is needed to "Gather information about the PAIS" in order to identify all its elements. Once gathered all the information about the PAIS under analysis, these have to be organised according to the different types of elements (human, system and interactive). This produces a "Preliminary description of the functions according to human, system and interactive type". This description gathers the main functions of the PAIS within the STS and is used for:

- "identifying scenarios". As defined by Caroll and Rosson, scenarios "consist of a setting, or situation state, one or more actors with personal motivations, knowledge, and capabilities, and various tools and objects that the actors encounter and manipulate. The scenario describes a sequence of actions and events that lead to an outcome. These actions and events are related in a usage context that includes the goals, plans, and reactions of the people taking part in the episode". Then applied to our approach, a scenario is a sequence of actions (including preliminary high level functions) performed by the various elements of the STS, including the PAIS. It provides detailed information about interactions between each element of the STS. The scenarios usually include plausible, relevant situations and problems that the analyst is interested in investigating. According to Chapter 1 – Section 1 and Chapter 3 – Section 1, a scenario based approach is useful for supporting the analysis and assessment of usability and resilience properties respectively.
- "Identifying performances objectives". According to the purpose of the analysis and the scenarios, related performance objectives can be described. A performance objective is a

specific end result or a behaviour that is expected from one or several elements of the STS.

At the end of the definition phase, the selected scenarios contain the description of the sequences of relevant actions and events in which the functions of the PAIS are used in correlation with the human functions and the interactive functions of the STS. These sequences of events also represent the interrelations between the human, system and interactive functions of the PAIS within the STS. Moreover, the related usability and resilience properties are described in terms of performance objectives.

In order to proceed with the next phase the analyst has to verify if performance objectives are coherent with scenarios. If they are not coherent ("No" back loop in Figure 23), the analyst has to redefine them or redefine the scenarios, if they are coherent ("Yes" arrow from phase I to phase II in Figure 23), s/he can carry on to phase II.

1.2 The Modelling phase

As described in the previous section, once a coherent description of scenarios and performance objectives has been obtained, the PAIS within STS can be modelled. This section explains in detail all the steps of the second main phase of the approach which is the II Modelling phase. This phase can be further decomposed into two sub-phases:

- The IIa "Identification/Description/Modelling" sub-phase
- The IIb "Integration of the Models" sub-phase

Both are detailed in the following sections.

1.2.1 The Identification/Description/Modelling sub-phase

Figure 24 illustrates the IIa "Identification/Description/Modelling" sub-phase which includes several steps that can be performed in parallel.

First, all the functions have to be identified ("Identify functions" step on the left side in Figure 24) and all the information, knowledge and objects required to perform the functions have to be identified ("Identify concepts" step on the right side in Figure 24) respectively. Once all the functions have been identified, they have to be described according to their type. Each scenario, built in the previous phase, can present "Humans", "Systems" and "Interactive" functions. The analyst can "Define a sequence of events" which produces a "Timeline" representing a sequence of related events arranged in chronological order and displayed along a line. According to Chapter 3 – Section 2.2, it is needed to "Characterise potential variability through FRAM". The following steps are performed:

- All the functions are describes as FRAM units with detailing the six basic aspects (Input, Output, Pre-conditions, Resources, Time, and Control) (please refer to Chapter 3 – Section 2.2);
- The endogenous factors are listed. Endogenous source of variability is related to the internal variability of the PAIS within STS (human, system or their interactions) that performs the function (please refer to Chapter 3 – Section 2.2);
- The potential impact of endogenous factors on the functions output variability is described. Variability of output can be described with a set of dimensions such as timing and precision (please refer to Chapter 3 – Section 2.2).

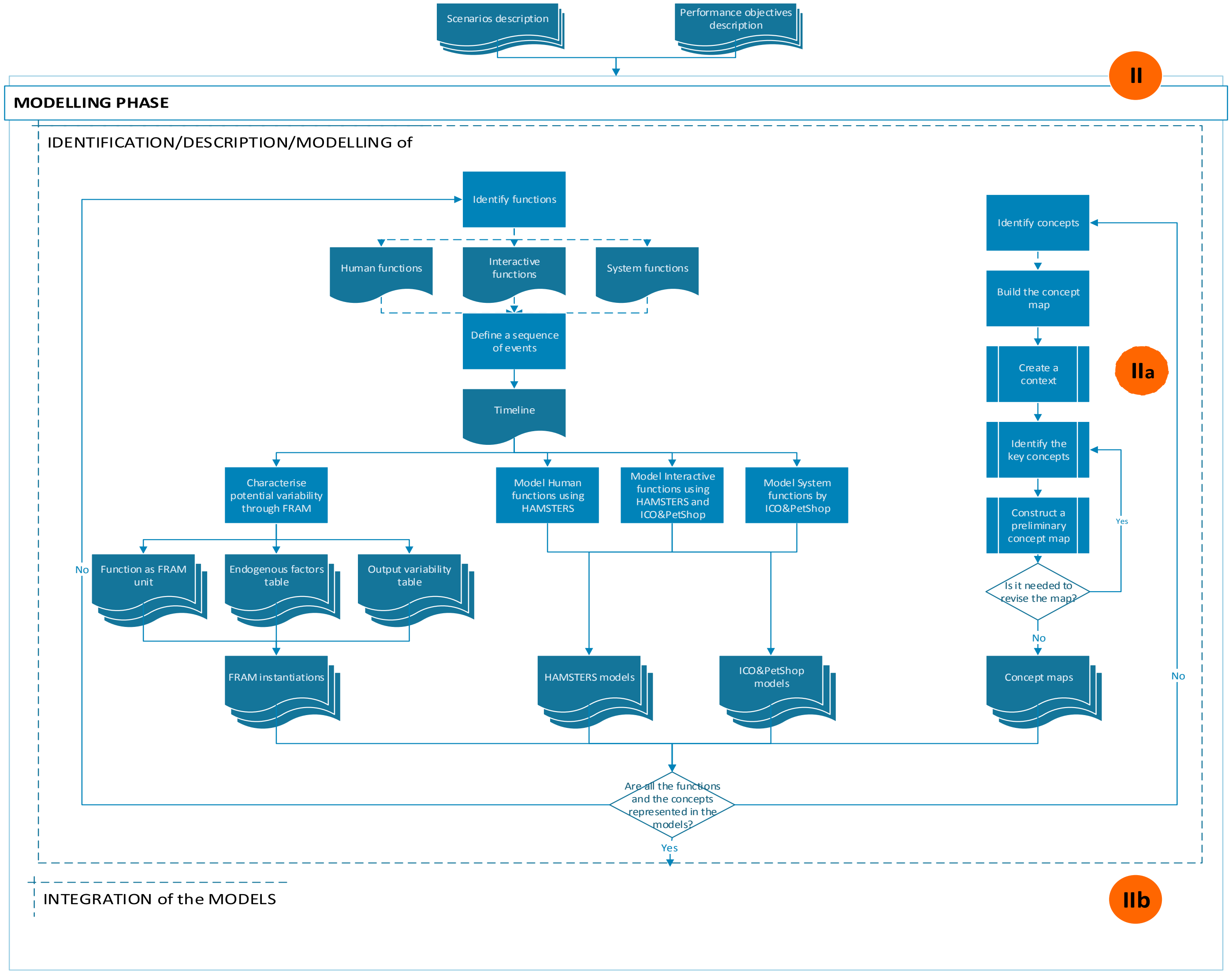


Figure 24: Modelling phase – Identification/description/Modelling sub-phase IIa

The step “characterise the variability” is performed in 2 sub-steps: first the potential variability of the functions described by the model are characterised, and then the potential (possible) actual variability for a set of instantiations of the model is characterised (through a “Variability analysis table” in sub-phase IIIa of the “Evaluation phase” – please refer to Section 1.3 – Sub-section 1.3.1). The produced outputs (FRAM units, endogenous factors table and output variability tables) are needed to create a “FRAM instantiation”. In parallel, for each identified type of function, the analyst will use dedicated modelling techniques for building the related models. According to Chapter 1 – Section 2.2, we selected HAMSTERS for modelling human functions. According to Chapter 2 – Section 2.2, we selected ICO&PetShop for modelling system functions. According to Chapter 2 – Section 5, we selected both modelling techniques for modelling interactive functions. This flow of the sub-phase outputs are “FRAM instantiations”, “HAMSTERS models”, and “ICO&PetShop models”.

In this IIa sub-phase, the other parallel flow is “Identify concepts” (on the right side in Figure 24), which will lead to the production of concept maps. According to Chapter 4 – Section 2, in order to “Build the concept map”, it is needed to accomplish some sub-processes that are: “Create a context”, “Identify the key concepts” and “Construct a preliminary concept map”. Then, to obtain a final version of the “Concept maps”, the analyst will iteratively perform the sub-processes described here above in order to revise and refine the preliminary versions. The “Concept maps” are the output of this flow of the sub-phase. As demonstrated in PART II – Contributions - Chapter 4 , theoretically in Section 2 and practically in Section 3, concept maps provides a unified description of all the concepts (information, objects, knowledge and their relationships) and they can be used either to represent knowledge and/or to support automated systems for reasoning about the represented knowledge.

In order to proceed with the next sub-phase (“Integration of the models”, labelled IIb in Figure 24), the analyst has to check whether or not all the functions and all the concepts are represented in the models. If all of them are not represented (“No” back loop in Figure 24), the analyst has to re-perform some previous steps. Hence, s/he has to rework on identifying and describing functions and concepts (flow of steps starting by “Identify functions” and/or flow of steps starting by “Identify concepts” in Figure 24). Once all the functions and the concepts are represented in the models, s/he can carry on to the next sub-phase.

1.2.2 The Integration of the Models sub-phase

Figure 25 illustrates the IIb “Integration of the Models” sub-phase which includes several steps that can be performed in parallel. Once a complete representation of all the functions has been obtained through the three modelling techniques (FRAM, HAMSTERS and ICO&PetShop) and once a complete representation of the concepts has been obtained through the concept maps, the correspondences between models have to be assessed (“Assess the correspondences between models at tool level” on the left side in Figure 25and “Assess the correspondences between models at model level using FRAM as correspondence editor” on the right side in Figure 25) respectively.

As explained in Chapter 2 – Section 5, HAMSTERS and ICO&PetShop models can be integrated both at model and at tool level. This means that a variation of the system model that impacts a human task will trigger a change in the corresponding task model and vice versa. This helps in iteratively refining models of both types until all the corresponding models are consistent and coherent between each other. In addition, the integration of the system and task models with the FRAM instantiations, at the model level, provides a complementary view for assessing consistency and coherence between system and task models, as it enables to verify the sequence of function calls between models.

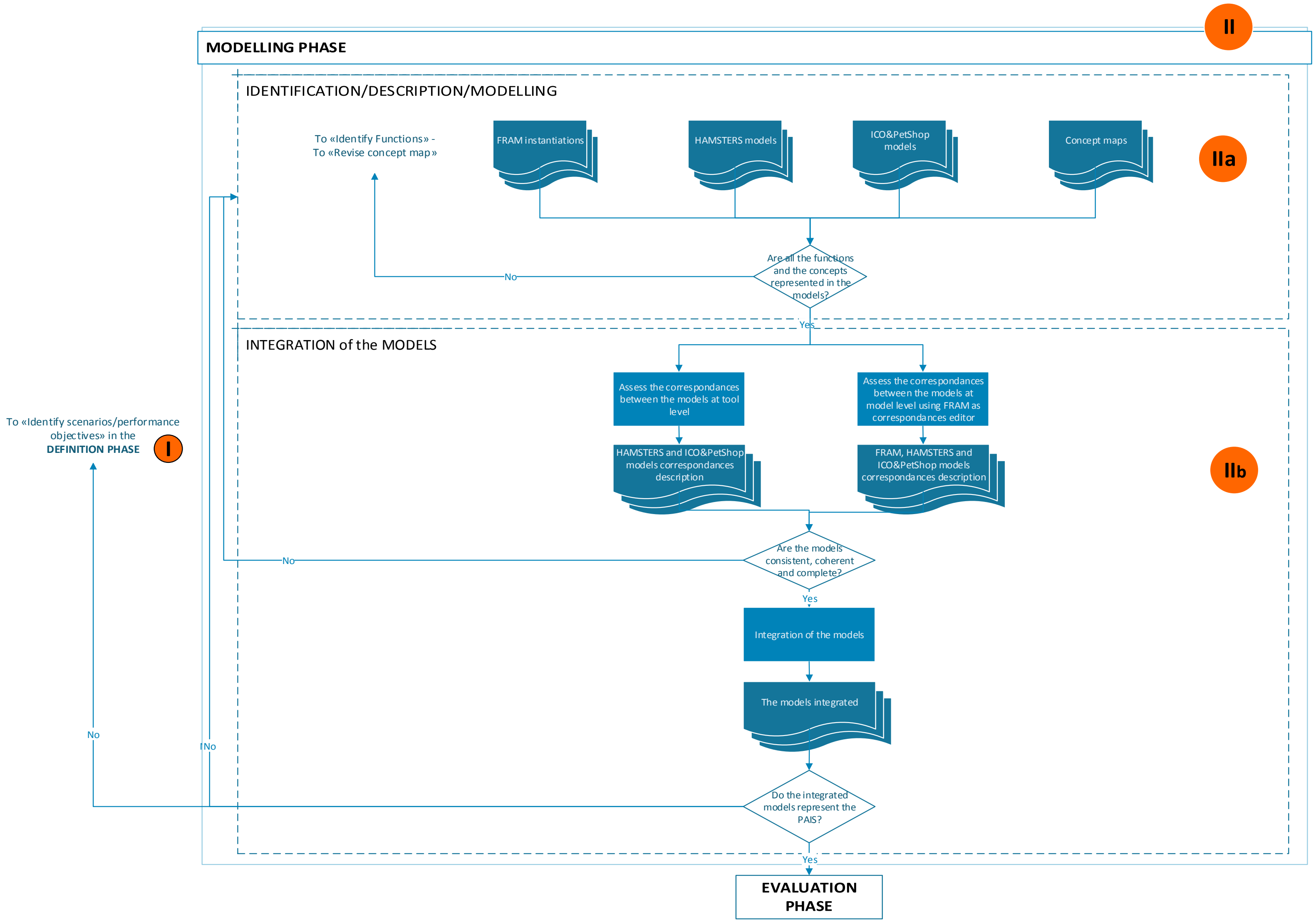


Figure 25: Modelling phase – Integration of the Models sub-phase IIb

These assessments of correspondences are further explained in the following sub-sections (please refer to sub-section “1.2.2.1 The correspondences assessment between models at tool level” and sub-section “1.2.2.2 The correspondences assessment between models at model level).

1.2.2.1 The correspondences assessment between models at tool level

This section details the correspondences at tool level between HAMSTERS 2.0 models and ICO models (which was summarised in Chapter 2 – Section 5). As far as the expressive power of task modelling techniques is concerned, we recapitulate hereafter the main requirements for putting in correspondence task models and ICO models:

- The description of artefacts used to perform task should be close to the representation of objects manipulated by the system;
- User tasks should include elements of the behaviour expected from the system; e.g. user providing an input to the system, requesting a feedback or any kind of system output, or both actions at the same time.
- Task models should be able to express temporal relationships in terms of qualitative relationships (e.g. task ordering) and quantitative relationships (e.g. amount of time required to perform a task). Both kinds of relationships are needed to describe accurately time constraints that are applied during system execution;
- It must be possible to describe tasks models as unities that cooperate rather than monolithic models. This aspect would support a better mapping between tasks and different system’s modules.

Concerning the tool, the following architectural aspects should be taken into account by tools:

- Be open source, or at least provide a powerful API for services enabling to control the models; this is critical to make it possible for the research community to contribute, extend and re-use such tools;
- Provide visual feedback on the current execution; this is important too in order to support the assessment of the adequacy of the task model with users’ real tasks. Indeed, without tool support for simulation it is very cumbersome to understand how the model behaves;
- Supports simulation of scenarios which supports the compatibility assessment activities;
- Implement an API observed/observer of events. This is very important if connection between task modelling tools and system model tools are expected. Without appropriate efficient API integration will be cumbersome and some expected benefits might not be reachable.

The correspondence between these modelling techniques relies on the task side on HAMSTERS environment that provides a set of tools for engineering task models (edition and simulation of models). Similarly, on the system side, the integration relies on the ICO environment (PetShop) that provides means for editing and simulating the system model:

- From the tasks specification we extract the set of interactive tasks (input and output tasks) representing a set of manipulations that can be performed by the user on the system and outputs from the system to the user.
- From the ICO specification we extract the activation and rendering function that may be seen as the set of inputs and outputs of the system model.

The principle of editing the correspondences between the two models is to put together interactive input tasks (from the task model) with system inputs (from the system model) and system outputs (from the system model) with interactive output tasks (from the task model).

Setting up this correspondence may show inconsistencies between the task and system model such as interactive tasks not supported by the system or rendering information not useful for the tasks performance.

This integration allows a real co-evolution of two models, as the execution of one tool impacts the execution of the other tool. This integration can provide designers with shorter iterations in the task and system modelling process. It also represents an improvement for the end user as the execution of the system should support training and provide contextual help.

1.2.2.2 The correspondences assessment between models at model level

This section describes the connections between FRAM and HAMSTERS 2.0 and ICO models. Furthermore, it explains the use of FRAM in assessing the connections between HAMSTERS 2.0 models and ICO models. As reported in Chapter 3 – Section 5, a support tool for FRAM is not available yet. However, this method can be extended and integrated with the others two modelling techniques at model level and it plays a fundamental role in assessing the models connections.

Indeed, FRAM (and its function instantiations) can be considered as a metamodel for connecting and representing the correspondences between HAMSTERS 2.0 and ICO models. It can be defined as a correspondence editor as illustrated in Figure 26.

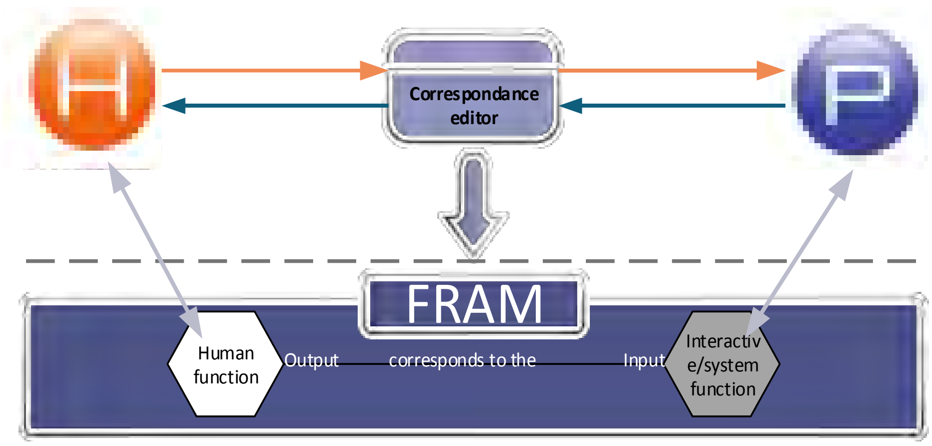


Figure 26: The role of FRAM as correspondence editor

The task models build through HAMSTERS 2.0 have their respective "Human functions" in the FRAM instantiations (represented by the grey bidirectional arrow between the HAMSTERS logo and the FRAM white unit). This is also true for the system models build through PetShop which have the correspondent "System functions" in FRAM (represented by the grey bidirectional arrow between the PetShop logo and the FRAM grey unit). In case of the interactive functions, at each variation on the task model, which impacts on system component, corresponds a change on the system model and vice versa. This means that in FRAM the output of the "Human function" (which corresponds to the HAMSTERS model) causes an input in the "Interactive function" generating a change in the correspondent system model (modelled through PetShop).

Figure 27 shows an illustrative example for better understanding the role of FRAM as correspondence editor. This example is taken from the Game of 15 presented in Chapter 4 – Section 3. On the left side, an excerpt of the "HAMSTERS 2.0 model of Game of 15- subroutine corresponding to the task Select token" (please refer to Figure 19 for the whole representation). The task model has its correspondent function, named "Select a token", in the FRAM instantiation.

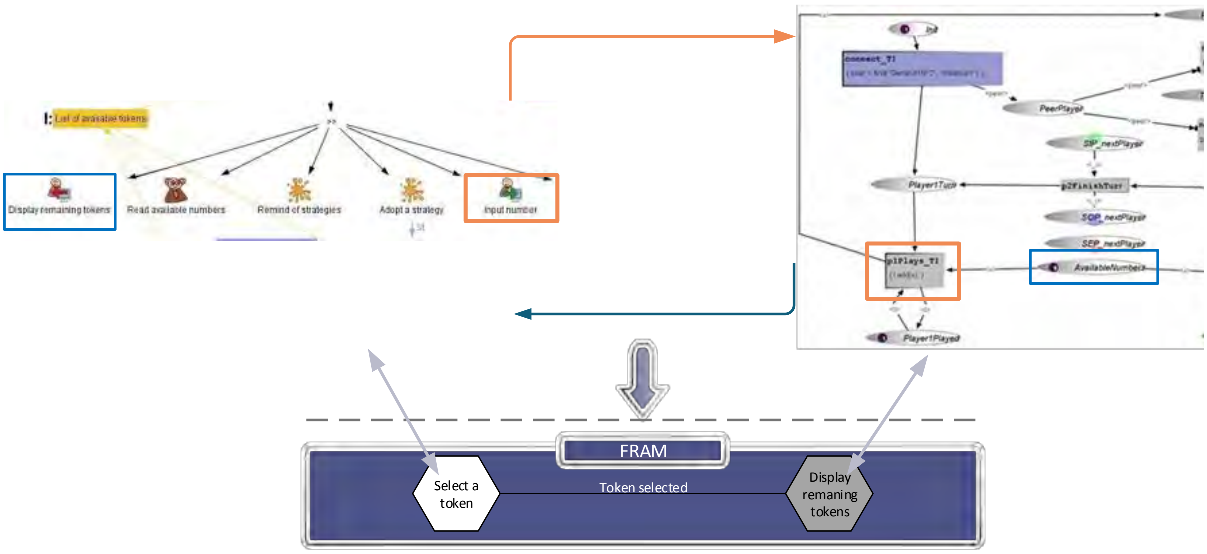


Figure 27: Illustrative example of the role of FRAM as correspondence editor

For accomplishing this task, the Leader has to perform some consecutive tasks. One of these is to "Input number" (the orange box in the task model). This HAMSTERS 2.0 Interactive Input task has its correspondent in the PetShop model (the orange box in the system model). In FRAM, this correspondence is assessed through the output "Token selected" of the function "Select a token" which is the input for activating the "Display remaining tokens" function. Consequently, this system function has its correspondent in the ICO model (the blue box in the system model) which has its correspondent in the HAMSTERS 2.0 model (the blue box in the task model). Once all the correspondences have been assessed, the analyst has to verify if the models properly represent the PAIS within its Socio-Technical System (according to the scenarios and the objectives of the analysis). If they do not ("No" back loop in Figure 25), the analyst has to understand why the representation is not consistent with scenarios and performance objectives. The cause of the problem may be a modelling issue, and in this case, models have to be re-worked/modified (sub-phase IIa in Figure 25). Or, if it is needed to redefine the PAIS under analysis (with associated scenarios and/or objectives of the analysis), the analyst has to re-work on/modify the scenarios and/or objectives re-perform ("Definition phase", labelled I in Figure 25). If the integrated models represent the PAIS within the Socio-Technical System ("Yes" arrow from sub-phase IIb to phase III in Figure 25), s/he can carry on to the "Evaluation phase".

1.3 The Evaluation phase

As described in the previous section, once the integration of the models has been obtained, these can be evaluated. This section explains in detail all the steps of the third main phase of the approach which is the III Evaluation phase. Figure 28 depicts the overview of this phase.

As illustrated in Figure 28: The Evaluation phase, this phase can be further decomposed into two sub-phases:

- The IIIa "Variability assessment" sub-phase
- The IIIb "Coverage of the performance objectives" sub-phase

Both are detailed in the following sections.

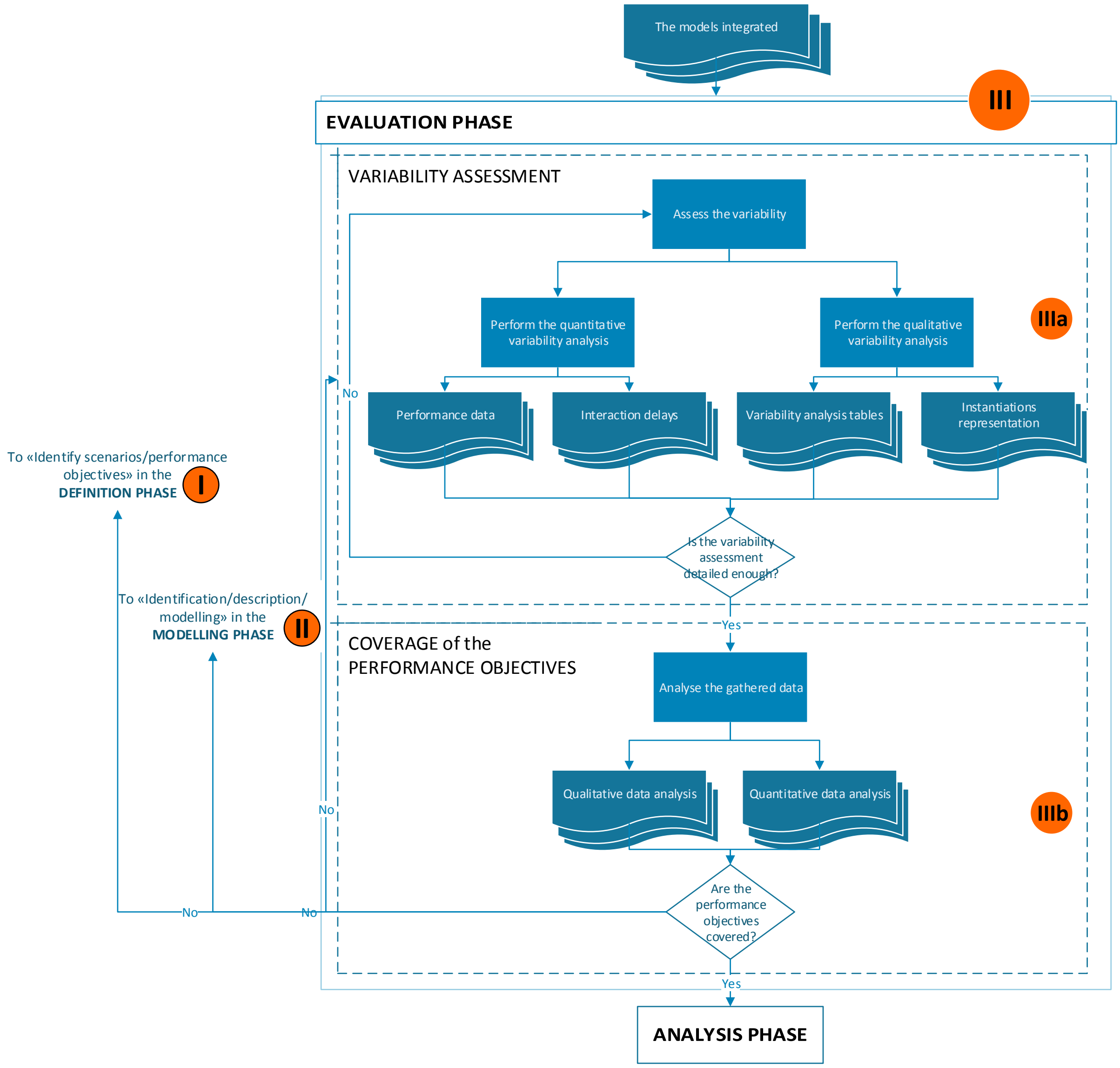


Figure 28: The Evaluation phase

1.3.1 The Variability assessment sub-phase

Figure 29 illustrates the "Variability assessment" sub-phase which consists of several steps. Indeed, the potential variability of the functions described in the  "Identification/Description/Modelling" sub-phase of the "Modelling phase", in this phase is assessed.

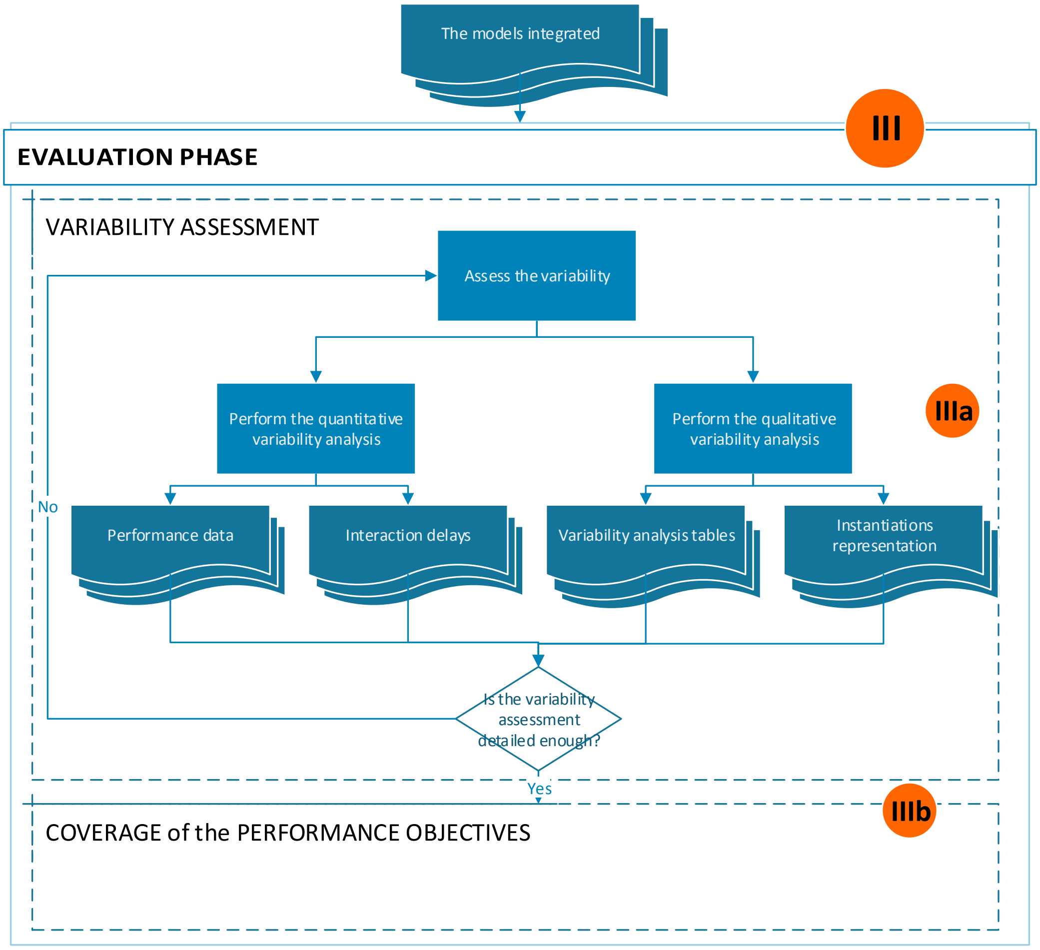


Figure 29: Evaluation phase – Variability assessment sub-phase IIIa

This phase starts with the step "Assess the variability" which consists of two complementary and parallel flows

- "Perform the quantitative variability analysis" and
- "Perform the qualitative variability analysis".

In the former flow, to allow performance assessment, timing issues have to be addressed at three levels: the operator side using the task models presented in Chapter 4 – Section 1, the system side exploiting the ICO behaviour models in Chapter 5 – Section 2.2, the interaction side related to the graphical interface described also in ICOs in Chapter 5 – Section 2.2. The focus on temporal quantitative assessment (represented by the "Delays introduced by the interactive application" table) is because of this aspect is particular relevant for the PAIS in which it plays a role in each elements (human, system and in the interactions). Moreover, time is assessed also in the qualitative variability analysis in which, together with precision, it offers the possibility to evaluate the dynamic couplings between functions (please refer to Chapter 3 – Section 2.2). So, timing issues allow the analyst in evaluating critical and transversal measurements taking into account all the elements of the PAIS and using both quantitative and qualitative variability analyses.

There are several types of performance data which can be assessed via a several evaluation approaches according with the performance objectives.

The quantitative performance variability analysis produces as outputs the "Performance data" and the "Delays introduced by the interactive application" while the qualitative performance variability analysis produces as outputs the "Variability analysis tables" in which dynamic couplings between functions are analysed.

Then, in the latter flow, thanks to variability analysis tables, the analysis of downstream couplings performed in order to identify functional resonance or dampening effects. The coupling factors of the designated function and how these can impact on its output, and consequently, on the input of the downstream function. There are three different severity degrees:

1. No impact which means that there is no impact on the output and on the downstream function,
2. Medium impact which means that the output presents a certain level of variability affecting the downstream function. However, this can be triggered by the output,
3. High impact which means that the output presents a severe level of variability affecting the downstream function and this cannot be triggered.

As described in Chapter 2 – Section 2.2, these couplings also comprise "instantiations representation" of the scenarios which can be used to predict how it develops and how functional resonance can propagate through the functions. Furthermore, these instantiations provide an explanation of what may happen, although it does not necessarily identify unique or specific outcomes. The explanation will be based on the couplings of the variability of everyday performance variability.

Once obtained all these outputs ("Performance data", "Interaction delays", Variability analysis tables", and "Instantiations representation), the analyst has to verify if the qualitative and quantitative assessments of performance variability are detailed enough for the analysis. If they are not ("No" back loop to sub-phase IIIa in phase III in Figure 29), the analyst has to re-work on variability assessments (sub-phase IIIa in Figure 29). If variability assessments are detailed enough ("Yes" arrow to sub-phase IIIb in phase III, in Figure 29), s/he can carry on to sub-phase IIIb.

1.3.2 The Coverage of the performance objectives sub-phase

Once the variability assessment has been detailed enough, it is possible to start the "Coverage of the performance objectives" sub-phase which is represented in the following figure.

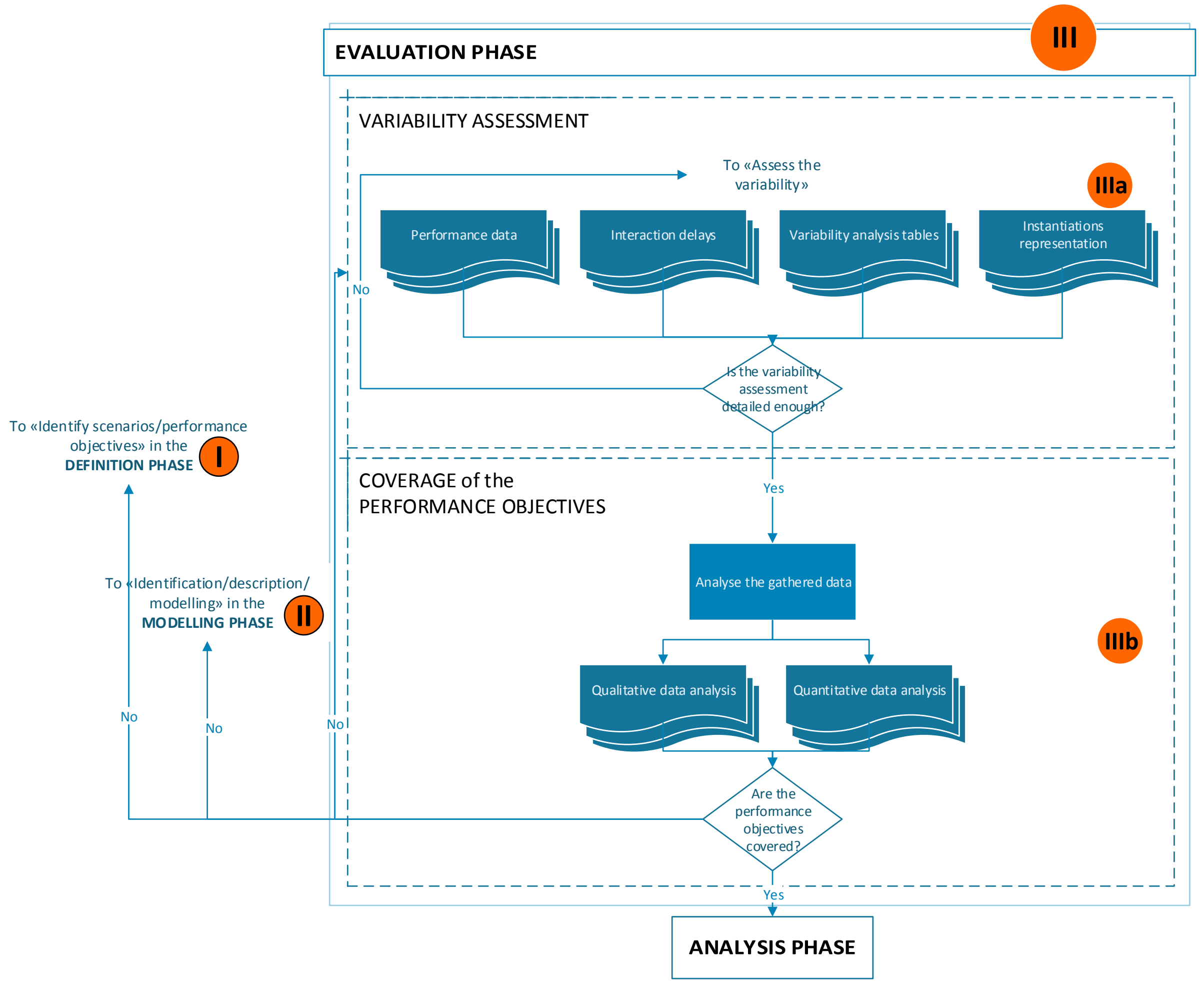


Figure 30: Evaluation phase – Coverage of the performance objectives sub-phase IIIb

Figure 30 illustrates that the first step is to "Analyse the gathered data". Indeed, it is needed to analyse both qualitative and quantitative data. These data can be consolidated and verified through a comparison between them. This should ensure that all the performance objectives are covered. To proceed with the next phase, the analyst has to verify if the performances objectives are covered. If they are not ("No" back loops to the "Identification/description/modelling" of the "Modelling phase" II to and to "Identify scenarios/performance objectives" of Definition phase I in Figure 30), the analyst has to understand why. There may be errors in the models and the analyst will have to re-work on them and to modify them (coming back to the "Identification/description/modelling" of the "Modelling phase" II in Figure 30). Or, the performance objectives may have to be redefined (to "Identify scenarios/performance objectives" of Definition phase I in Figure 30). If the performance objectives are covered ("Yes" arrow to phase IV in Figure 22), s/he can carry on to phase IV.

1.4 The Analysis phase

As described in the previous section, once qualitative and quantitative data have been obtained, these can be adopted for identifying re-design opportunities. This section explains in detail all the steps of the final main phase of the approach which is the IV Analysis phase. Figure 31 depicts the overview of this phase

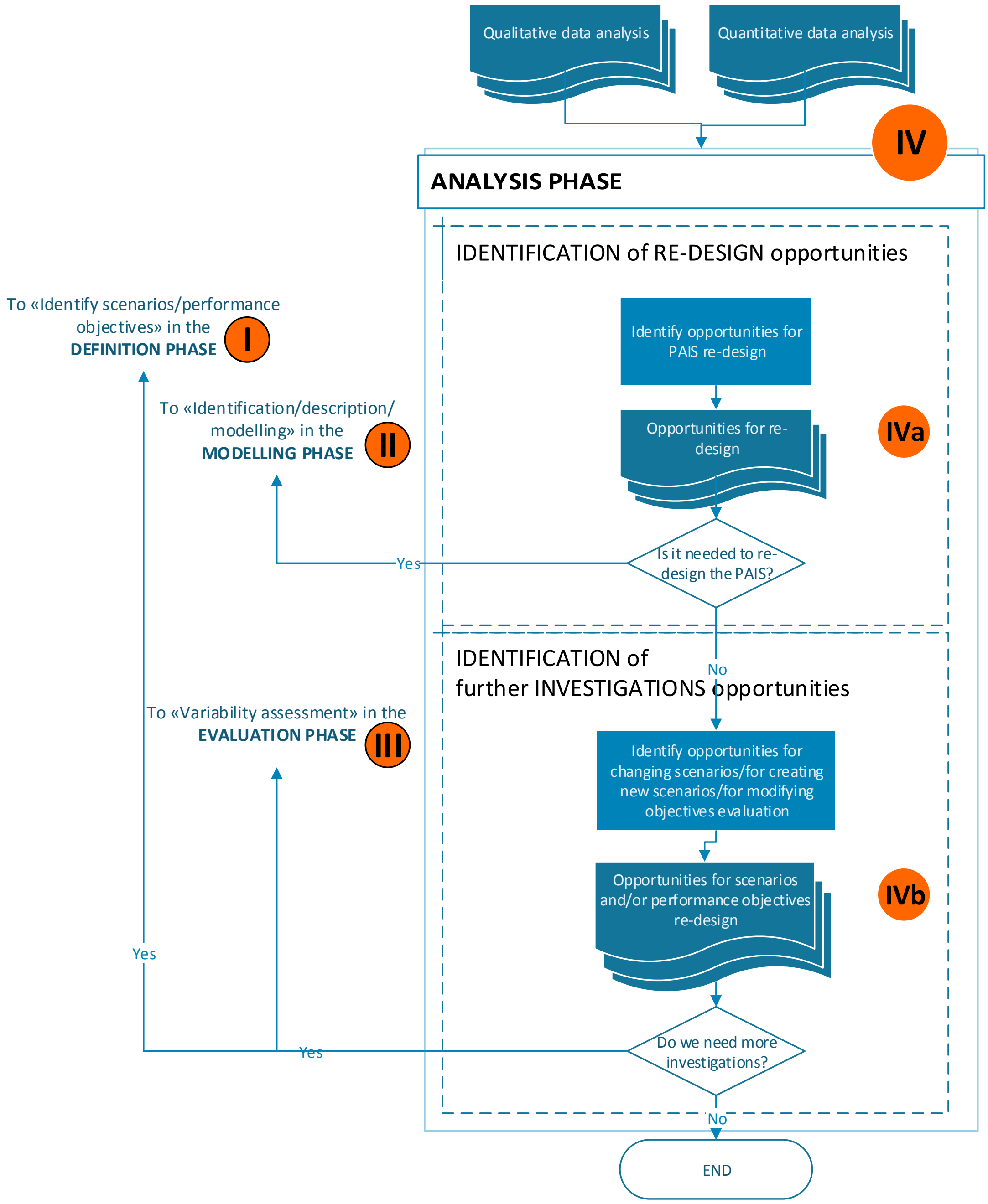


Figure 31: The analysis phase

As illustrated in Figure 31, this phase can be further decomposed into two sub-phases:

- The IVa "Identification of re-design opportunities" sub-phase
- The IVb "Identification of further investigations opportunities" sub-phase

Both are detailed in the following sections.

1.4.1 The Identification of re-design opportunities sub-phase

Figure 32 illustrates the IVa "Identification of re-design opportunities" sub-phase. Once qualitative and quantitative data have been obtained, these can be examined in order to "identify opportunities for PAIS re-design". If the PAIS needs to be re-designed, the analyst rework on the models ("Yes" back loop to the "Identification/description/modelling" of the "Modelling phase" II labelled II in Figure 32) for identifying what it is needed to be improved or changed. This may require modifying one or more of the built models according to the identified opportunities.

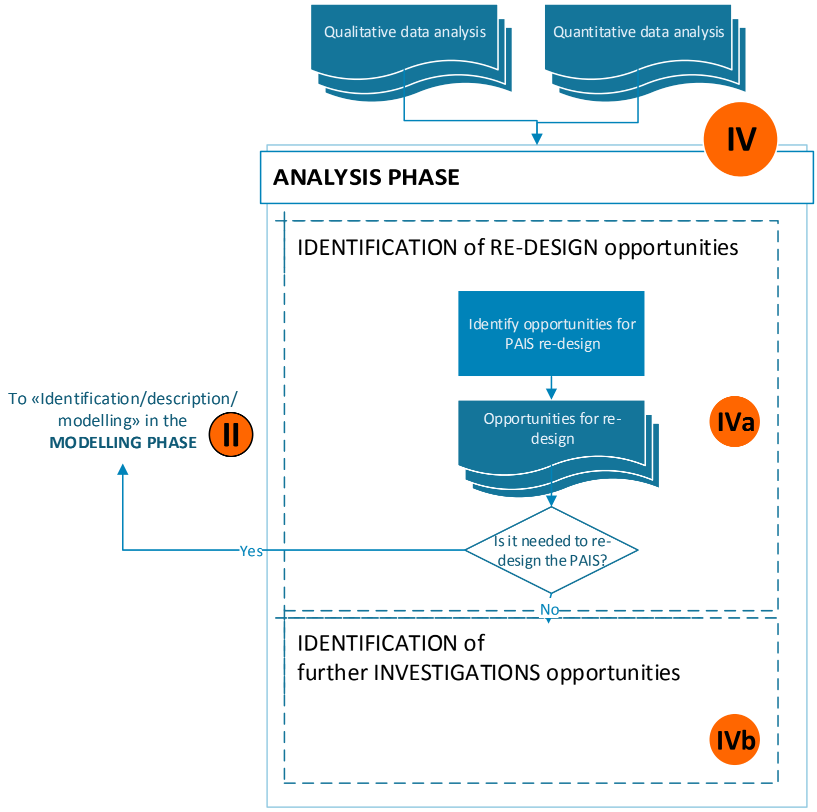


Figure 32: Analysis phase – Identification of re-design opportunities sub-phase IVa

If is the PAIS does not need to be re-designed, the analyst can proceed with the next sub-phase IVb.

1.4.2 The Identification of further investigations opportunities sub-phase

Figure 33 illustrates the IVb "Identification of further investigations opportunities" sub-phase. Once that it has been evaluated which is not needed to re-design the PAIS, it is needed to focus on the scenarios and performance objectives and then, it is required to "Identify opportunities for changing scenarios/for creating new scenarios/for modifying objectives".

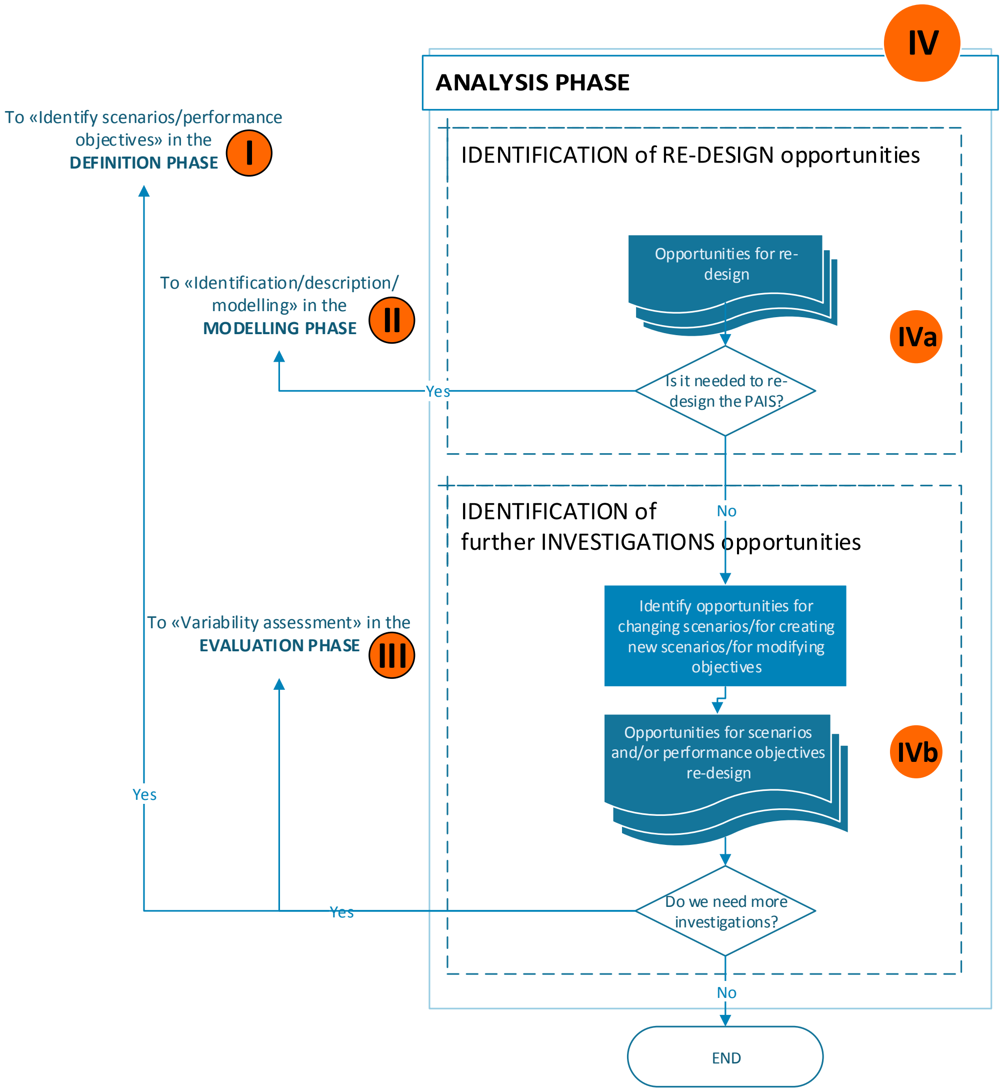


Figure 33: Analysis phase – Identification of further investigations opportunities sub-phase IVb

The analyst can identify opportunities for:

- changing scenarios,
- creating new scenarios, and
- modifying performances objectives.

If additional investigations are required, the analyst has to re-perform some previous steps. These can belong to the previous phase ("Yes" back loop to the "Variability assessment" of the "Evaluation phase" III labelled (III) in Figure 33) if the analyst needs to assess another level(s) of variability. The analyst may also need to change and/or define new scenarios and/or to modify the performance objectives ("Yes" back loop to the "Identify scenarios/performance objectives" of the "Definition phase" I labelled (I) in Figure 33). If further investigations are not required, the process can end.

Albeit with some minor changes, the whole multi-modelS based approach has been applied to the illustrative case study which is presented in the following section.

2 An illustrative example: The Game of 15

To illustrate the multi-modelS based approach developed for modelling and analysing a partly autonomous interactive system within a Socio-Technical System and for assessing its resilience and usability, we propose to apply the approach to the example of the Game of 15 (this example is also used in Chapter 4 for demonstrating the HAMSTERS extensions). The Game of 15 is simple but meaningful as it enables to exemplify the key concepts of our approach. The main principles of the Game of 15 have been presented in Chapter 4 section 3

2.1 The Definition phase applied to the Game of 15

According to the Definition phase of the multi-models based approach (please refer to Chapter 5 – Section 1.1), the first step is the definition of the purpose and of the boundaries of the partly autonomous interactive system to be analysed. Then, it is needed to "Gather information about the PAIS" in order to identify all of its elements. Once gathered all the information about the PAIS under analysis, these have to be organised according to the different types of elements (human, system and interactive). This produces a "Preliminary description of the functions according to human, system and interactive type".

In this case, we are interested in the Game of 15 which is a two player traditional game. Each player takes his/her turn to choose a remaining number (a token) ranging from 1 to 9. The first player whose sum of selected number is exactly 15 wins the game. The purpose is to model and analyse the tasks performed by the players using the interactive application depicted in Figure 17 (please refer to Chapter 4 – Section 3). Despite it is a simple example, it presents all the needed elements for making the list: two players (human element), the system (system element) and the interface (interactive element).

Then, it is needed to "Identify scenarios" and "Identify performance objectives" (as explained in Chapter 5 – Section 1.1). For the Game of 15, a scenario describing the sequence of events that takes place during a game is identified. We hereafter show the extract of this scenario that is relevant to describe the application of the multi-models based approach.

> *"The game starts with the first player (named "Leader") turn. The Leader examines the available tokens that are displayed in the user interface of the game application. All of the tokens are shown as available. S/he thinks about which token to choose and decides to pick up the token 5. Once s/he has selected a token with the user interface of the game application, the turn of the Leader ends. After the end of the Leader turn, the second player (named "Challenger") can start her/his turn. The Challenger examines the remaining tokens that are displayed in the user interface of the game application. All of the tokens are shown as available except token 5. The Challenger then understands that the Leader has already taken token 5. The Challenger then thinks about which token to choose and decides to pick up the token 9…"*

Each event of the scenario may be refined in several ones and the level of the details provided in the scenario relays on the analyst interest.

In parallel of or after having prepared the scenarios, the other step to accomplish is the definition of performance objectives (as described in Chapter 5 – Section 1.1). For the Game of 15, the selected performance objectives are:

“With regard to usability:

- *The interactive application should provide support to perform the token selection in less than 10 seconds.*

With regard to resilience:

- *The players should be able to continue playing whatever the time it takes to one player to perform his/her turn (in terms of continuity of service)”*

Before proceeding with the next phase, the analyst has to verify whether if performance objectives are coherent with scenarios. If they are not coherent (“No” back loop in Figure 23), the analyst has to redefine them or redefine the scenarios, if they are coherent (“Yes” arrow from phase I to phase II in Figure 23), s/he can carry on to phase II

In the described illustrative example, the performance objectives are coherent with the scenario. So, we can carry on to the Modelling phase.

2.2 The Modelling phase applied to the Game of 15

The Modelling phase of the multi-models based approach (described in Chapter 5 – Section 1.2) starts after that a coherent description of scenarios and performance objectives has been obtained. The modelling phase consists in producing models of the PAIS within its associated STS and can be further decomposed into two sub-phases:

- The IIa “Identification/Description/Modelling” sub-phase
- The IIb “Integration of the Models” sub-phase

Both are applied to the Game of 15 and detailed in the following sections.

2.2.1 The Identification/Description/Modelling sub-phase applied to the Game of 15

The IIa “Identification/Description/Modelling” sub-phase includes several steps that can be performed in parallel (please refer to Chapter 5 – Section 1.2.1).

First, all the functions have to be identified and all the information, knowledge and objects required to perform the functions have to be identified respectively. Once all the functions have been identified, they have to be described according to their type. Each scenario, built in the previous phase, can present “Humans”, “Systems” and “Interactive” functions. The analyst can “Define a sequence of events” which produces a “Timeline” representing a sequence of related events arranged in chronological order and displayed along a line.

In the case of the illustrative example, with regard to the “Identify functions” step, the main identified functions are:

- Start to play Game of 15 (Leader)
- Select a token
- Display remaining tokens
- Give turn to the other player

Once identified the functions, these have to be described according to their type (human, system and interactive). Table 7 reports the type of the listed functions.

TABLE 7: FUNCTIONS TYPE DESCRIPTION – GAME OF 15

Function	Type of function		
	Human	*System*	*Interactive*
Start to play Game of 15 (Leader)	x		
Select a token			x
Display remaining tokens		x	
Give turn to the other player	x		

Then, we can build a "Timeline" which represents a sequence of related events arranged in chronological order and displayed along a line. As illustrated in Figure 34, at time 0 (T0) the game starts with the Leader turn. Once s/he has selected a token (T1), the system display remaining tokens (T2). This means that the leader has accomplished all her/his tasks and s/he can give the turn to the other player (T3).

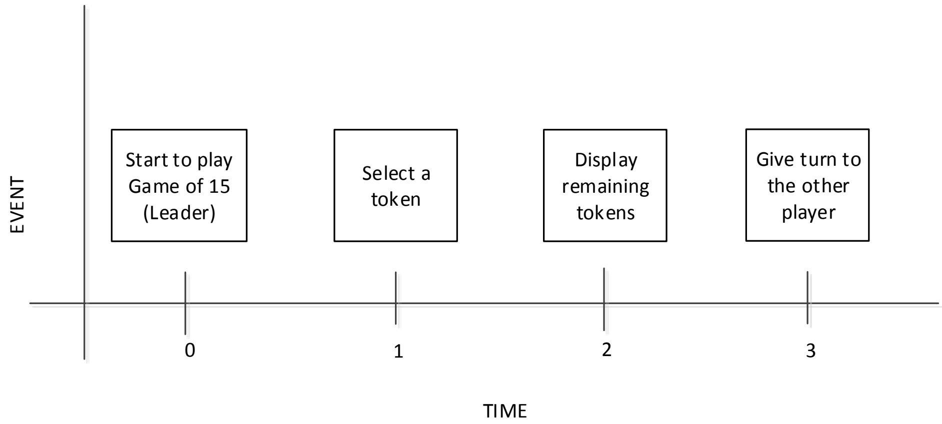


Figure 34: Game of 15 timeline

Once the functions have been listed and the timeline has been described, it is needed to "Characterise potential variability through FRAM" (as described in Chapter 3 – Section 2.2). The following steps are performed on the Game of 15 functions:

- All the functions are described as FRAM units with detailing the six basic aspects (Input, Output, Pre-conditions, Resources, Time, and Control) (please refer to Chapter 3 – Section 2.2);
- The endogenous factors are listed. Endogenous source of variability is related to the internal variability of the PAIS within STS (human, system or their interactions) that performs the function (please refer to Chapter 3 – Section 2.2);
- The potential impact of endogenous factors on the functions output variability is described. Variability of output can be described with a set of dimensions such as timing and precision (please refer to Chapter 3 – Section 2.2).

The produced outputs (FRAM units, endogenous factors table and output variability tables) are needed to create a "FRAM instantiation" (please refer to Chapter 5 – Section 1.2.1).

For the illustrative example, Figure 35 represents the function "Start to play game of 15" as FRAM unit.

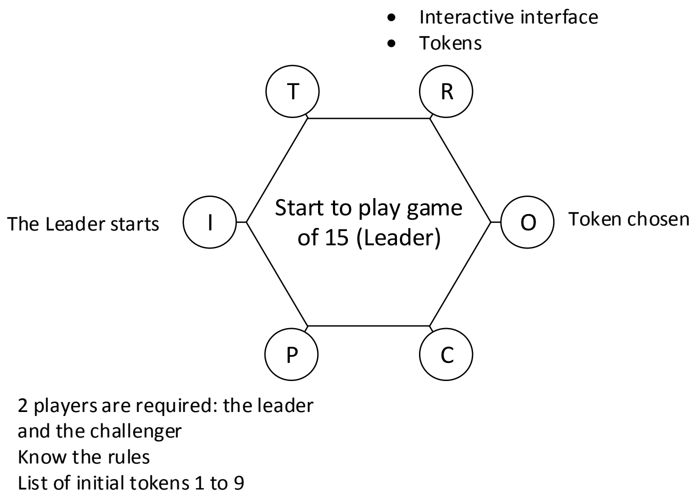


Figure 35: "Start to play game of 15" function as FRAM unit

Moreover, for each FRAM unit, we describe the six aspects through a dedicated table. Some aspects are marked as N/A ("Not Applicable") because they are not relevant for the specific function. Indeed, Table 8 reports the explanation of Input, Output, Precondition, Resource, Control, and Time aspects.

TABLE 8: "START TO PLAY GAME OF 15" FUNCTION FRAM TABLE

START TO PLAY GAME OF 15	
Input(s)	• The Leader starts
Output(s)	• Token chosen
Precondition	• 2 players are required: the leader and the challenger • Know the rules • List of initial tokens 1 to 9
Resource(s)	• Interactive interface • Tokens
Control(s)	N/A
Time	N/A

Then, it is needed to identify all the endogenous factors. As described in Chapter 3 – Section 2.2, this description is supported by a table (Table 9 in the current example) in which are listed the endogenous factors that can affect the variability of the described FRAM function. The impact of these factors can be positive, neutral or negative represented by a green, grey or red arrow respectively. For example, on the first line, the endogenous factor "Experience" can vary in several ways. The experience of the user can be high (s/he has been playing to the Game of 15 for years). The experience of the user can be medium (s/he has been playing several times). The experience of the user can be low (s/he has never played or a very few times).

TABLE 9: ENDOGENOUS FACTORS OF VARIABILITY RELATED TO "START TO PLAY GAME OF 15" FUNCTION

Experience	High (n of times) = ⬆	Medium (n of times) = ⬌	Low (never played)
Focus of attention	High (level …) = ⬆	Medium (level …) = ⬌	Low (level …) = ⬇
Situation Awareness	High (level …) = ⬆	Medium (level …) = ⬌	Low (level …) = ⬇
Workload	High (level …) = ⬇	Medium (level …) = ⬌	Low (level …) = ⬆
Stress	High (level …) = ⬇	Medium (level …) = ⬌	Low (level …) = ⬆
Complacency	High (level …) = ⬇	Medium (level …) = ⬌	Low (level …) = ⬆
Fatigue	High (level …) = ⬇	Medium (level …) = ⬌	Low (level …) = ⬆
Knowledge	High (n of times) = ⬆	Medium (n of times) = ⬌	Low (no knowledge)

In addition to the endogenous factors suggested by the literature, in this PhD thesis, we introduce knowledge (highlighted in blue in each table) as endogenous factor to highlight and explicitly describe the connection between the knowledge artefacts of the selected modelling techniques. Knowledge and its different types are modelled through HAMSTERS 2.0 (as explained and demonstrated in Chapter 4 – Section 1 and 3.1 respectively) and represented in the concept map (as explained and demonstrated in Chapter 4 – Section 2 and 3.2 respectively).

The endogenous factors can have an impact on the output variability in terms of time and precision as reported in Table 10. With regard to time, we can have 3 possible conditions that are "Too early", "On time" and "Too late". With regard to precision, we can have 3 possible conditions that are "Imprecise", "Acceptable" and "Precise". The impact of each condition is explained in the table and it can affect a linked downstream function with different severity degree.

TABLE 10: OUTPUT VARIABILITY OF "START TO PLAY GAME OF 15" FUNCTION

Token chosen		
Time	Too early	N/A
	On time	The Leader is aware of the token chosen.
	Too late	The Leader waits a long time before choosing the token.
Precision	Imprecise	The Leader checks partially or doesn't pay appropriate attention to the choice of the token.
	Acceptable	The Leader chooses the token in an acceptable manner.
	Precise	The Leader properly chooses the token

These sub-steps have to be accomplished for each function. Consequently, the second identified function is "Select a token". This is represented as a FRAM unit in the following figure:

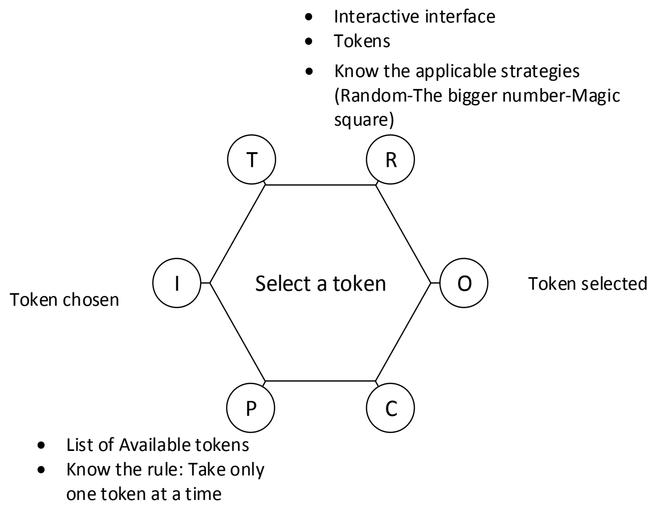


Figure 36: "Select a token" function as FRAM unit

Table 12 reports the explanation of Input, Output, Precondition, Resource, Control, and Time aspects.

TABLE 11: "SELECT A TOKEN" FUNCTION FRAM TABLE

SELECT A TOKEN	
Input(s)	• Token chosen
Output(s)	• Token selected
Precondition	• List of Available tokens • Know the rule: Take only one token at a time
Resource(s)	• Game of 15 application user interface • Tokens • Know the applicable strategies (Random-The bigger number-Magic square)
Control(s)	N/A
Time	N/A

Table 12 reports the endogenous factors that can affect the variability of the described function. The impact of these factors can be positive, neutral or negative represented by a green, grey or red arrow respectively.

TABLE 12: ENDOGENOUS FACTORS OF VARIABILITY RELATED TO" SELECT A TOKEN" FUNCTION

Experience	High (n of times) = ↑	Medium (n of times) = ↔	Low (never played)
Focus of attention	High (level …) = ↑	Medium (level …) = ↔	Low (level …) = ↓
Situation Awareness	High (level …) = ↑	Medium (level …) = ↔	Low (level …) = ↓
Workload	High (level …) = ↓	Medium (level …) = ↔	Low (level …) = ↑
Stress	High (level …) = ↓	Medium (level …) = ↔	Low (level …) = ↑
Complacency	High (level …) = ↓	Medium (level …) = ↔	Low (level …) = ↑
Fatigue	High (level …) = ↓	Medium (level …) = ↔	Low (level …) = ↑
System performance	High (level) = ↑	Medium (level …) = ↔	Low (level …) = ↓
Status of the system	High (level …) = ↑	Medium (level …) = ↔	Low (level …) = ↓
Knowledge	High (n of times) = ↑	Medium (n of times) = ↔	Low (no knowledge)

These factors can impact on the output variability in terms of time and precision as reported in the following table.

TABLE 13: OUTPUT VARIABILITY OF "SELECT A TOKEN" FUNCTION

Token selected		
Time	Too early	N/A
	On time	The Leader is aware of the token selected.
	Too late	The Leader waits a long time before selecting the token.
Precision	Imprecise	The Leader checks partially or doesn't pay appropriate attention to the selection of the token.
	Acceptable	The Leader selects the token s/he was targeting
	Precise	The Leader takes care of ensuring that s/he selects the targeted token

The third identified function is "Display remaining tokens" which is represented as a FRAM unit in the following figure:

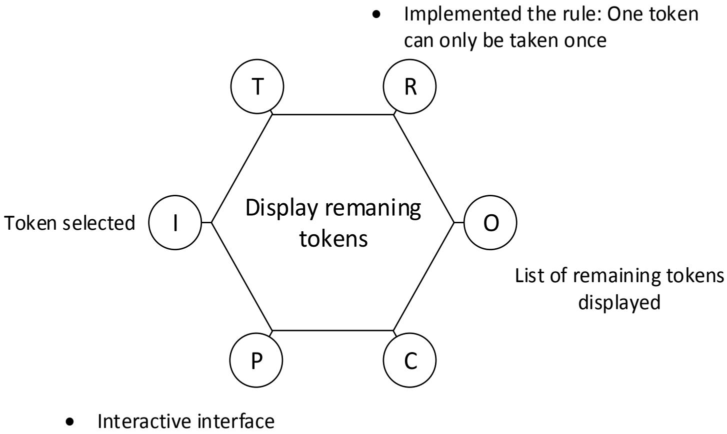


Figure 37: "Display remaining tokens" function as FRAM unit

Table 15 the description of Input, Output, Precondition, Resource, Control, and Time aspects.

TABLE 14: "DISPLAY REMAINING TOKENS" FUNCTION FRAM TABLE

DISPLAY REMANING TOKENS	
Input(s)	• Token selected
Output(s)	• List of remaining tokens displayed
Precondition	• Game of 15 user interface
Resource(s)	• List of available tokens
Control(s)	• Implemented the rule: One token can only be taken once
Time	N/A

Table 15 reports the endogenous factors that can affect the variability of the described function. The impact of these factors can be positive, neutral or negative represented by a green, grey or red arrow respectively.

TABLE 15: ENDOGENOUS FACTORS OF VARIABILITY RELATED TO "DISPLAY REMAINING TOKENS" FUNCTION

System performance	High (level) =	Medium (level …) =	Low (level …) =
Status of the system	High (level …) =	Medium (level …) =	Low (level …) =

These factors can impact on the output variability in terms of time and precision as reported in Table 16.

TABLE 16: OUTPUT VARIABILITY OF "DISPLAY REMAINING TOKENS" FUNCTION

List of remaining tokens displayed		
Time	Too early	N/A
	On time	The displayed list of remaining tokens is successfully updated
	Too late	The displayed list of remaining tokens is not updated in time
Precision	Imprecise	The displayed list of remaining tokens is not correctly updated
	Acceptable	The displayed list of remaining tokens is correctly updated
	Precise	The displayed list of remaining tokens is correctly updated

The last identified function is "Give turn to the other player" which is represented as a FRAM unit in the following figure:

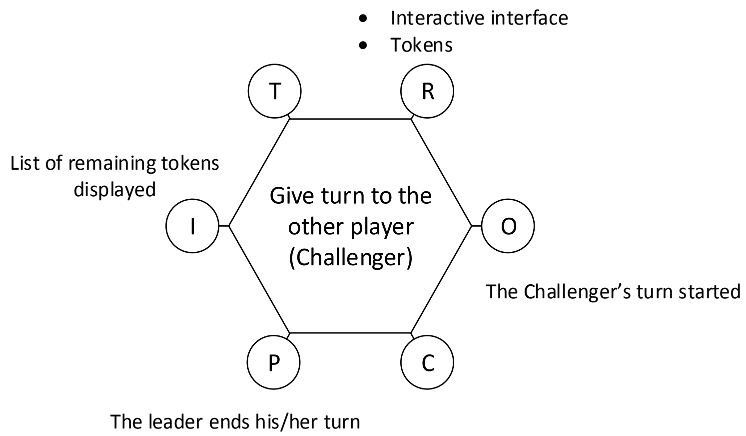


Figure 38: "Give turn to the other player" function as FRAM unit

Table 18 reports the description of Input, Output, Precondition, Resource, Control, and Time aspects.

TABLE 17: "GIVE TURN TO THE OTHER PLAYER" FUNCTION FRAM TABLE

GIVE TURN TO THE OTHER PLAYER	
Input(s)	• List of remaining tokens displayed
Output(s)	• The Challenger's turn started
Precondition	• The Leader ends his/her turn
Resource(s)	• Game of 15 application user interface • Tokens
Control(s)	N/A
Time	N/A

Table 18 reports the endogenous factors that can affect the variability of the described function. The impact of these factors can be positive, neutral or negative represented by a green, grey or red arrow respectively.

TABLE 18: ENDOGENOUS FACTORS OF VARIABILITY RELATED TO "GIVE TURN TO THE OTHER PLAYER" FUNCTION

Experience	High (n of times) =	Medium (n of times) =	Low (never played)
Focus of attention	High (level …) =	Medium (level …) =	Low (level …) =
Situation Awareness	High (level …) =	Medium (level …) =	Low (level …) =
Workload	High (level …) =	Medium (level …) =	Low (level …) =
Stress	High (level …) =	Medium (level …) =	Low (level …) =
Complacency	High (level …) =	Medium (level …) =	Low (level …) =
Fatigue	High (level …) =	Medium (level …) =	Low (level …) =
Knowledge	High (n of times) =	Medium (n of times) =	Low (no knowledge)

These factors can impact on the output variability in terms of time and precision as reported in Table 19.

TABLE 19: OUTPUT VARIABILITY OF "GIVE TURN TO THE OTHER PLAYER" FUNCTION

The Challenger's turn started		
Time	Too early	The Leader gives turn before accomplishing her/his choice
	On time	The Leader gives turn at the right time.
	Too late	The Leader gives turn too late. The Challenger may quit the game
Precision	Imprecise	The Leader doesn't pay appropriate attention to give the turn
	Acceptable	The Leader indicates the challenger that it is her/his turn.
	Precise	The Leader takes care of ensuring that s/he is giving the turn in an understandable way.

Once all the identified functions have been represented as FRAM units, the endogenous factors have been listed and their potential impact on the function output variability has been expressed in terms of time and precision, it is possible to translate the Timeline as FRAM instantiation. The instantiation supports in preliminary detecting the functional resonance between the functions.

Figure 39 illustrates the FRAM instantiation of the Game of 15. The instantiation shows how the functions are coupled between them. These couplings allow us to preliminary detect the functional resonance and how they can be affected from its propagation.

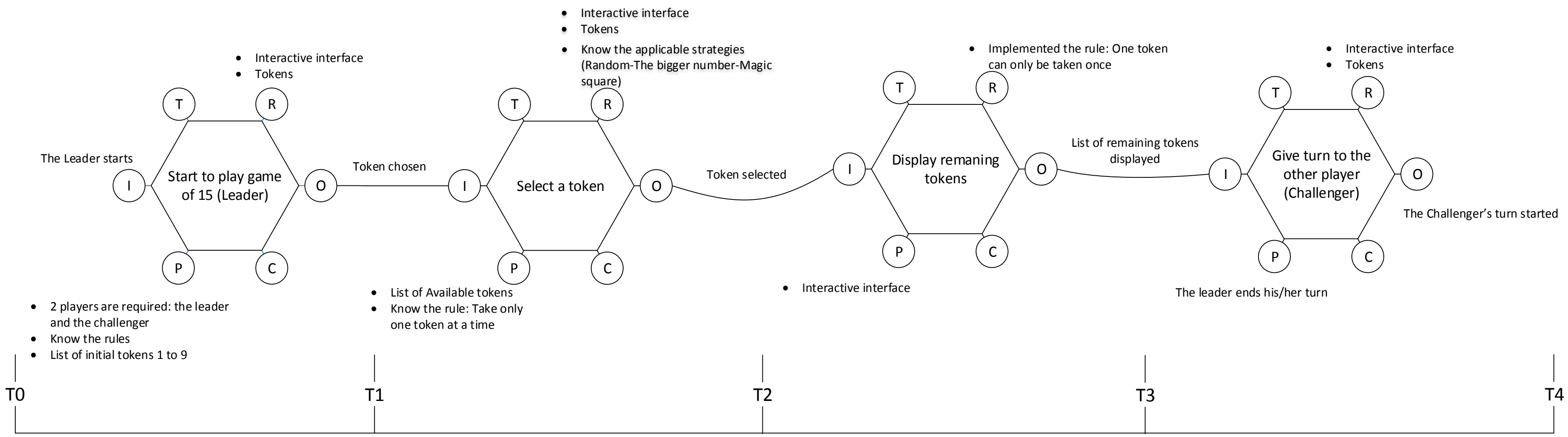


Figure 39: FRAM instantiation of the Game of 15

As described in Chapter 5 – Section 1.2.1, in parallel, for each identified type of function, the analyst will use dedicated modelling techniques for building the related models. According to Chapter 1 – Section 2.2, we selected HAMSTERS for modelling human functions. According to Chapter 2 – Section 2.2, we selected ICO&PetShop for modelling system functions. According to Chapter 2 – Section 5, we selected both modelling techniques for modelling interactive functions. This flow of the sub-phase outputs are "FRAM instantiations", "HAMSTERS models", and "ICO&PetShop models".

For the illustrative example of the Game of 15, the HAMSTERS models have been already provided and described in Chapter 4 – Section "3.1 Task models of Game of 15". However, in order to illustrate the proposed approach in a seamless way, the HAMSTERS models are presented in Figure 40 and Figure 41.

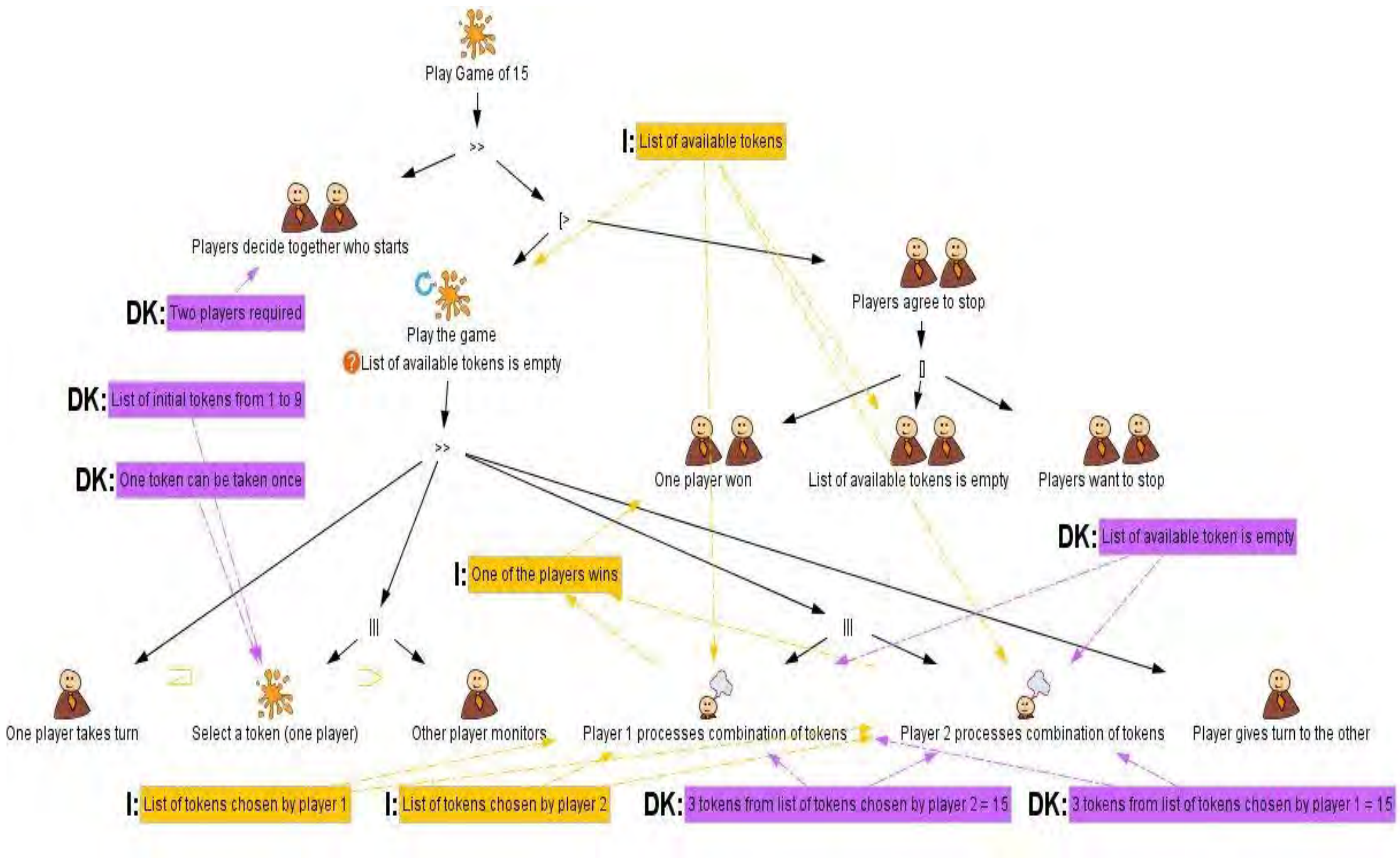


Figure 40: HAMSTERS 2.0 model of Game of 15 – main task to play the game (taken from Chapter 4 – Section "3.1 Task models of Game of 15")

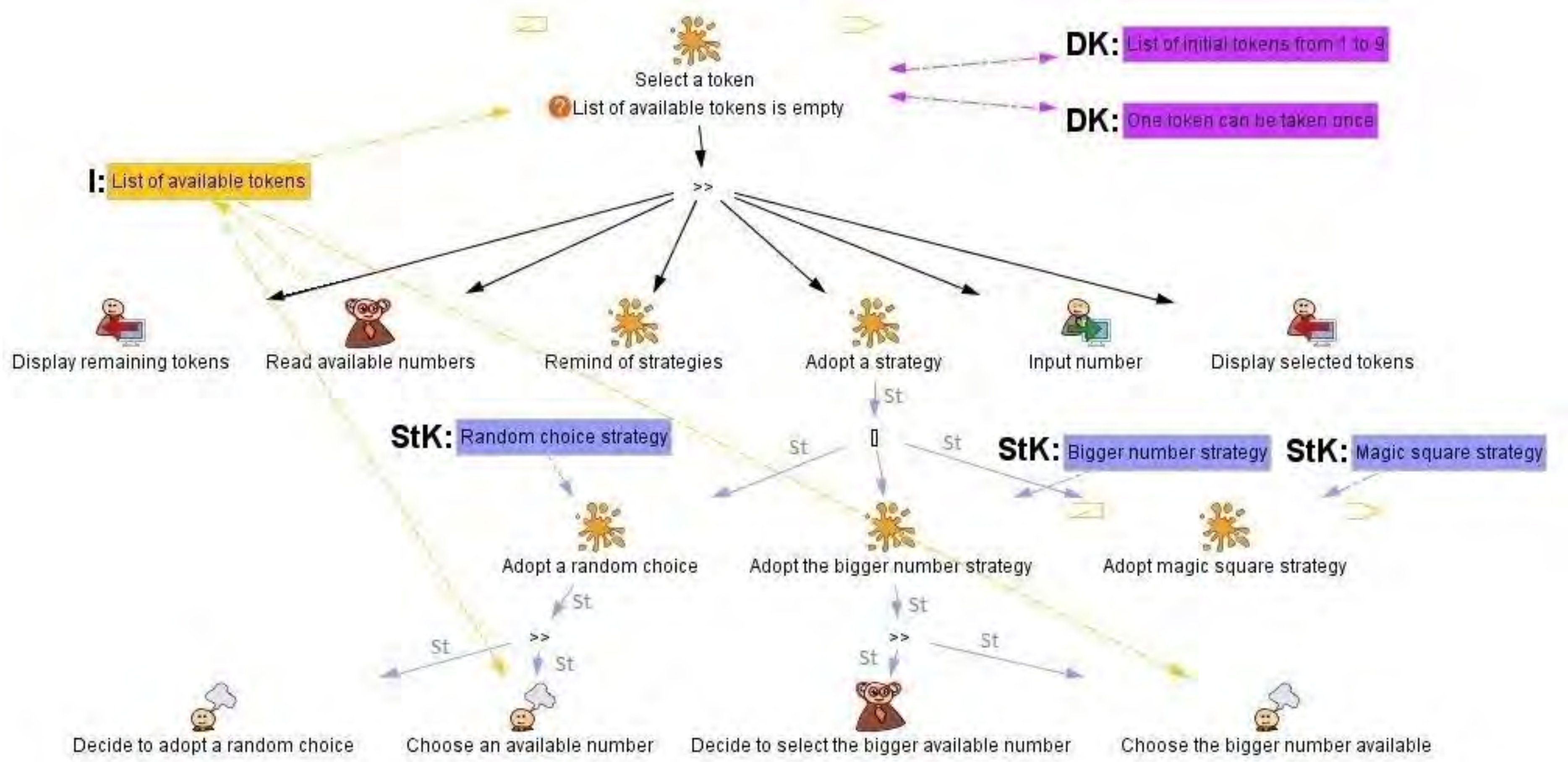


Figure 41: HAMSTERS 2.0 model of Game of 15- subroutine corresponding to the task Select token in Figure 40 (taken from Chapter 4 – Section "3.1 Task models of Game of 15")

Regarding the system models, Figure 42 shows the ICO&PetShop model of Player 1 (Leader). As also represented through FRAM instantiation, the leader turn ends when s/he gives the turn to the Player 2 (Challenger) and the system registers this event. The event is received and the selected value is extracted by the transition called "notifyTurnFinished". The place "AvailableNumbers" contains the available numbers between 1 and 9.

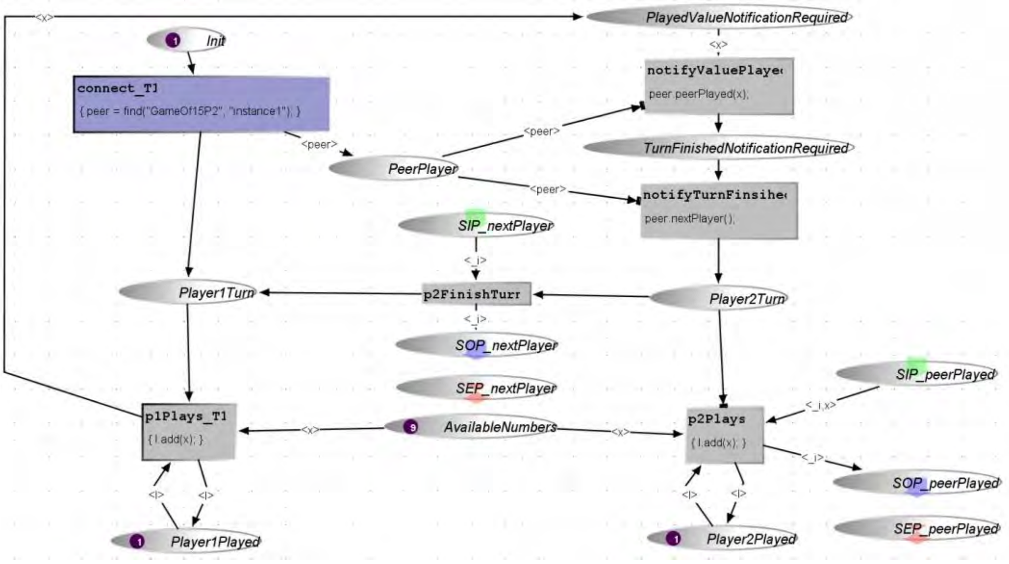


Figure 42: ICO&PetShop model of player 1 (Leader)

According to the number selected by the Player1, this list is updated and the interface shows only the remaining numbers which can be chosen by the player2 in her/his turn. The ICO&PetShop model of player 2 (challenger) is illustrated in Figure 43

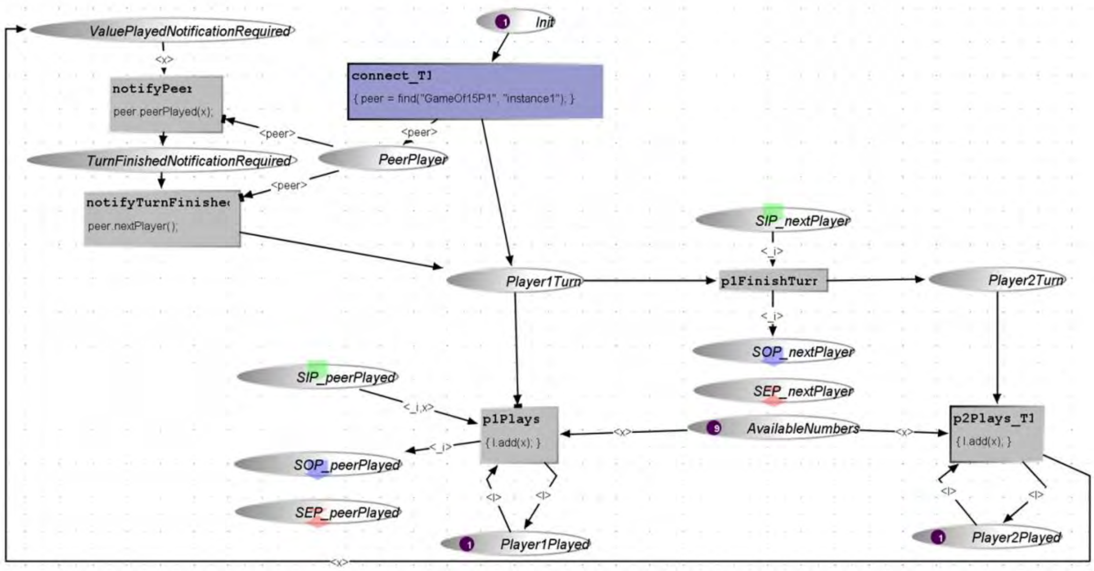


Figure 43: ICO&PetShop model of player 2 (challenger)

Once the turn of the player 1 is ended, the player2 can start her/his turn. S/he can select from the "AvailableNumbers" list the number. The event is received and the selected value is extracted by the transition called "notifyPeer". The place "AvailableNumbers" is accordingly updated and the player2 turn ends via the transition called "notifyTurnFinished".

In this IIa sub-phase, the other parallel flow is "Identify concepts", which will lead to the production of concept maps (for all sub-processes description please refer to Chapter 5 – Section 1.2.1). As demonstrated in PART II – Contributions - Chapter 4 , theoretically in Section 2 and practically in Section 3, concept maps provides a unified description of all the concepts (information, objects, knowledge and their relationships) and they can be used either to represent knowledge and/or to support automated systems for reasoning about the represented knowledge.

For this illustrative example of the Game of 15, the concept map has already been built in Chapter 4 – Section 3.2 and is re-presented in Figure 44 (in order to facilitate the reading process).

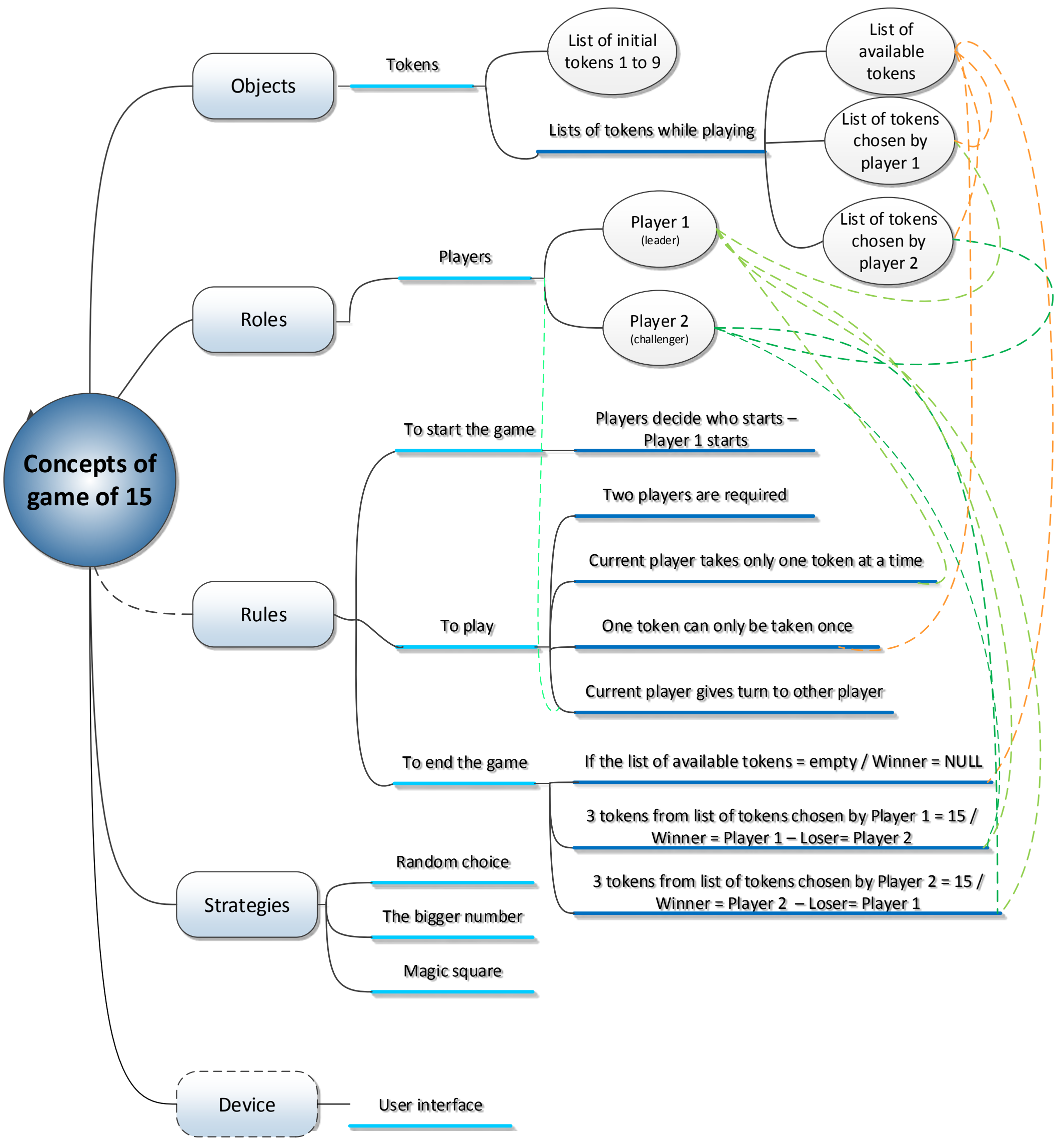


Figure 44: Concept map of Game of 15 (taken from Chapter 4 – Section 3.2)

The concept map can be integrated into the models, as presented in next section.

In order to proceed with the next sub-phase, the analyst has to check whether or not all the functions and all the concepts are represented in the models. If all of them are not represented ("No" back loop in Figure 24), the analyst has to re-perform some previous steps. Hence, s/he has to rework on identifying and describing functions and concepts (flow of steps starting by "Identify functions" and/or flow of steps starting by "Identify concepts" in Figure 24). Once all the functions and the concepts are represented in the models, s/he can carry on to the next sub-phase.

In this case, all the functions and the concepts are presented in the models. So, we can carry on the next sub-phase.

2.2.1 The Integration of the Models sub-phase applied to the Game of 15

The IIb "Integration of the Models" sub-phase includes several steps that can be performed in parallel (please refer to Chapter 5 – Section 1.2.2). Once a complete representation of all the functions has been obtained through the three modelling techniques (FRAM, HAMSTERS and ICO&PetShop) and once a complete representation of the concepts has been obtained through the concept maps, the correspondences between models have to be assessed.

As explained in Chapter 2 – Section 5, HAMSTERS and ICO&PetShop models can be integrated both at model and at tool level. This means that a variation of the system model that impacts a human task will trigger a change in the corresponding task model and vice versa. This helps in iteratively refining models of both types until all the corresponding models are consistent and coherent between each other. In addition, the integration of the system and task models with the FRAM instantiations, at the model level, provides a complementary view for assessing consistency and coherence between system and task models, as it enables to verify the sequence of function calls between models.

2.2.1.1 The correspondences assessment between models at tool level applied to the Game of 15

This section details the correspondences at tool level between HAMSTERS 2.0 models and ICO models regarding the illustrative example of the Game of 15.

As reported in Table 20 and according to the corresponding models, the interactive input task "Select a token" (from the task model built through HAMSTERS 2.0 and shown in Figure 19) corresponds to an Event handler "p1Plays_T1" in the system model (which is built through ICO&PetShop and shown in Figure 42). Moreover, the principle of editing the correspondences between the two models is to put together system outputs Place Event "TokenEntered" in Place "AvailableNumbers" (from the ICO&PetShop model) with Interactive output tasks "Display remaining tokens" (from the HAMSTERS 2.0 model).

TABLE 20: GAME OF 15 – HAMSTERS 2.0AND ICO&PETSHOP MODELS CORRESPONDENCES (LEADER)

Leader		
Input correspondences table		
HAMSTERS	**ICO**	
Interactive input task	**Event handler**	
Select a token	p1Plays_T1	
Output correspondences table		
HAMSTERS	**ICO**	
Interactive output task	**Place**	**Place Event**
Display remaining tokens	AvailableNumbers	TokenEntered

As for the Player 1, that is the Leader, we can assess the correspondences for the Player 2 that is the Challenger. As reported in Table 21 and according to the corresponding models, the interactive input task "Select a token" (from the task model built through HAMSTERS 2.0 and shown in Figure 19) corresponds to an Event handler "p2Plays_T1" in the system model (which is built through ICO&PetShop and shown in Figure 43). Moreover, the principle of editing the correspondences between the two models is to put together system outputs Place Event "TokenEntered" in Place "AvailableNumbers" (from the ICO&PetShop model) with Interactive output tasks "Display remaining tokens" (from the HAMSTERS 2.0 model).

TABLE 21: GAME OF 15 – HAMSTERS 2.0AND ICO&PETSHOP MODELS CORRESPONDENCES (CHALLENGER)

Challenger		
Input correspondences table		
Interactive input task	**Event handler**	
Select a token	p2Plays_T1	
Output correspondences table		
Interactive output task	**Place**	**Place Event**
Display remaining tokens	AvailableNumbers	TokenEntered

The synergistic use allows us in representing all the elements of a partly autonomous interactive system, and in this particular case, the user interface for playing the Game (illustrated in Figure 17 of Chapter 4 – Section 3). Setting up this correspondence may show inconsistencies between the task and system model such as interactive tasks not supported by the system or rendering information not useful for the tasks performance.

This integration allows a real co-evolution of two models, as the execution of one model impacts the execution of the other related model. This integration can provide designers with shorter iterations in the task and system modelling process. It also represents an improvement for the end user as the execution of the system should support training and provide contextual help.

2.2.1.1 The correspondences assessment between models at model level applied to the Game of 15

This section describes the connections between FRAM and HAMSTERS 2.0 and ICO models applied to the Game of 15. As reported in Chapter 3 – Section 5, a support tool for FRAM is not available yet. However, this method can be extended and integrated with the others two modelling techniques at model level and it plays a fundamental role in assessing the models connections.

Indeed, FRAM (and its function instantiations) can be considered as a metamodel for connecting and representing the correspondences between HAMSTERS 2.0 and ICO models. It can be defined as a correspondence editor (please refer to Chapter 5 – Section 1.2.2.2).

The task models built through HAMSTERS 2.0 have their respective "Human functions" in the FRAM instantiations. This is also true for the system models built through PetShop which have the correspondent "System functions" in FRAM. In case of the interactive functions, at each variation on the task model, which impacts on system component, corresponds a change on the system model and vice versa. This means that in FRAM the output of the "Human function" (which corresponds to the HAMSTERS model) causes an input in the "Interactive function" generating a change in the correspondent system model (modelled through PetShop).

Figure 45 (already presented in Chapter 5 – Section 1.2.2.2) shows the role of FRAM as correspondence editor in the illustrative example of the Game of 15. On the left side, an excerpt of the "HAMSTERS 2.0 model of Game of 15- subroutine corresponding to the task Select token" (please refer to Figure 19 for the whole representation). The task model has its correspondent function, named "Select a token", in the FRAM instantiation.

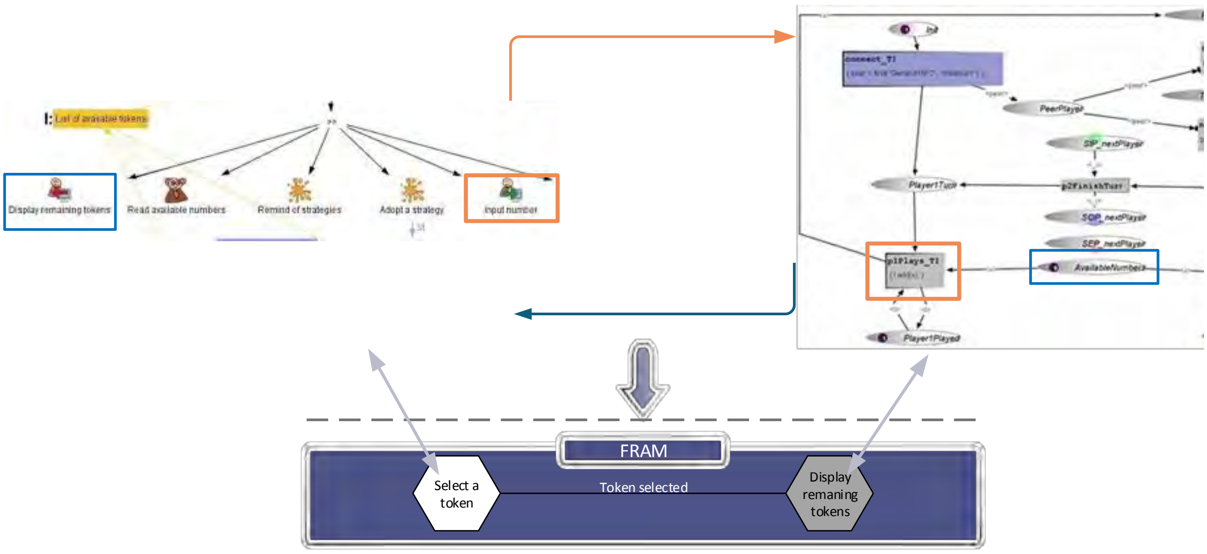


Figure 45: FRAM as correspondence editor in the Game of 15

For accomplishing this task, the Leader has to perform some consecutive tasks. One of these is to “Input number” (the orange box in the task model). This HAMSTERS 2.0 Interactive Input task has its correspondent in the PetShop model (the orange box in the system model). In FRAM, this correspondence is assessed through the output “Token selected” of the function “Select a token” which is the input for activating the “Display remaining tokens” function. Consequently, this system function has its correspondent in the ICO model (the blue box in the system model) which has its correspondent in the HAMSTERS 2.0 model (the blue box in the task model).

Once all the correspondences have been assessed, the analyst has to verify if the models properly represent the PAIS within its Socio-Technical System (according to the scenarios and the objectives of the analysis). If they do not (“No” back loop in Figure 25), the analyst has to understand why the representation is not consistent with scenarios and performance objectives. The cause of the problem may be a modelling issue, and in this case, models have to be re-worked/modified (sub-phase IIa in Figure 25). Or, if it is needed to redefine the PAIS under analysis (with associated scenarios and/or objectives of the analysis), the analyst has to re-work on/modify the scenarios and/or the performance objectives (“Definition phase”, labelled I in Figure 25). If the integrated models represent the PAIS within the Socio-Technical System (“Yes” arrow from sub-phase IIb to phase III in Figure 25), s/he can carry on to the “Evaluation phase”.

In the Game of 15, all the correspondences have been assessed both at tool and also at model level. These assessments ensure that the models are consistent, coherent, and complete. Then, they have been integrated for properly representing the scenario and the related performance objectives. So, we can carry on to the Evaluation phase.

2.3 The Evaluation phase applied to the Game of 15

According to the Evaluation phase of the multi-models based approach (please refer to Chapter 5 – Section 1.3), once the integration of the models has been obtained, these can be evaluated. The third main phase of the approach can be further decomposed into two sub-phases:

- The IIIa “Variability assessment” sub-phase
- The IIIb “Coverage of the performance objectives” sub-phase

Both are applied to the Game of 15 and detailed in the following sections.

2.3.1 The Variability assessment sub-phase applied to the Game of 15

The “Variability assessment” sub-phase consists of several steps (please refer to Chapter 5 – Section 1.3.1). Indeed, the potential variability of the functions described in the “Identification/Description/Modelling” sub-phase of the “Modelling phase”, in this phase is assessed.

This phase starts with the step “Assess the variability” which consists of two complementary and parallel flows:

- “Perform the quantitative variability analysis” and
- “Perform the qualitative variability analysis”.

The quantitative analysis flow provides an output which is the result of the estimations of the variability of functions.. The output is “Performance data” of the scenario. To allow performance assessment we have to assess time performance for the three types of functions: the human functions using the task models presented in Chapter 4 – Section 1, the system functions exploiting the ICO behaviour models in Chapter 5 – Section 2.2, the interactive functions side related to the graphical interface described also in ICOs in Chapter 5 – Section 2.2.

With regard to the users’ performance from task models, we first restricted the study to the interaction with the Game of 25 interface (please refer to “Figure 17: Example of a user interface for playing Game of 15” in Chapter 4 – Section 3). We have estimated the time it will take a user to point and click on a control of graphical interface. One of the evaluation approaches used in human factors domain is based on Fitts’s law (Fitts, 1954) which is suitable for assessing motor movements. Fitts’s law is presented in Formula (1) representing an index of difficulty for reaching a target (of a given size) from a given distance. Movement time (MT) for a user to access a target depends on width of the target (W) and the distance between the start point of the pointer and the centre of target (A).

$$MT = a + b\, log_2\left(1 + \frac{2A}{W}\right) \quad (1)$$

For predicting movement time on the systems under consideration constants are set as follows: a=0 and b=100ms (mean value for users).

The following table presents the set of interactive widgets used within the Game of 15. For each widget (named as Tok in the following table), it provides a short name used for the following tables and the size used as the width for the Fitts’s law (we use the minimum value between the width and the height to provide the assessment of the maximum difficulty to reach the considered widget).

TABLE 22: MIN WIDTH OF THE WIDGETS OF THE GAME OF 15 USER INTERFACE

Interactive widgets	Tok1	Tok2	Tok3	Tok4	Tok5	Tok6	Tok7	Tok8	Tok9	Button Start
Min width (mm)	10	10	10	10	10	10	10	10	10	25

The following table provides the distances from the centre to each widget and between each widget. These distances are used to apply the Fitts’s law when reaching a widget with a start point that can be the centre of the control panel or any widget.

TABLE 23: A) DISTANCE BETWEEN A WIDGET AND THE CONTROL PANEL CENTER IN THE GAME OF 15 USER INTERFACE– B) TEMPORAL VALUES (IN MS) FOR USER INTERACTION USING FITTS’S

a)

Interactive widgets	Tok1	Tok2	Tok3	Tok4	Tok5	Tok6	Tok7	Tok8	Tok9	Button Start

Distance	59	43	22	11	3	11	22	43	59	18

b)

Interactive widgets	Tok1	Tok2	Tok3	Tok4	Tok5	Tok6	Tok7	Tok8	Tok9	Button Start
Temporal values (ms)	369	327	244	168	0	168	244	327	369	129

In addition to these motor values cognitive and perceptive values have to be used in order to cover all the elements of the task models. From (Card, Moran, & Newell, 1986) we know that the mean time for performing a comparison at the cognitive level is 100ms (ranging from 25ms to 170ms) while eye perception mean is 100mn too (ranging from 50ms to 200ms).

With regard to system models, in the ICO Petri net dialect, time is directly related to transition, which invokes services from the Game of 15 (this is the case for transition "notifyTurnFinished" in Figure 42 of Chapter 5 – Section 2.2 which makes the application connect to the remote Player2 application). The duration of each invocation is presented in the following table (each value is coarse grain and depends on the type of interface for playing the Game). In this illustrative example, the application is very light and thus introduces a negligible delay.

TABLE 24: INTERACTION DELAYS INTRODUCED BY THE GAME OF 15 APPLICATION

Model	Transition	Duration (ms)
ICO&PetShop model of player 1 (leader)	connect_T1	20
	p1PlaysCB	20
	notifyValuePlayed	20
	notifyTurnFinished	20
	p2Plays	20
	p2FinishTurn	20
ICO&PetShop model of player 2 (challenger)	connect_T1	20
	p2PlaysCB	20
	notifyPeer	20
	notifyTurnFinished	20
	p1Plays	20
	p1FinishTurn	20

Using the HAMSTERS 2.0 task model in Figure 41 of Chapter 5 – Section 2.2.1 and the time estimations of the table above, we calculate that:

- the maximum time for selecting a token with an interactive widget is 369ms,
- the minimum time for selecting a token with an interactive widgetis 0ms,
- and the average time for selecting a token with an interactive widget is 244,22ms

In parallel to the quantitative analysis, we can perform the qualitative variability analysis. In particular, for the scenario under analysis, an example of functional resonance can be between the two factors, time and precision, of the function "Select a token" (both factors have been individually analysed in Table 13 in Section 2.2.1). The results of the analysis are presented in a variability analysis table. This table synthesizes the downstream coupling between functions in order to identify functional resonance or dampening effects. Table 25 presents the variability analysis of the coupling factors of the function "Select a token" and describes how they can have an impact on the output of the function, and consequently, on the input of the "Display remaining tokens" downstream function. Indeed, the output "Token selected" of the "Select a token" upstream function can have an impact on the input "Token selected" of the "Display remaining tokens" downstream function with the three following different severity degrees:

- No impact (green box in the following table) which means that there is no impact on the output and on the downstream function,
- Medium impact (orange box in the following table) which means that the output presents a certain level of variability affecting the downstream function. However, this can be triggered by the output,
- High impact (red box in the following table) which means that the output presents a severe level of variability affecting the downstream function and this cannot be triggered.

TABLE 25: GAME OF 15 – TIME AND PRECISION COUPLING FACTORS OF "SELECT A TOKEN" FUNCTION

		Precision		
		Precise	Acceptable	Imprecise
Time	Too early	The selection is precise but too early. No impact on the output "Token selected" and on the downstream function.	The selection is acceptable but too early. No impact on the output "Token selected" and on the downstream function.	The selection is imprecise and too early. The wrong selection cannot be corrected. HIGH IMPACT on the output "Token selected" and on the downstream function.
	On time	The selection is precise and on time. No impact on the output "Token selected" and on the downstream function.	The clearance is acceptable and on time. No impact on the output "Token selected" and on the downstream function.	The selection is imprecise but on time. However, the wrong selection cannot be corrected. HIGH IMPACT on the output "Token selected" and on the downstream function.
	Too late	The selection is precise but too late. MEDIUM IMPACT (The other player could quit the game)	The selection is acceptable but too late. MEDIUM IMPACT (The other player could quit the game)	The selection is precise and too late. HIGH IMPACT (The other player quit the game)

Moreover, these three different severity degrees can be represented in the FRAM instantiation as:

- No impact: the FRAM instantiation remains the same (as illustrated in Figure 39: FRAM instantiation of the Game of 15)
- Medium impact: the affected function is marked with an orange wave representing this level of severity but the downstream function can be active (as illustrated in Figure 46),
- High impact: a red cross and a dotted line indicate that the output and the downstream function do not exist anymore (as illustrated in Figure 47).

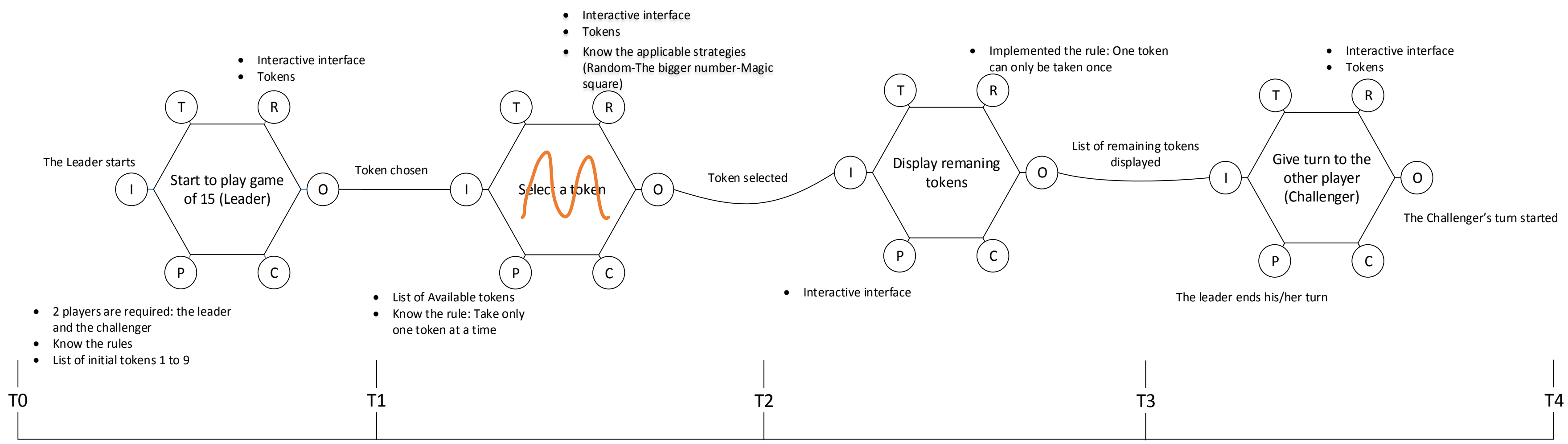


Figure 46: Medium impact of "Select token" function coupling factors – FRAM instantiation of the Game of 15

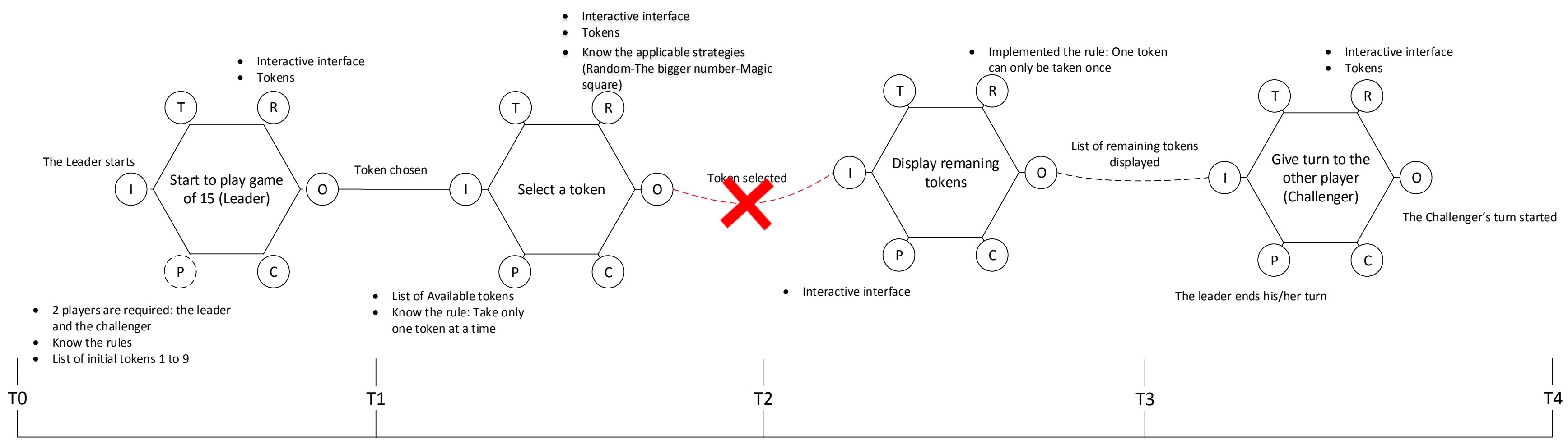


Figure 47: High impact of "Select token" function coupling factors – FRAM instantiation of the Game of 15

Going back to FRAM model, as reported in Table 25 and represented in the instantiations Figure 46 and Figure 47, the function "Select a token" is a strong bottleneck and influences the entire STS under analysis because the impact of its time variability, can strongly affect the downstream function and the entire scenario (one of the player could quit the game).

Once obtained all these outputs ("Performance data", "Interaction delays", "Variability analysis tables", and "Instantiations representation"), the analyst has to verify if the qualitative and quantitative assessments of performance variability are detailed enough for the analysis. If they are not ("No" back loop to sub-phase IIIa in phase III in Figure 29), the analyst has to re-work on variability assessments (sub-phase IIIa in Figure 29). If variability assessments are detailed enough ("Yes" arrow to sub-phase IIIb in phase III, in Figure 29), s/he can carry on to phase IIIb.

In this case, both the qualitative and quantitative assessments of performance variability are detailed enough. So, we can carry on the next sub-phase.

2.3.2 The Coverage of the performance objectives sub-phase applied to the Game of 15

Once the variability assessment has been detailed enough, it is possible to start the "Coverage of the performance objectives" sub-phase (please refer to Chapter 5 – Section 1.3.2). The first step is to "Analyse the gathered data". Indeed, it is needed to analyse both qualitative and quantitative data. These data can be consolidated and verified through a comparison between them. This should ensure that all the performance objectives are covered.

To proceed with the next phase, the analyst has to verify if the performances objectives are covered. If they are not ("No" back loops to the "Identification/description/modelling" of the "Modelling phase" II to and to "Identify scenarios/performance objectives" of Definition phase I in Figure 30), the analyst has to understand why. There may be errors in the models and the analyst will have to re-work on them and to modify them (coming back to the "Identification/description/modelling" of the "Modelling phase" II in Figure 30). Or, the performance objectives may have to be redefined (to "Identify scenarios/performance objectives" of Definition phase I in Figure 30). If the performance objectives are covered ("Yes" arrow to phase IV in Figure 22), s/he can carry on to phase IV.

For the Game of 15, the performance objectives defined in Section 2.1, have been covered both through the quantitative variability analysis and also through the qualitative one in Section 2.3.1. In particular,

- with regard to usability objective:
  - it has been fulfilled because of the token selection is less than 10 seconds;
- with regard to resilience objective:
  - it has not been fulfilled because the players will not be able to continue playing whatever the time it takes to one player to perform his/her turn (in terms of continuity of service)

Now we have to understand why the resilience objective has not been fulfilled and, find a possible solution for achieving it (e.g. through a re-design or through further investigations). So, we can carry on the Analysis phase.

2.4 The Analysis phase applied to the Game of 15

According to the Analysis phase of the multi-models based approach (please refer to Chapter 5 – Section 1.4), once qualitative and quantitative data have been obtained, these can be adopted for identifying re-design opportunities. The fourth main phase of the approach can be further decomposed into two sub-phases:

- The IVa "Identification of re-design opportunities" sub-phase
- The IVb "Identification of further investigations opportunities" sub-phase

Both are applied to the Game of 15 and detailed in the following sections

2.4.1 The Identification of re-design opportunities sub-phase applied to the Game of 15

The first step of the IVa "Identification of re-design opportunities" sub-phase (please refer to Chapter 5 – Section 1.4.1) is to "Identify opportunities for PAIS re-design". If the PAIS needs to be re-designed, the analyst rework on the models ("Yes" back loop to the "Identification/description/modelling" of the "Modelling phase" II labelled II in Figure 32) for identifying what it is needed to be improved or changed. This may require modifying one or more of the built models according to the identified opportunities.

In the case of the Game of 15, according to the results of the evaluation, the resilience objective is not achieved because the players will not be able to continue playing whatever the time it takes to one player to perform his/her turn (in terms of continuity of service), as reported in Table 25. Indeed, in this table the qualitative variability analysis between the coupling factors (Time and Precision) of the "Select token" function have shown that its impact, medium and high, can affect the downstream function and the entire system under analysis in term of Challenger's quit. Then, we can identify an opportunity for re-designing the Game of 15 such as adding, on the system side, a function for displaying an information message to avoid delays in selecting a token. For example, Figure 48 depicts the information message that could be displayed if it takes more than 10 seconds for a user to select a token.

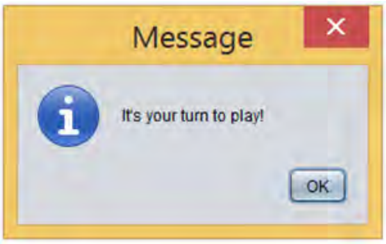


Figure 48: Pop up for the Game of 15

In order to obtain this output, it is needed to modify the ICO models. For example, for what concern the ICO model interactive application for the Player 1, the following figure (Figure 49) shows the revised model in which the "TimeOut" transition (highlighted by a blue box) enables the pop up message.

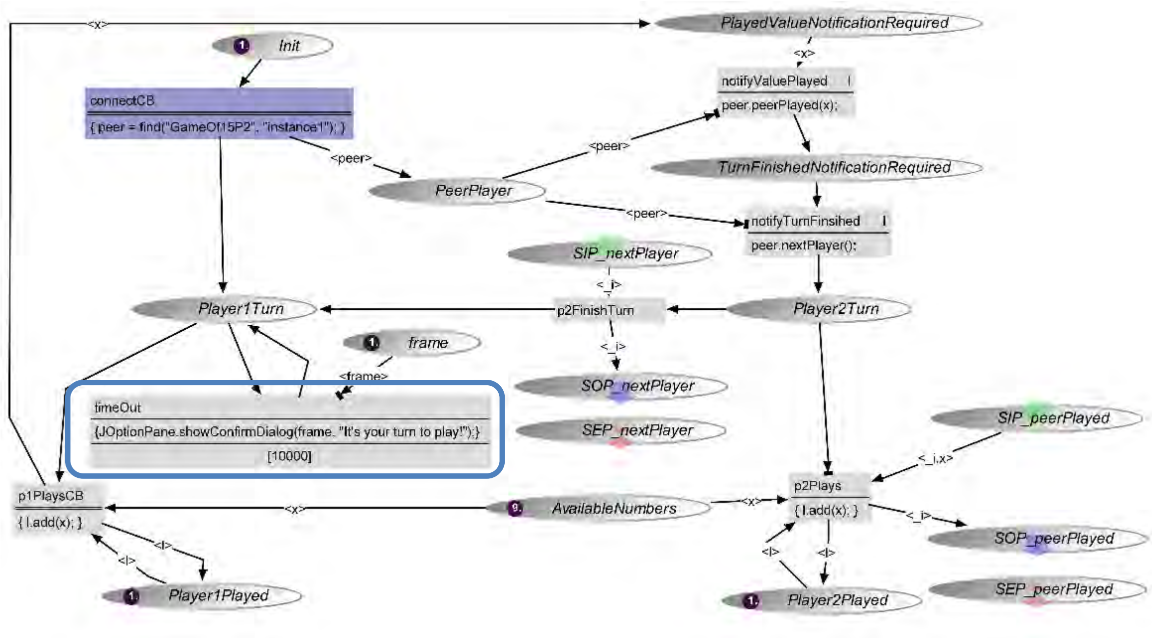


Figure 49: Revised ICO&PetShop model of player 1 (leader)

For what concern the ICO model of the interactive application for Player 2, the following figure (Figure 50) shows the revised model in which the "TimeOut" transition (highlighted by a blue box) enables the pop up message.

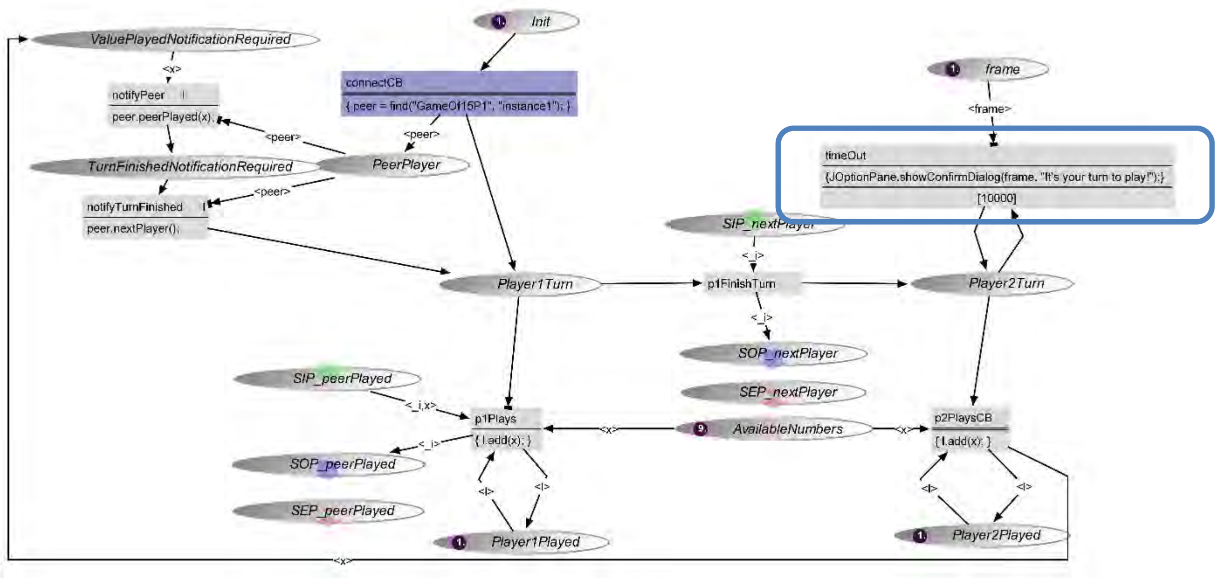


Figure 50: Revised ICO&PetShop model of player 2 (challenger)

These small improvements on the system side could help in:

- improving the resilience of the PAIS within its STS by reducing the medium and high impact of the coupling factors for the "Select a token" function presented in the Table 25 of the FRAM model and illustrated in the instantiations (Figure 46 and Figure 47 respectively)
- improving the usability of the PAIS within its STS by supporting the user in being aware of the elapsed time and of the remaining time for accomplishing the task.

Other opportunities for re-designnig the Game of 15 interactive application could be identified, such as automatically selecting a token for the player who did not select a token in the required time. Once all of the re-design opportunities have been identified, it is needed to select one of the re-design opportunities and to come back to the modelling phase in order to integrate all the models and to re-perform all the required steps for evaluating, qualitatively and quantitatively, the whole variability of the revised partly autonomous interactive system and analysing Iterations can be perfomed until the more appropriate re-design opportunity has been elicited. Furthermore, as argued in Chapter 2 – Section 5 and demonstrated in this section, the developed multi-models approach can support automation design, as it enables the comparison between several design opportunities to allocate tasks and functions between users and systems.

So we can carry on to the next sub-phase.

2.4.2 The Identification of further investigations opportunities sub-phase applied to the Game of 15

Once that it has been ensured that it is not needed to re-design the PAIS, the IVb "Identification of further investigations opportunities" sub-phase (please refer to Chapter 5 – Section 1.4.2) starts in order to focus on the scenarios and on the performance objectives (sub-phase "Identify opportunities for changing scenarios/for creating new scenarios/for modifying objectives").

The analyst can identify opportunities for:

- changing scenarios,
- creating new scenarios, and
- modifying performances objectives.

If additional investigations are required, the analyst has to re-perform some previous steps. These can belong to the previous phase ("Yes" back loop to the "Variability assessment" of the "Evaluation phase" III labelled III in Figure 33) if the analyst needs to assess another level(s) of variability. The analyst may also need to change and/or define new scenarios and/or to modify the performance objectives ("Yes" back loop to the "Identify scenarios/performance objectives" of the "Definition phase" I labelled I in Figure 33). If further investigations are not required, the process can end.

For what concern the illustrative example of the Game of 15, we decide that further investigations are not required. So the application of the process ends here.

3 Conclusion

This application of the proposed approach to the illustrative example of the Game of 15 demonstrates that the multi-models based approach provides a structured process for modelling and analysing a partly autonomous interactive system within its associated STS and provides support for taking into account usability and resilience properties. Then, the contribution is twofold by providing support for both modelling and analysis activities.

On the modelling side, this approach takes the best of each single modelling technique and through their integration into a multi-modelS based approach offers a holistic view on a partly autonomous interactive system within its associated STS. This approach consists in integrating the FRAM method with HAMSTERS 2.0 and ICO modelling techniques and tools. As demonstrated through the illustrative example:

- FRAM provides a high-level view of possible dependencies and explores situations of functional resonance within the partly autonomous interactive system under consideration. It can be considered as the right glue to fix this integration and to ensure consistency, coherence and completeness between the other two modelling techniques acting as a correspondence editor (as explained in detail in Chapter 5 – Section 1.2.2.2 and demonstrated in Section 2.2.1.1);
- The dependencies between the different elements of a STS have been further explored using HAMSTERS for human activities and ICO-PetShop for technical systems covering both interaction techniques on the user interfaces and the underlying hardware and software systems.

The integration of FRAM, HAMSTERS 2.0 and ICO leverages the high-level view on partly autonomous interactive systems, provided by FRAM, with the fine-grain view on human-system interaction, provided by HAMSTERS 2.0 and ICO. Consequently, the approach provides a framework allowing the modelling of partly autonomous interactive system performance variability under different conditions, with different levels of granularity.

On the analysis side, the main contribution is to associate qualitative performance variability analysis of the FRAM method with quantitative user and system performances evaluation support from HAMSTERS 2.0 and ICO (as explained in Chapter 5 – Section 1.3 and demonstrated in Section 2.2.1.1). This integration ensures consistency, coherence and completeness of the data collected during the modelling phase. Thanks to these qualitative and quantitative analyses, we can individuate areas for improving resilience and usability (as demonstrated through the illustrative example). Consequently, the developed multi-models based approach can support automation design.

In order to demonstrate the scalability of the approach, the next Chapter presents a more complex case study from the ATM World. This case study, dealing with aircraft route change due to bad weather conditions, highlights the ability of the integration of models to cope with performance variability of the various parts of partly autonomous interactive system within the Socio-Technical System.

Chapter 6 – A partly autonomous interactive system: the weather radar (WXR) case study

This chapter describes how the multi-models based approach can be applied to a complex case study from the ATM World. This case study, dealing with aircraft route change due to bad weather conditions, provides support for highlighting the ability of the integration of models to cope with performance variability of the various elements of a partly autonomous interactive system and of its associated Socio-Technical System. The WXR case study is suitable for demonstrating the proposed approach as it contains all the different types of elements involved in a partly autonomous interactive system and in its associated STS: the pilot (human element), the WXR (system element) and the WXR interface (interactive element).

The scope of this chapter is to:

- Apply the process and its main phases for modelling and analysing usability and resilience aspects to the weather radar (WXR)
- Demonstrate the scalability of the multi-models based approach

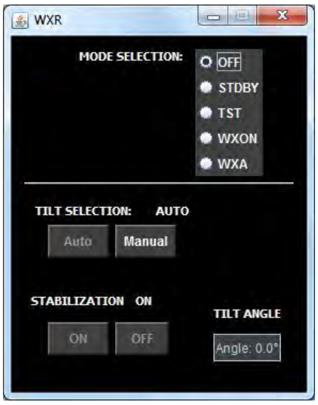
The following sections are organised according to the main phases of the process (in the same way as the application of the proposed process to the Game of 15 presented in previous chapter).

1 The Definition phase applied to the WXR case study

The weather radar (WXR) is an application currently deployed in many cockpits of commercial aircraft. It provides support to pilots' activities by increasing their awareness of meteorological phenomena during the flight journey, allowing them to determine if they may have to request a trajectory change, in order to avoid storms or precipitations for example. Figure 51 shows, on the cockpit of the Airbus A380, the distribution of the various devices related to the weather radar.

Figure 51: WXR system implemented in the FCU of A380 cockpit

Figure 52 presents a screenshot of the weather radar control panel, used to operate the weather radar application. This panel provides two functionalities to the crew. The first one is dedicated to the mode selection of weather radar and provides information about status of the radar, in order to ensure that the weather radar can be set up correctly. The operation of changing from one mode to another can be performed in the upper part of the panel.

a) 

b) 

Figure 52: Image of a) the weather radar control panel b) of the radar display manipulation

The second functionality, available in the lower part of the window, is dedicated to the adjustment of the weather radar orientation (Tilt angle). This can be done in an automatic way or manually (Auto/manual buttons). Additionally, a stabilization function aims to keep the radar beam stable even in case of turbulences. The right-hand part of Figure 52 presents an image of the controls used to configure radar display, particularly to set up the range scale (right-hand side knob with ranges 20, 40, … nautical miles).

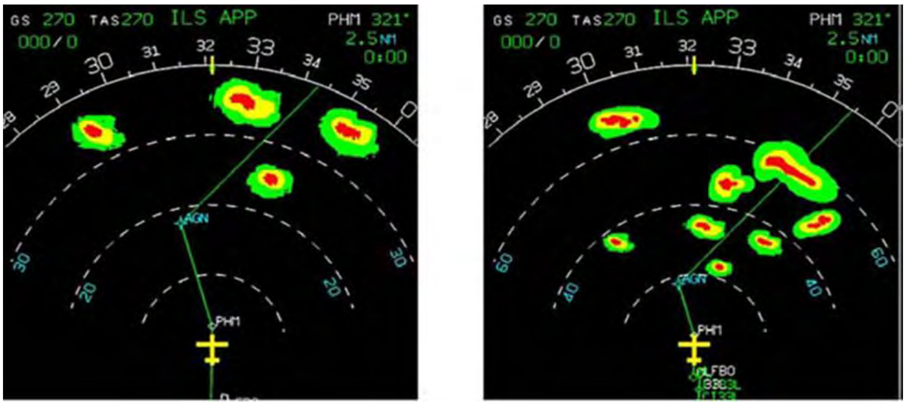


Figure 53: Screenshot of weather radar displays

Figure 53 shows screenshots of weather radar displays according to two different range scales (40 NM for the left display and 80 NM for the right display). Spots in the middle of the images show the current position, importance, composition and size of the clouds.

For the WXR case study, a scenario describing the sequence of events that takes place during an en-route flight phase is identified. We hereafter show the extract of this scenario that is relevant to describe the application of the multi-models based approach.

Extract from Scenario:

> *"While the aircraft is flying, the pilot continuously checks weather conditions in order to detect as soon as possible perturbations. For accomplishing this task, the pilot verifies through the mode selection functionality of weather radar the information about the status of the radar for being sure that the weather radar has been set up correctly and it is properly working. Then, through the second functionality which is dedicated to the adjustment of the weather radar orientation, the pilot checks if there are weather perturbations on the flight path. Turning the knob, s/he can set up different range scale. After having setup the range to 160 NM, s/he sees on the display that there is a perturbation. The pilot recognises a storm cloud and remembers that the planned procedure is to change the current route of the aircraft to avoid passing in this storm cloud. Then, the pilot decides to request a new heading for avoiding the weather perturbation. S/he contacts the executive air traffic controller (EXC) in charge of the sector, and requests to turn left heading 320. When the EXC receives the request, s/he deals with it in order to safely manage the current air traffic flow and coordinate with the pilot the manoeuvre. The EXC analyses that the requested heading clearance is compliant with the trajectories of all the other aircraft of her/his area. The EXC then gives her/his acknowledgement through the clearance turn left heading 320, and the pilot can execute it. The pilot enters the new heading through the FCU and the weather perturbation is avoided."*

Each event of the scenario may be refined in several ones and the level of the details provided in the scenario relays on the analyst interest.

In parallel of or after having prepared the scenario, the other step that has to be accomplished is the definition of the performance objectives (as described in Chapter 5 – Section 1.1). For the WXR case study, the selected performance objectives are:

"With regard to usability:

- *The maximum time to configure the WXR interactive application for displaying weather conditions on the current route shall not exceed 10 seconds.*

With regard to resilience:

- *The pilot should be able to find an appropriate new route and to get approval to implement this new route whatever the surrounding traffic and weather conditions.*

In order to proceed with the next phase we have to verify if performance objectives are coherent with scenarios. Taking into account the identified scenario, it seems that it can be useful for covering only the first performance objective related to the usability. So, as suggested by the developed approach in Chapter 5 – Section 1.1, if the scenario and the performance objectives are not coherent ("No" back loop in Figure 23), the analyst has to redefine them or redefine the scenario. In this case, we redefine the scenario (however, the multi-models based approach has been applied to this first iteration of the scenario and it is synthetically reported in Annex). Indeed, we hereafter show the extract of the redefined scenario.

Extract from redefined Scenario:

"While the aircraft is flying, the pilot continuously checks weather conditions in order to detect as soon as possible perturbations. For accomplishing this task, the pilot verifies through the mode selection functionality of weather radar the information about status of the radar for being sure that the weather radar has been set up correctly and it is properly working. Then, through the second functionality which is dedicated to the adjustment of the weather radar orientation, the pilot checks if there are weather perturbations on the flight path. Turning the knob, s/he can set up different range scale. After having setup the range to 160 NM, s/he sees on the display that there is a perturbation. The pilot recognises a storm cloud remember that the planned procedure is to change the current route of the aircraft to avoid passing in this storm cloud. Then, the pilot decides to request a new heading for avoiding the weather perturbation. S/he contacts the executive air traffic controller (EXC) in charge of the sector, and requests to turn left heading 320. When the EXC receives the request, s/he deals with it in order to safely manage the current air traffic flow and coordinate with the pilot the manoeuvre. The EXC analyses that the requested heading clearance is not compliant with the trajectories of all the other aircraft of her/his sector. Then, the EXC rejects the clearance that was requested by the pilot. The pilot decides to ask for another possible heading and s/he asks the EXC to turn left heading 310. The EXC analyses that the requested heading clearance is not compliant with the trajectories of all the other aircraft of her/his area. Then, the EXC rejects the clearance that was requested by the pilot. The pilot decides to ask for another possible heading and s/he asks the EXC to turn right heading 60. The EXC analyses that the requested heading clearance is not compliant with the trajectories of all the other aircraft of her/his area. Then, the EXC rejects the clearance that was requested by the pilot. The pilot decides to ask for another possible heading and s/he asks the EXC to turn right heading 70. The EXC analyses that the requested heading clearance is compliant with the trajectories of all the other aircraft of her/his area. The EXC then gives her/his acknowledgement through the clearance consisting of the turn right heading 70, and the pilot can execute it. The pilot enters the new heading through the FCU and the weather perturbation is avoided.

This scenario is coherent with the usability and resilience performance objectives, so we can carry on to the next phase.

2 The Modelling phase applied to the WXR case study

2.1.1 The Identification/Description/Modelling sub-phase applied to the WXR case study

The main identified functions are:

- Aircraft flies
- Check weather conditions
  - Scan airspace
  - Rendering of radar information
  - Manage weather radar status
    - Manage mode
    - Manage tilt angle
      - Edit angle
      - Display updated value
      - Change tilt angle
  - Analyse current weather map
  - Select different range map
  - Detect weather target
  - Recognize weather target
- Identify the adequate procedure
- Request a clearance
- Coordinate with EXC
- Execute the clearance

These functions can be represented at higher levels of abstraction or with greater detail as required by the analyst according to her/his interest. For example, “Check weather conditions” function is an abstraction of several lower-level functions such as “Read information” and “Select range map” and so on. This “macro” function includes system, human and interactive functions. Such an abstraction provides support for the representation of a larger number of functions while keeping the graphical model representation understandable.

Once identified the functions, these have to be described according to their type (human, system and interactive). Table 26 reports the type of the listed functions.

TABLE 26: FUNCTIONS TYPE DESCRIPTION – WXR CASE STUDY

Function	Type of function		
	Human	*System*	*Interactive*
Aircraft flies			x
Check weather conditions			
Rendering of radar information		x	
Scan airspace		x	
Manage weather radar status			x
Manage modes			x
Manage tilt angle			x
Edit tilt angle	x		
Display updated value		x	
Change tilt angle			x
Analyse current weather map	x		
Select different range map			x
Detect weather target		x	
Recognize weather target	x		
Identify the adequate procedure	x		
Request a clearance	x		
Coordinate with EXC	x		
Execute the clearance			x

Then, according to the scenario under analysis, we can build a "Timeline" which represents a sequence of related events arranged in chronological order and displayed along a line. As illustrated in Figure 54, at time 0 (T0) the aircraft is flying. In the meantime, the pilot checks the weather conditions (T1). In performing this task, s/he identifies a perturbation (a weather target) and needs to apply the adequate procedure (T2). So, the pilot requests a clearance to the EXC (T3). This means that the pilot has to make coordination with the EXC who is in charge for the sector (T4). When the EXC receives the request, s/he deals with it in order to safely manage the current air traffic flow and coordinate with the pilot the manoeuvre. The EXC analyses that the requested heading clearance is not compliant with the trajectories of all the other aircraft of her/his sector. Then, the EXC rejects the clearance that was requested by the pilot (T5). The pilot decides to ask for another possible heading and s/he asks the EXC to turn left heading 310 (T6). The EXC analyses that the requested heading clearance is not compliant with the trajectories of all the other aircraft of her/his area (T4a). Then, the EXC rejects the clearance that was requested by the pilot (T5a). The pilot decides to ask for another possible heading and s/he asks the EXC to turn right heading 60 (T6b). The EXC analyses that the requested heading clearance is not compliant with the trajectories of all the other aircraft of her/his area (T4b). Then, the EXC rejects the clearance that was requested by the pilot (T5b). The pilot decides to ask for another possible heading and s/he asks the EXC to turn right heading 70 (T6c). The EXC analyses that the requested heading clearance is compliant with the trajectories of all the other aircraft of her/his area (T7). The EXC then gives her/his acknowledgement through the clearance consisting of the turn right heading 70, and the pilot can execute it (T8). The pilot enters the new heading through the FCU and the weather perturbation is avoided (T9).

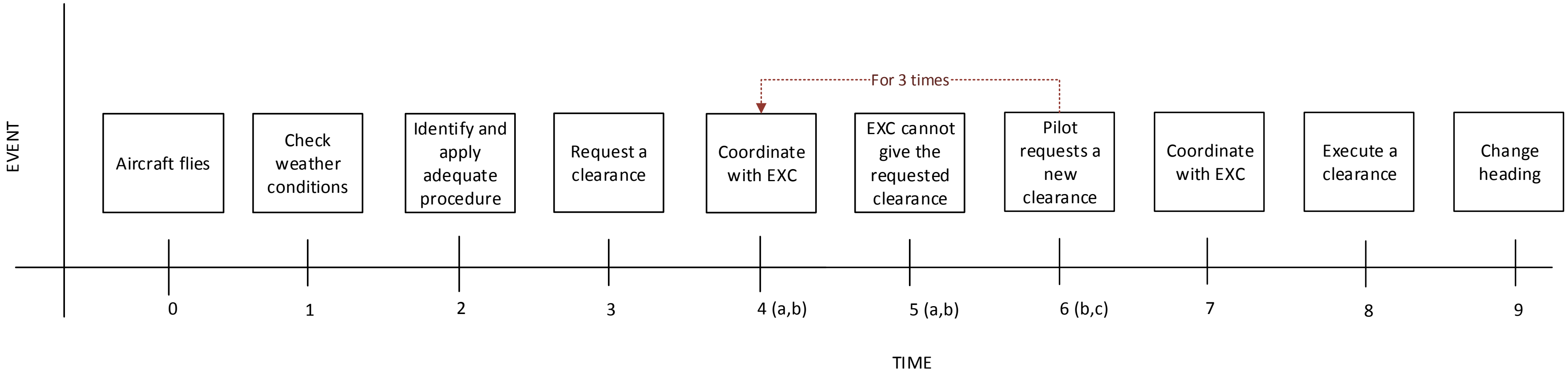


Figure 54: The WXR case study timeline

Once the functions have been listed and the timeline has been described, it is needed to "Characterise potential variability through FRAM" (as described in Chapter 3 – Section 2.2). The following steps are performed on the WXR case study functions:

- All the functions are described as FRAM units with detailing the six basic aspects (Input, Output, Pre-conditions, Resources, Time, and Control) (please refer to Chapter 3 – Section 2.2);
- The endogenous factors are listed. Endogenous source of variability is related to the internal variability of the PAIS within STS (human, system or their interactions) that performs the function (please refer to Chapter 3 – Section 2.2);
- The potential impact of endogenous factors on the functions output variability is described. Variability of output can be described with a set of dimensions such as timing and precision (please refer to Chapter 3 – Section 2.2).

The produced outputs (FRAM units, endogenous factors table and output variability tables) are needed to create a "FRAM instantiation" (please refer to Chapter 5 – Section 1.2.1).

For the WXR case study, Figure 55 represents the function "Aircraft flies" as FRAM unit.

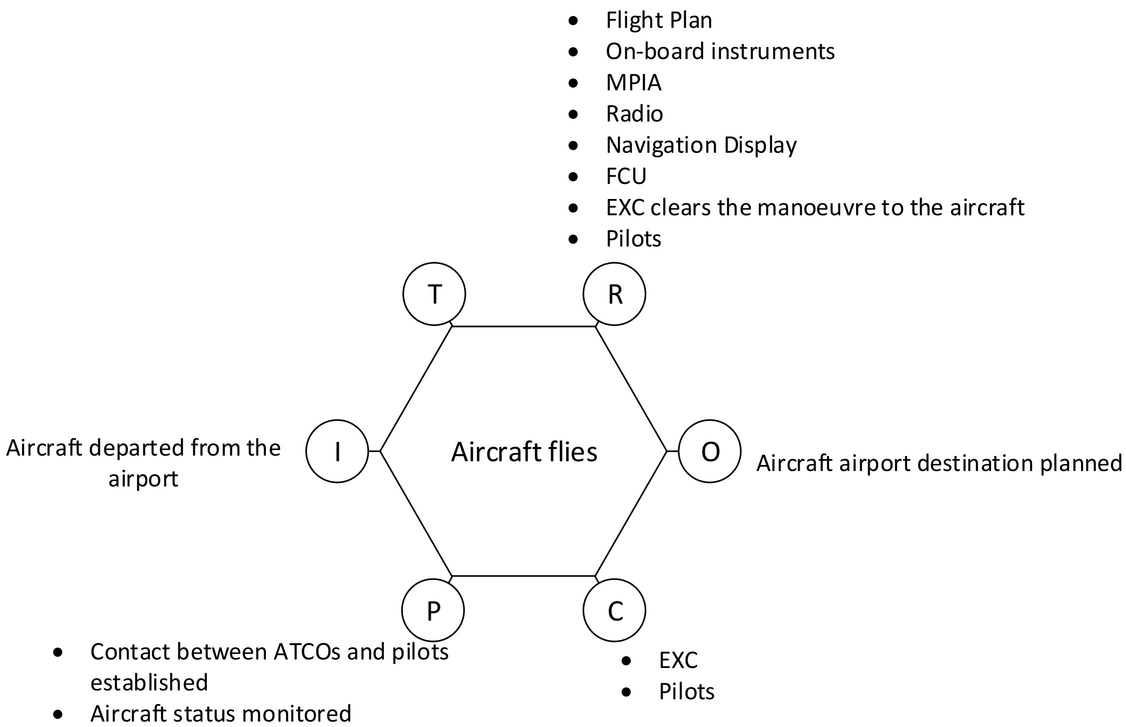


Figure 55: "Aircraft flies" function as FRAM unit

Moreover, for each FRAM unit, we describe the six aspects through a dedicated table. Some aspects may be marked as N/A ("Not Applicable") because of they are not relevant for the specific function. Indeed, Table 27 reports the explanation of Input, Output, Precondition, Resource, Control, and Time aspects.

TABLE 27: "AIRCRAFT FLIES" FUNCTION FRAM TABLE

AIRCRAFT FLIES	
Input(s)	• Aircraft departed from the airport
Output(s)	• Aircraft airport destination planned
Precondition	• Contact between ATCOs and Flight Crew established • Aircraft status monitored
Resource(s)	• Flight Plan • On-board instruments • MPIA • Radio • Navigation Display • FCU • EXC clears the manoeuvre to the aircraft • Pilots
Control(s)	• EXC • Pilots
Time	Time related to speed and airports distance

Then, it is needed to identify all the endogenous factors. As described in Chapter 3 – Section 2.2, this description is supported by a table (Table 28 in the current case study) in which are listed the endogenous factors that can affect the variability of the described FRAM function. In the WXR case study, we can have also exogenous factors (e.g. the number of aircraft in the same airspace and the weather conditions) that can affect the variability of the described FRAM function (these are reported in Table 29). The impact of both these factors can be positive, neutral or negative represented by a green, grey or red arrow respectively.

TABLE 28: ENDOGENOUS FACTORS OF VARIABILITY RELATED TO "AIRCRAFT FLIES" FUNCTION

Experience	High (years …) = ↑	Medium (years …) = ↔	N/A (people have been trained and qualified)
Focus of attention	High (level …) = ↑	Medium (level …) = ↔	Low (level …) = ↓
Situation Awareness	High (level …) = ↑	Medium (level …) = ↔	Low (level …) = ↓
Workload	High (level …) = ↓	Medium (level …) = ↔	Low (level …) = ↑
Stress	High (level …) = ↓	Medium (level …) = ↔	Low (level …) = ↑
Complacency	High (level …) = ↓	Medium (level …) = ↔	Low (level …) = ↑
Fatigue	High (level …) = ↓	Medium (level …) = ↔	Low (level …) = ↑
System performance	High (level) = ↑	Medium (level …) = ↔	Low (level …) = ↓
Status of the system	High (level …) = ↑	Medium (level …) = ↔	Low (level …) = ↓
Knowledge	High (No. of procedures/strategies) = ↑	Medium (No. of procedures) = ↔	N/A

TABLE 29: EXOGENOUS FACTORS OF VARIABILITY RELATED TO "AIRCRAFT FLIES" FUNCTION

N° Aircraft	High (n …) = ↓	Medium (n …) = ↔	Low (n …) = ↑
Weather perturbations	High (level …) = ↓	Medium (level …) = ↔	Low (level …) = ↑

The endogenous and exogenous factors can have an impact on the output variability in terms of time and precision as reported in Table 30. With regard to time, we can have 3 possible conditions that are "Too early", "On time" and "Too late". With regard to precision, we can have 3 possible conditions that are "Imprecise", "Acceptable" and "Precise". The impact of each condition is explained in the table and it can affect a linked downstream function with different severity degree.

TABLE 30: OUTPUT VARIABILITY OF "AIRCRAFT FLIES" FUNCTION

Aircraft airport destination planned		
Time	Too early	The aircraft airport destination is planned well in advance with respect to the regulation.
	On time	The aircraft airport destination is planned on time with respect to the regulation.
	Too late	N/A
Precision	Imprecise	N/A
	Acceptable	N/A
	Precise	The aircraft airport destination is planned and well defined.

These sub-steps have to be accomplished for each function. In this case study the next function is "Check weather conditions" that is a "macro" function. As explained before, this means that it can be decomposed into several lower-level functions as "Scan airspace". This is represented as a FRAM unit in the following figure.

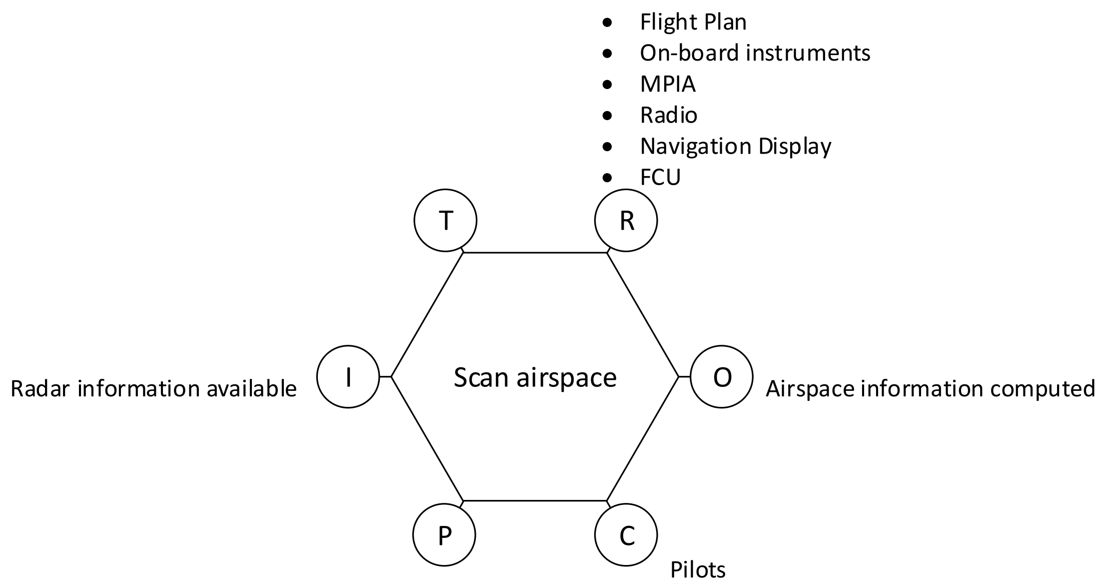


Figure 56: "Scan airspace" function as FRAM unit

Table 31 reports the explanation of Input, Output, Precondition, Resource, Control, and Time aspects.

TABLE 31: "SCAN AIRSPACE" FUNCTION FRAM TABLE

SCAN AIRSPACE	
Input(s)	• Radar information available
Output(s)	• Airspace information computed
Precondition	N/A
Resource(s)	• Flight plan • On-board instruments • MPIA • Radio • Navigation Display • FCU
Control(s)	• Pilots
Time	*Continuous Function – in parallel with the "Rendering of radar information"*

Table 32 and Table 33 report respectively the endogenous and the exogenous factors that can affect the variability of the described function. The impact of these factors can be positive, neutral or negative represented by a green, grey or red arrow respectively.

TABLE 32: ENDOGENOUS FACTORS OF VARIABILITY RELATED TO "SCAN AIRSPACE" FUNCTION

System performance	High (level) =	Medium (level …) =	Low (level …) =
Status of the system	High (level …) =	Medium (level …) =	Low (level …) =

TABLE 33: EXOGENOUS FACTORS OF VARIABILITY RELATED TO "SCAN AIRSPACE" FUNCTION

Weather perturbations	High (level …) =	Medium (level …) =	Low (level …) =

These factors can impact on the output variability in terms of time and precision as reported in Table 34.

TABLE 34: OUTPUT VARIABILITY OF "SCAN AIRSPACE" FUNCTION

Airspace information computed		
Time	Too early	N/A
	On time	The airspace information are successfully computed
	Too late	The airspace information are not computed in time
Precision	Imprecise	The airspace information are not correctly computed
	Acceptable	The airspace information are correctly computed
	Precise	The airspace information are correctly computed

The third identified function is "Rendering of radar information". This is represented as a FRAM unit in the following figure.

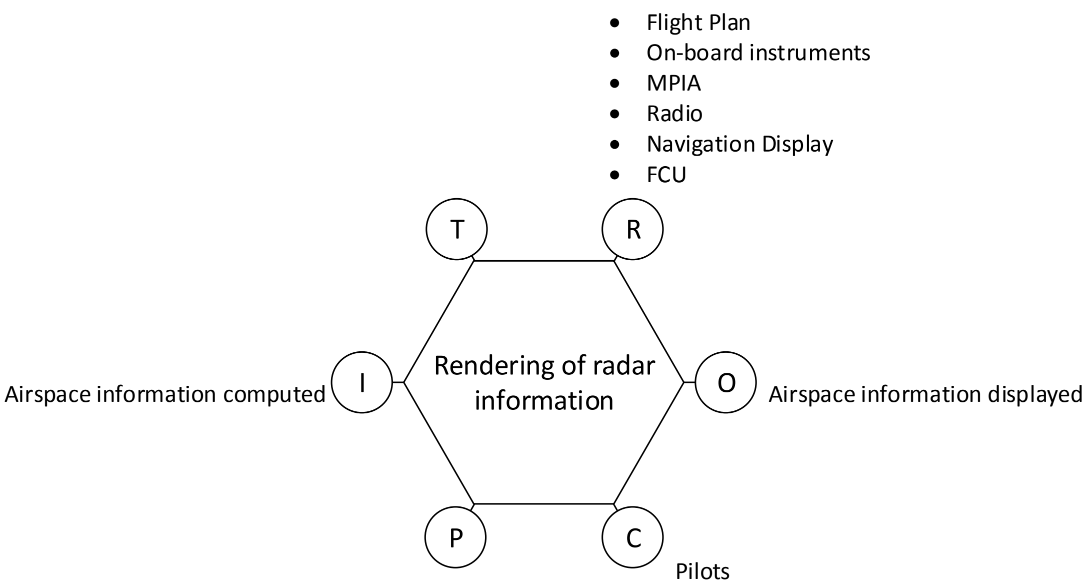


Figure 57: "Rendering of radar information" function as FRAM unit

Table 35 reports the explanation of Input, Output, Precondition, Resource, Control, and Time aspects.

TABLE 35: "RENDERING OF RADAR INFORMATION" FUNCTION FRAM TABLE

RENDERING OF RADAR INFORMATION	
Input(s)	• Airspace information computed
Output(s)	• Airspace information displayed
Precondition	N/A
Resource(s)	• Flight plan • On-board instruments • MPIA • Radio • Navigation Display • FCU
Control(s)	• Pilots
Time	*Continuous Function – in parallel with the "Scan airspace" function*

Table 36 and Table 37 report respectively the endogenous and the exogenous factors that can affect the variability of the described function. The impact of these factors can be positive, neutral or negative represented by a green, grey or red arrow respectively.

TABLE 36: ENDOGENOUS FACTORS OF VARIABILITY RELATED TO "RENDERING OF RADAR INFORMATION" FUNCTION

System performance	High (level) =	Medium (level …) =	Low (level …) =
Status of the system	High (level …) =	Medium (level …) =	Low (level …) =

TABLE 37: EXOGENOUS FACTORS OF VARIABILITY RELATED TO "RENDERING OF RADAR INFORMATION" FUNCTION

Weather perturbations	High (level …) =	Medium (level …) =	Low (level …) =

These factors can impact on the output variability in terms of time and precision as reported in Table 38.

TABLE 38: OUTPUT VARIABILITY OF "RENDERING OF RADAR INFORMATION" FUNCTION

Airspace information displayed		
Time	Too early	N/A
	On time	The airspace information are successfully displayed
	Too late	The airspace information are not displayed in time
Precision	Imprecise	The airspace information are not correctly displayed
	Acceptable	The airspace information are correctly displayed
	Precise	The airspace information are correctly displayed

The following identified function is "Manage weather radar status" which is also a macro function consisting of several lower-level functions that are "Manage modes" and "Manage tilt angle". Figure 58 represents "Manage modes" as FRAM unit.

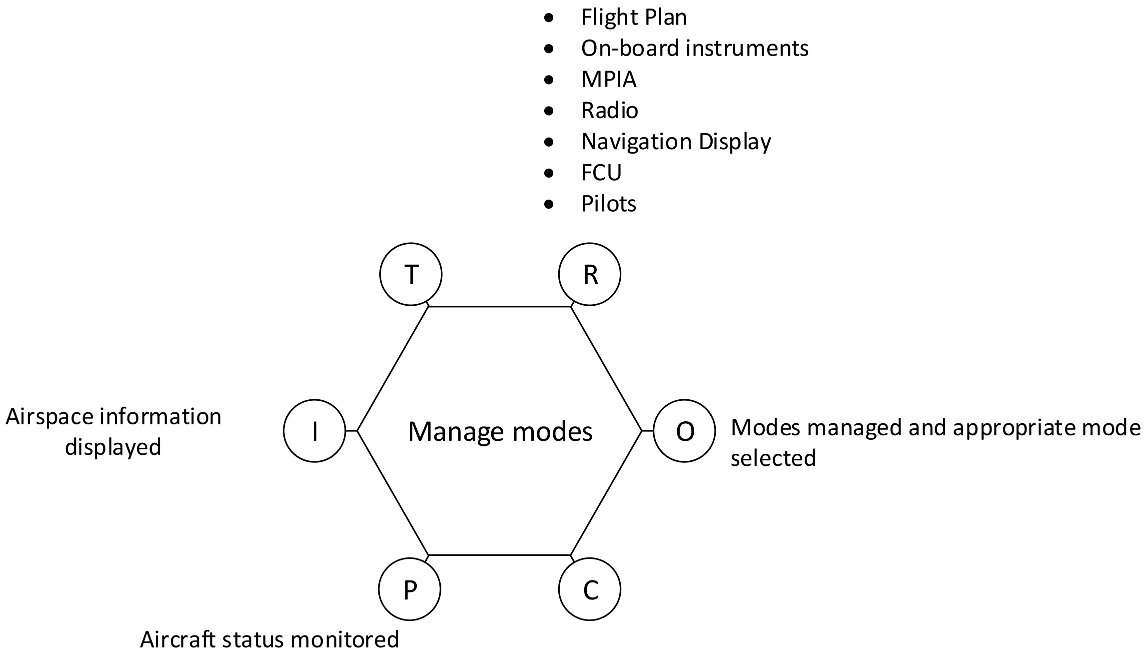


Figure 58: "Manage modes" function as FRAM unit

Table 39 reports the explanation of Input, Output, Precondition, Resource, Control, and Time aspects.

TABLE 39: "MANAGE MODES" FUNCTION FRAM TABLE

MANAGE MODES	
Input(s)	• Airspace information displayed
Output(s)	• Modes managed and appropriate mode selected
Precondition	• Aircraft status monitored
Resource(s)	• Flight plan • On-board instruments • MPIA • Radio • Navigation Display • FCU • Pilots
Control(s)	N/A
Time	*Repeated Function – in parallel with "Manage tilt angle"*

Table 40 and Table 41 report respectively the endogenous and the exogenous factors that can affect the variability of the described function. The impact of these factors can be positive, neutral or negative represented by a green, grey or red arrow respectively.

TABLE 40: ENDOGENOUS FACTORS OF VARIABILITY RELATED TO "MANAGE MODES" FUNCTION

Experience	High (years …) = ↑	Medium (years …) = ↔	N/A (people have been trained and qualified)
Focus of attention	High (level …) = ↑	Medium (level …) = ↔	Low (level …) = ↓
Situation Awareness	High (level …) = ↑	Medium (level …) = ↔	Low (level …) = ↓
Workload	High (level …) = ↓	Medium (level …) = ↔	Low (level …) = ↑
Stress	High (level …) = ↓	Medium (level …) = ↔	Low (level …) = ↑
Complacency	High (level …) = ↓	Medium (level …) = ↔	Low (level …) = ↑
Fatigue	High (level …) = ↓	Medium (level …) = ↔	Low (level …) = ↑
System performance	High (level) = ↑	Medium (level …) = ↔	Low (level …) = ↓
Status of the system	High (level …) = ↑	Medium (level …) = ↔	Low (level …) = ↓
Knowledge	High (No. of procedures/strategies) = ↑	Medium (No. of procedures) = ↔	N/A

TABLE 41: EXOGENOUS FACTORS OF VARIABILITY RELATED TO "MANAGE MODES" FUNCTION

Weather perturbations	High (level …) = ↓	Medium (level …) = ↔	Low (level …) = ↑

These factors can impact on the output variability in terms of time and precision as reported in Table 42.

TABLE 42: OUTPUT VARIABILITY OF "MANAGE MODES" FUNCTION

Modes managed and appropriate mode selected		
Time	Too early	The pilot manages the modes too early and the appropriate one is not successfully selected.
	On time	The pilot manages the modes on time. The appropriate one successfully selected
	Too late	The pilot manages the modes too late. The appropriate one is not successfully selected.
Precision	Imprecise	The pilot manages the modes partially or doesn't pay appropriate attention in managing them. The selected mode is not the appropriate one.
	Acceptable	The pilot manages the modes in an acceptable manner. S/he selects the appropriate one
	Precise	The pilot precisely manages the modes selecting the appropriate one.

The following identified function is "Manage tilt angle" which is itself a macro function consisting of several lower-level functions that are "Edit angle", "Display updated value", and "Change tilt angle". The following figure represents "Edit angle" as FRAM unit.

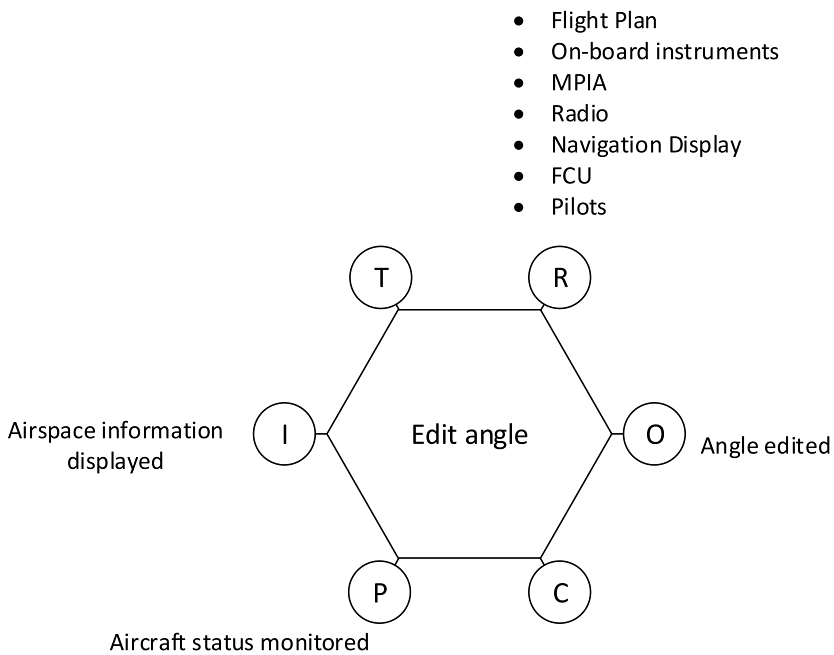


Figure 59: "Edit angle" function as FRAM unit

Table 43 reports the explanation of Input, Output, Precondition, Resource, Control, and Time aspects.

TABLE 43: "EDIT ANGLE" FUNCTION FRAM TABLE

EDIT ANGLE	
Input(s)	• Airspace information displayed
Output(s)	• Angle edited
Precondition	• Aircraft status monitored
Resource(s)	• Flight plan • On-board instruments • MPIA • Radio • Navigation Display • FCU • Pilots
Control(s)	N/A
Time	*On demand*

Table 44 and

Table 45 report respectively the endogenous and the exogenous factors that can affect the variability of the described function. The impact of these factors can be positive, neutral or negative represented by a green, grey or red arrow respectively.

TABLE 44: ENDOGENOUS FACTORS OF VARIABILITY RELATED TO "EDIT ANGLE" FUNCTION

Experience	High (years …) =	Medium (years …) =	N/A (people have been trained and qualified)
Focus of attention	High (level …) =	Medium (level …) =	Low (level …) =
Situation Awareness	High (level …) =	Medium (level …) =	Low (level …) =
Workload	High (level …) =	Medium (level …) =	Low (level …) =
Stress	High (level …) =	Medium (level …) =	Low (level …) =
Complacency	High (level …) =	Medium (level …) =	Low (level …) =
Fatigue	High (level …) =	Medium (level …) =	Low (level …) =
Knowledge	High (No. of procedures/strategies) =	Medium (No. of procedures) =	N/A

TABLE 45: EXOGENOUS FACTORS OF VARIABILITY RELATED TO "EDIT ANGLE" FUNCTION

Weather perturbations	High (level …) =	Medium (level …) =	Low (level …) =

These factors can impact on the output variability in terms of time and precision as reported in Table 46.

TABLE 46: OUTPUT VARIABILITY OF "EDIT ANGLE" FUNCTION

Angle edited		
Time	Too early	N/A
	On time	The pilot edits the angle on time.
	Too late	The pilot edits the angle too late. The value cannot be elaborated by the WXR application.
Precision	Imprecise	The pilot edits the angle partially or doesn't pay appropriate attention in editing the angle. The edited angle is not the appropriate one.
	Acceptable	The pilot edits the angle in an acceptable manner. S/he selects the appropriate one
	Precise	The pilot precisely edits the angle selecting the appropriate one.

The following identified function is "Display updated value" which is represented as FRAM unit in the figure below.

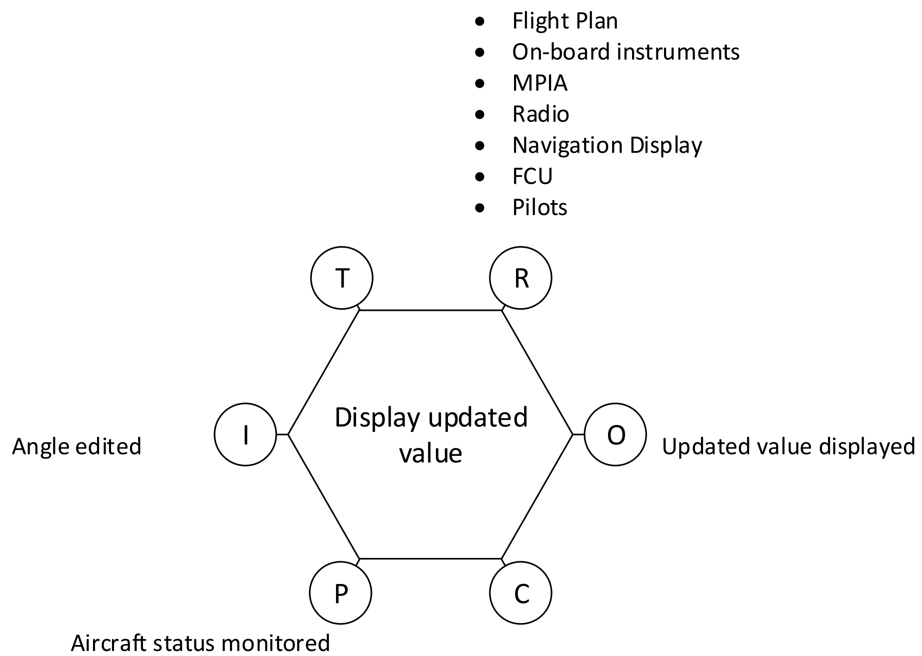


Figure 60: "Display updated value" function as FRAM unit

Table 47**Table 43** reports the explanation of Input, Output, Precondition, Resource, Control, and Time aspects.

Table 47: "Display updated value" function FRAM table

DISPLAY UPDATED VALUE	
Input(s)	• Angle edited
Output(s)	• Updated value displayed
Precondition	• Aircraft status monitored
Resource(s)	• Flight plan • On-board instruments • MPIA • Radio • Navigation Display • FCU • Pilots
Control(s)	N/A
Time	*On demand*

Table 48 and Table 49 report respectively the endogenous and the exogenous factors that can affect the variability of the described function. The impact of these factors can be positive, neutral or negative represented by a green, grey or red arrow respectively.

Table 48: Endogenous factors of variability related to "Display updated value" function

System performance	High (level) = ↑	Medium (level …) = ↔	Low (level …) = ↓
Status of the system	High (level …) = ↑	Medium (level …) = ↔	Low (level …) = ↓

Table 49: Exogenous factors of variability related to "Display updated value" function

Weather perturbations	High (level …) = ↓	Medium (level …) = ↔	Low (level …) = ↑

These factors can impact on the output variability in terms of time and precision as reported in Table 50.

TABLE 50: OUTPUT VARIABILITY OF "DISPLAY UPDATED VALUE" FUNCTION

Updated value displayed		
Time	Too early	N/A
	On time	The displayed value is successfully updated
	Too late	The displayed value is not updated in time
Precision	Imprecise	The displayed value is not correctly updated
	Acceptable	The displayed value is correctly updated
	Precise	The displayed value is correctly updated

The following identified function is "Change tilt angle" which is represented in Figure 61 as FRAM unit.

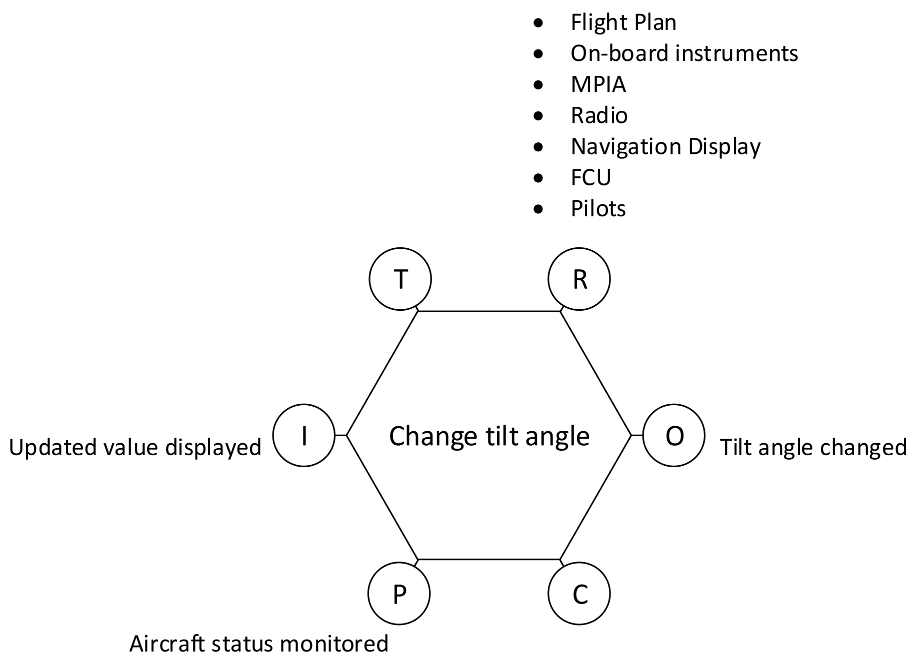


Figure 61: "Change tilt angle" function as FRAM unit

Table 63 reports the explanation of Input, Output, Precondition, Resource, Control, and Time aspects.

TABLE 51: "CHANGE TILT ANGLE" FUNCTION FRAM TABLE

CHANGE TILT ANGLE	
Input(s)	• Updated value displayed
Output(s)	• Tilt angle changed
Precondition	• Aircraft status monitored
Resource(s)	• Flight plan • On-board instruments • MPIA • Radio • Navigation Display • FCU • Pilots
Control(s)	N/A
Time	*On demand*

Table 64 and Table 65 report respectively the endogenous and the exogenous factors that can affect the variability of the described function. The impact of these factors can be positive, neutral or negative represented by a green, grey or red arrow respectively.

TABLE 52: ENDOGENOUS FACTORS OF VARIABILITY RELATED TO "CHANGE TILT ANGLE" FUNCTION

Experience	High (years …) =	Medium (years …) =	N/A (people have been trained and qualified)
Focus of attention	High (level …) =	Medium (level …) =	Low (level …) =
Situation Awareness	High (level …) =	Medium (level …) =	Low (level …) =
Workload	High (level …) =	Medium (level …) =	Low (level …) =
Stress	High (level …) =	Medium (level …) =	Low (level …) =
Complacency	High (level …) =	Medium (level …) =	Low (level …) =
Fatigue	High (level …) =	Medium (level …) =	Low (level …) =
System performance	High (level) =	Medium (level …) =	Low (level …) =
Status of the system	High (level …) =	Medium (level …) =	Low (level …) =
Knowledge	High (No. of procedures/strategies) =	Medium (No. of procedures) =	N/A

TABLE 53: EXOGENOUS FACTORS OF VARIABILITY RELATED TO "CHANGE TILT ANGLE" FUNCTION

Weather perturbations	High (level …) =	Medium (level …) =	Low (level …) =

These factors can impact on the output variability in terms of time and precision as reported in Table 66.

TABLE 54: OUTPUT VARIABILITY OF "CHANGE TILT ANGLE" FUNCTION

Tilt angle changed		
Time	Too early	N/A
	On time	The pilot changes the angle on time.
	Too late	The pilot changes the angle too late.
Precision	Imprecise	The pilot doesn't pay appropriate attention in changing the angle. The changed one is not the right.
	Acceptable	The pilot changes the angle in an acceptable manner. S/he selects the right one
	Precise	The pilot precisely changes the angle selecting the right one.

Figure 62 represents the "Analyse current weather map" function as FRAM unit.

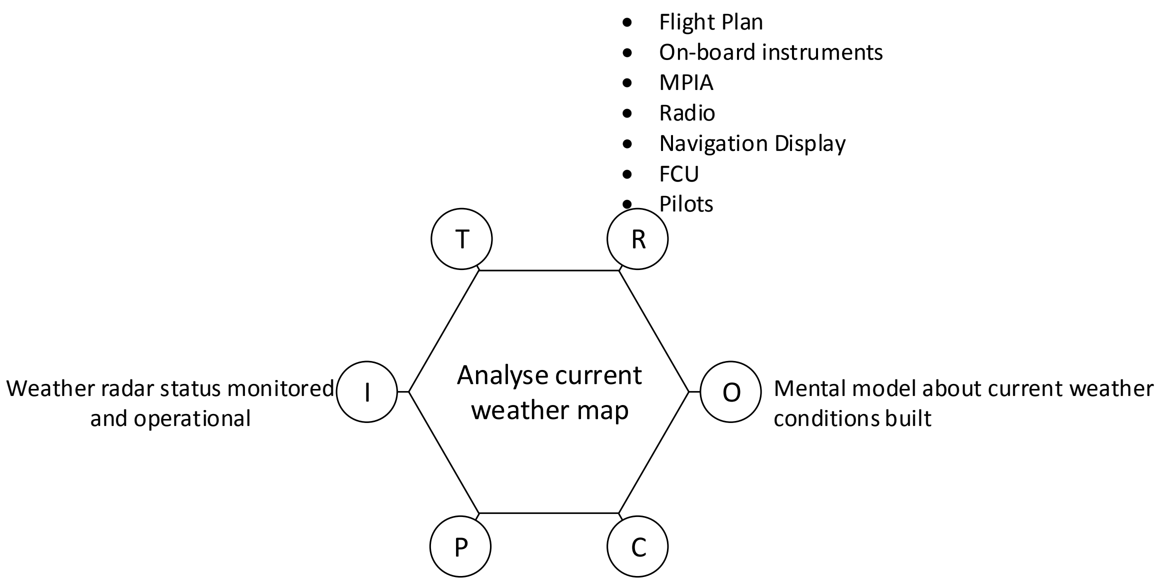


Figure 62: "Analyse current weather map" function as FRAM unit

Table 55 reports the explanation of Input, Output, Precondition, Resource, Control, and Time aspects.

TABLE 55: "ANALYSE CURRENT WEATHER MAP" FUNCTION FRAM TABLE

ANALYSE CURRENT WEATHER MAP	
Input(s)	• Weather radar status monitored and operational
Output(s)	• Mental model about current weather conditions built
Precondition	N/A
Resource(s)	• Flight plan • On-board instruments • MPIA • Radio • Navigation Display • FCU • Pilots
Control(s)	N/A
Time	*Repeated Function – competing with other functions*

Table 56 and Table 69 report respectively the endogenous and the exogenous factors that can affect the variability of the described function. The impact of these factors can be positive, neutral or negative represented by a green, grey or red arrow respectively.

TABLE 56: ENDOGENOUS FACTORS OF VARIABILITY RELATED TO "ANALYSE CURRENT WEATHER MAP" FUNCTION

Experience	High (years …) =	Medium (years …) =	N/A (people have been trained and qualified)
Focus of attention	High (level …) =	Medium (level …) =	Low (level …) =
Situation Awareness	High (level …) =	Medium (level …) =	Low (level …) =
Workload	High (level …) =	Medium (level …) =	Low (level …) =
Stress	High (level …) =	Medium (level …) =	Low (level …) =
Complacency	High (level …) =	Medium (level …) =	Low (level …) =
Fatigue	High (level …) =	Medium (level …) =	Low (level …) =
Knowledge	High (No. of procedures/strategies) =	Medium (No. of procedures) =	N/A

TABLE 57: EXOGENOUS FACTORS OF VARIABILITY RELATED TO "ANALYSE CURRENT WEATHER MAP" FUNCTION

Weather perturbations	High (level …) = ⬇	Medium (level …) = ⬌	Low (level …) = ⬆

These factors can impact on the output variability in terms of time and precision as reported in Table 58.

TABLE 58: OUTPUT VARIABILITY OF "ANALYSE CURRENT WEATHER MAP" FUNCTION

Mental model about current weather conditions built		
Time	Too early	The pilot builds her/his mental model too early without taking into account the current information.
	On time	The pilot builds her/his mental model on time taking into account the current information.
	Too late	The pilot builds her/his mental model too late taking into account dated information.
Precision	Imprecise	The pilot builds her/his mental model taking into account only some information
	Acceptable	The pilot builds her/his mental model in an acceptable manner
	Precise	The pilot precisely builds her/his mental model

Figure 63 represents the "Select different range map" function as FRAM unit.

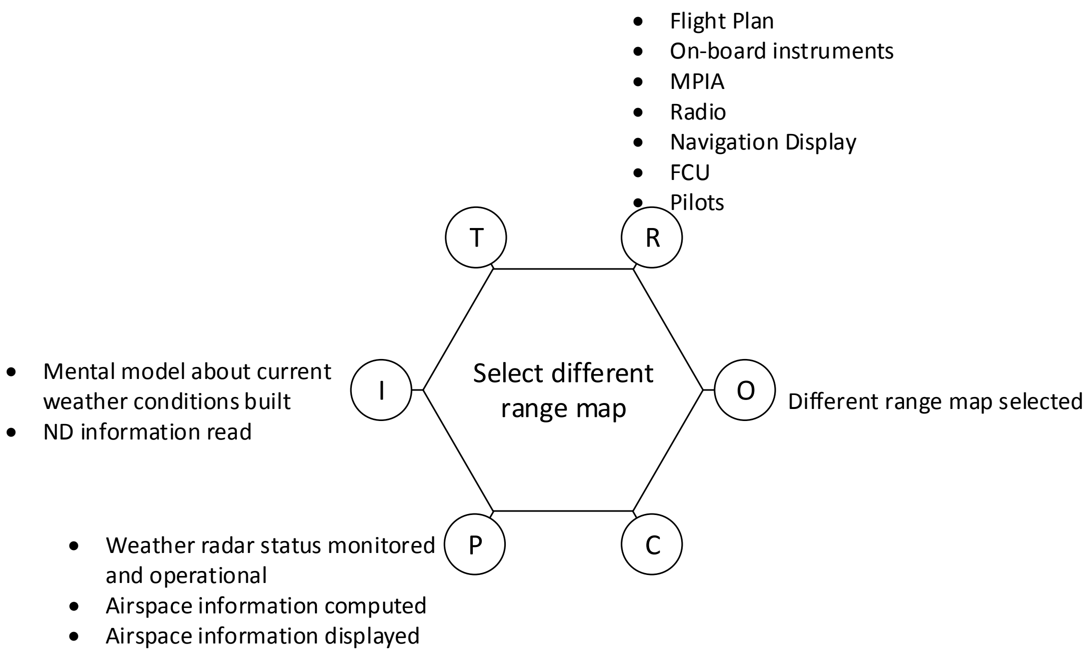


Figure 63: "Select different range map" function as FRAM unit

Table 59 reports the explanation of Input, Output, Precondition, Resource, Control, and Time aspects.

TABLE 59: "SELECT DIFFERENT RANGE MAP" FUNCTION FRAM TABLE

SELECT DIFFERENT RANGE MAP	
Input(s)	• ND information read • Mental model about current weather conditions built
Output(s)	• Different range map selected
Precondition	• Weather radar status monitored and operational • Airspace information computed • Airspace information displayed
Resource(s)	• Flight plan • On-board instruments • MPIA • Radio • Navigation Display • FCU • Pilots
Control(s)	N/A
Time	*Repeated Function*

Table 60 and Table 61 report respectively the endogenous and the exogenous factors that can affect the variability of the described function. The impact of these factors can be positive, neutral or negative represented by a green, grey or red arrow respectively.

TABLE 60: ENDOGENOUS FACTORS OF VARIABILITY RELATED TO "SELECT DIFFERENT RANGE MAP" FUNCTION

Experience	High (years …) =	Medium (years …) =	N/A (people have been trained and qualified)
Focus of attention	High (level …) =	Medium (level …) =	Low (level …) =
Situation Awareness	High (level …) =	Medium (level …) =	Low (level …) =
Workload	High (level …) =	Medium (level …) =	Low (level …) =
Stress	High (level …) =	Medium (level …) =	Low (level …) =
Complacency	High (level …) =	Medium (level …) =	Low (level …) =
Fatigue	High (level …) =	Medium (level …) =	Low (level …) =
System performance	High (level) =	Medium (level …) =	Low (level …) =
Status of the system	High (level …) =	Medium (level …) =	Low (level …) =
Knowledge	High (No. of procedures/strategies) =	Medium (No. of procedures) =	N/A

TABLE 61: EXOGENOUS FACTORS OF VARIABILITY RELATED TO "SELECT DIFFERENT RANGE MAP" FUNCTION

Weather perturbations	High (level …) =	Medium (level …) =	Low (level …) =

These factors can impact on the output variability in terms of time and precision as reported in Table 62TABLE 58.

TABLE 62: OUTPUT VARIABILITY OF "SELECT DIFFERENT RANGE MAP" FUNCTION

Different range map selected		
Time	Too early	The pilot selects the different range map too early without taking into account the previous range.
	On time	The pilot selects the different range map on time taking into account the previous range.
	Too late	The pilot selects the different range map too late and s/he needs to select again the previous one.
Precision	Imprecise	The pilot selects the different range map but doesn't pay appropriate attention to the selection of the range
	Acceptable	The pilot selects the different range s/he was targeting.
	Precise	The pilot takes care of ensuring that s/he selects the targeted range.

Figure 64 represents the "Detect weather target" function as FRAM unit.

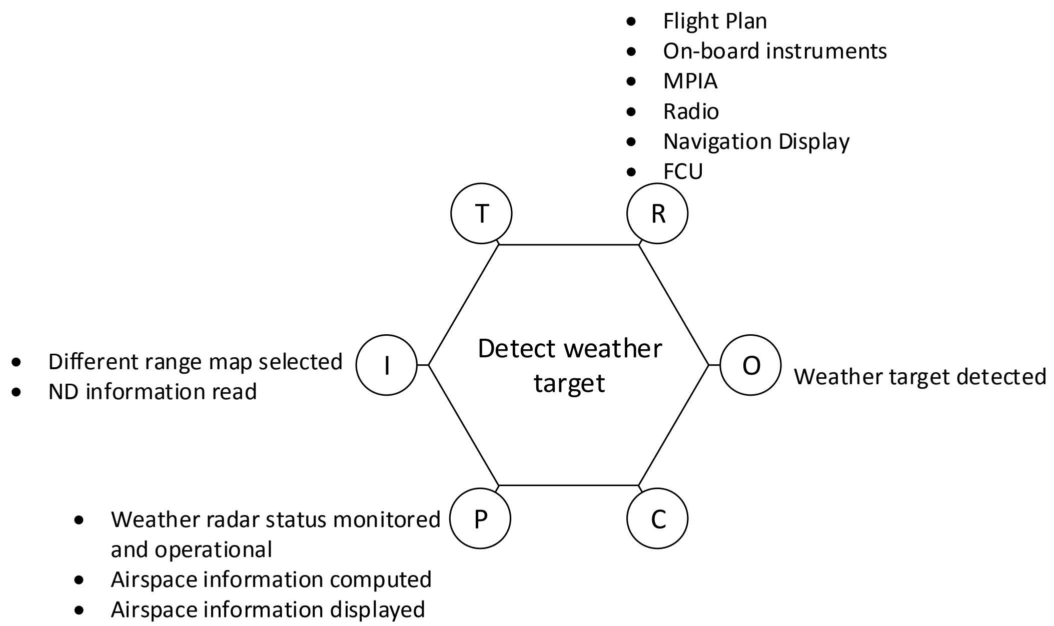


Figure 64: "Detect weather target" function as FRAM unit

Table 63 reports the explanation of Input, Output, Precondition, Resource, Control, and Time aspects.

TABLE 63: "DETECT WEATHER TARGET" FUNCTION FRAM TABLE

DETECT WEATHER TARGET	
Input(s)	• Different range map selected • ND information read
Output(s)	• Weather target detected
Precondition	• Weather radar status monitored and operational • Airspace information computed • Airspace information displayed
Resource(s)	• Flight plan • On-board instruments • MPIA • Radio • Navigation Display • FCU
Control(s)	N/A
Time	*Continuous Function – competing with other functions*

Table 64 and Table 65 report respectively the endogenous and the exogenous factors that can affect the variability of the described function. The impact of these factors can be positive, neutral or negative represented by a green, grey or red arrow respectively.

TABLE 64: ENDOGENOUS FACTORS OF VARIABILITY RELATED TO "DETECT WEATHER TARGET" FUNCTION

System performance	High (level) =	Medium (level …) =	Low (level …) =
Status of the system	High (level …) =	Medium (level …) =	Low (level …) =

TABLE 65: EXOGENOUS FACTORS OF VARIABILITY RELATED TO "DETECT WEATHER TARGET" FUNCTION

Weather perturbations	High (level …) =	Medium (level …) =	Low (level …) =

These factors can impact on the output variability in terms of time and precision as reported in Table 66.

TABLE 66: OUTPUT VARIABILITY OF "DETECT WEATHER TARGET" FUNCTION

Weather target detected		
Time	Too early	N/A
	On time	The weather target are successfully detected
	Too late	The weather target are not detected in time
Precision	Imprecise	The weather target are not correctly detected
	Acceptable	The weather target are correctly detected
	Precise	The weather target are correctly detected

Figure 65 represents the "Recognise weather target" function as FRAM unit.

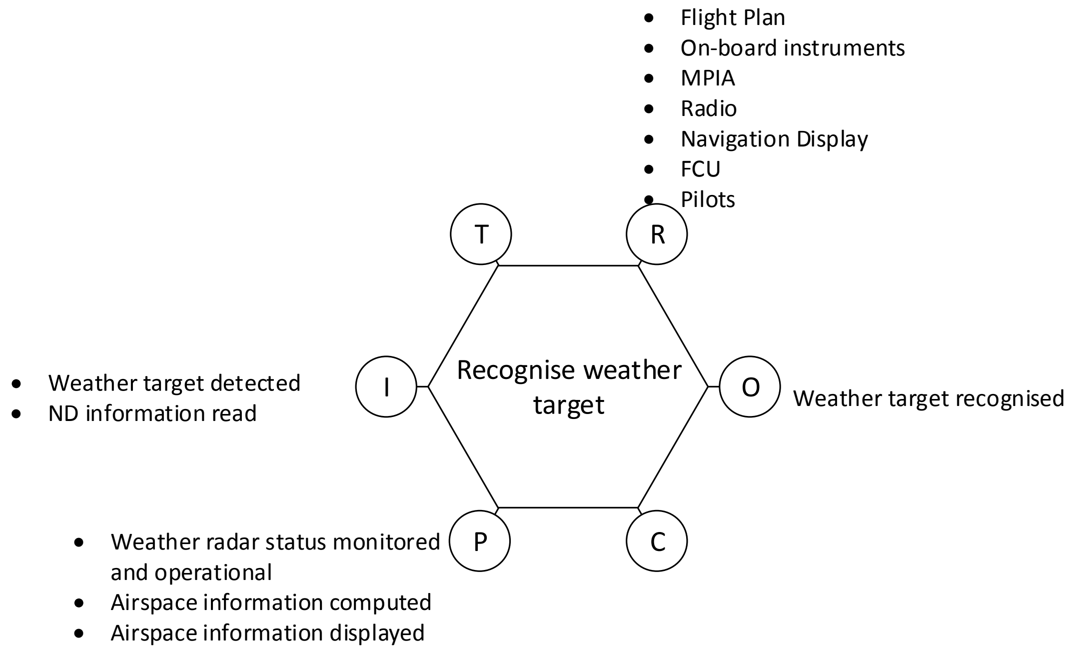


Figure 65: "Recognise weather target" function as FRAM unit

Table 67 reports the explanation of Input, Output, Precondition, Resource, Control, and Time aspects.

TABLE 67: "RECOGNISE WEATHER TARGET" FUNCTION FRAM TABLE

RECOGNISE WEATHER TARGET	
Input(s)	• Weather target detected • ND information read
Output(s)	• Weather target recognised
Precondition	• Weather radar status monitored and operational • Airspace information computed • Airspace information displayed
Resource(s)	• Flight plan • On-board instruments • MPIA • Radio • Navigation Display • FCU • Pilots
Control(s)	N/A
Time	*Repeated Function – competing with other functions*

Table 68 and Table 69 report respectively the endogenous and the exogenous factors that can affect the variability of the described function. The impact of these factors can be positive, neutral or negative represented by a green, grey or red arrow respectively.

TABLE 68: ENDOGENOUS FACTORS OF VARIABILITY RELATED TO "RECOGNISE WEATHER TARGET" FUNCTION

Experience	High (years …) =	Medium (years …) =	N/A (people have been trained and qualified)
Focus of attention	High (level …) =	Medium (level …) =	Low (level …) =
Situation Awareness	High (level …) =	Medium (level …) =	Low (level …) =
Workload	High (level …) =	Medium (level …) =	Low (level …) =
Stress	High (level …) =	Medium (level …) =	Low (level …) =
Complacency	High (level …) =	Medium (level …) =	Low (level …) =
Fatigue	High (level …) =	Medium (level …) =	Low (level …) =
Knowledge	High (No. of procedures/strategies) =	Medium (No. of procedures) =	N/A

TABLE 69: EXOGENOUS FACTORS OF VARIABILITY RELATED TO "RECOGNISE WEATHER TARGET" FUNCTION

Weather perturbations	High (level …) =	Medium (level …) =	Low (level …) =

These factors can impact on the output variability in terms of time and precision as reported in Table 70.

TABLE 70: OUTPUT VARIABILITY OF "RECOGNISE WEATHER TARGET" FUNCTION

Weather target recognised		
Time	Too early	N/A
	On time	The pilot recognises the weather target on time for applying the adequate procedure.
	Too late	The pilot recognises the weather target too late for applying the adequate procedure.
Precision	Imprecise	The pilot partially recognises the weather target or doesn't pay appropriate attention to the recognition of the weather target
	Acceptable	The pilot recognises the weather target in an acceptable manner
	Precise	The pilot precisely recognises the weather target

Figure 66 represents the "Identify the adequate procedure" function as FRAM unit.

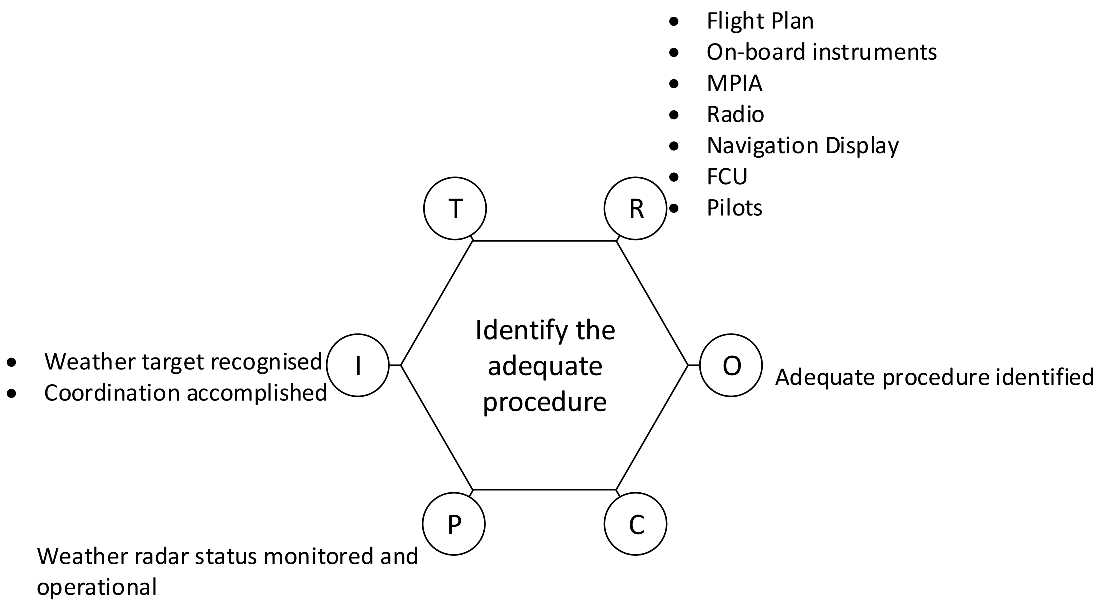


Figure 66: "Identify the adequate procedure" function as FRAM unit

Table 71 reports the explanation of Input, Output, Precondition, Resource, Control, and Time aspects.

TABLE 71: "IDENTIFY THE ADEQUATE PROCEDURE" FUNCTION FRAM TABLE

IDENTIFY THE ADEQUATE PROCEDURE	
Input(s)	• Weather target recognised • Coordination accomplished
Output(s)	• Adequate procedure identified
Precondition	• Weather radar status monitored and operational
Resource(s)	• Flight plan • On-board instruments • MPIA • Radio • Navigation Display • FCU • Pilots
Control(s)	N/A
Time	*Continuous Function*

Table 72 and Table 73 report respectively the endogenous and the exogenous factors that can affect the variability of the described function. The impact of these factors can be positive, neutral or negative represented by a green, grey or red arrow respectively.

TABLE 72: ENDOGENOUS FACTORS OF VARIABILITY RELATED TO "IDENTIFY THE ADEQUATE PROCEDURE" FUNCTION

Experience	High (years …) = ⇧	Medium (years …) = ⇔	N/A (people have been trained and qualified)
Focus of attention	High (level …) = ⇧	Medium (level …) = ⇔	Low (level …) = ⇩
Situation Awareness	High (level …) = ⇧	Medium (level …) = ⇔	Low (level …) = ⇩
Workload	High (level …) = ⇩	Medium (level …) = ⇔	Low (level …) = ⇧
Stress	High (level …) = ⇩	Medium (level …) = ⇔	Low (level …) = ⇧
Complacency	High (level …) = ⇩	Medium (level …) = ⇔	Low (level …) = ⇧
Fatigue	High (level …) = ⇩	Medium (level …) = ⇔	Low (level …) = ⇧
Knowledge	High (No. of procedures/strategies) = ⇧	Medium (No. of procedures) = ⇔	N/A

TABLE 73: EXOGENOUS FACTORS OF VARIABILITY RELATED TO "APPLY THE ADEQUATE PROCEDURE" FUNCTION

Weather perturbations	High (level …) =	Medium (level …) =	Low (level …) =

These factors can impact on the output variability in terms of time and precision as reported in Table 74.

TABLE 74: OUTPUT VARIABILITY OF "IDENTIFY THE ADEQUATE PROCEDURE" FUNCTION

Adequate procedure identified		
Time	Too early	N/A
	On time	The pilot identifies the adequate procedure on time.
	Too late	The pilot identifies the adequate procedure too late for avoiding the recognised weather target.
Precision	Imprecise	The pilot partially identifies the adequate procedure or doesn't pay appropriate attention to the identification of the adequate procedure
	Acceptable	The pilot identifies the adequate procedure in an acceptable manner
	Precise	The pilot precisely identifies the adequate procedure

Figure 67 represents the "Request a clearance" function as FRAM unit.

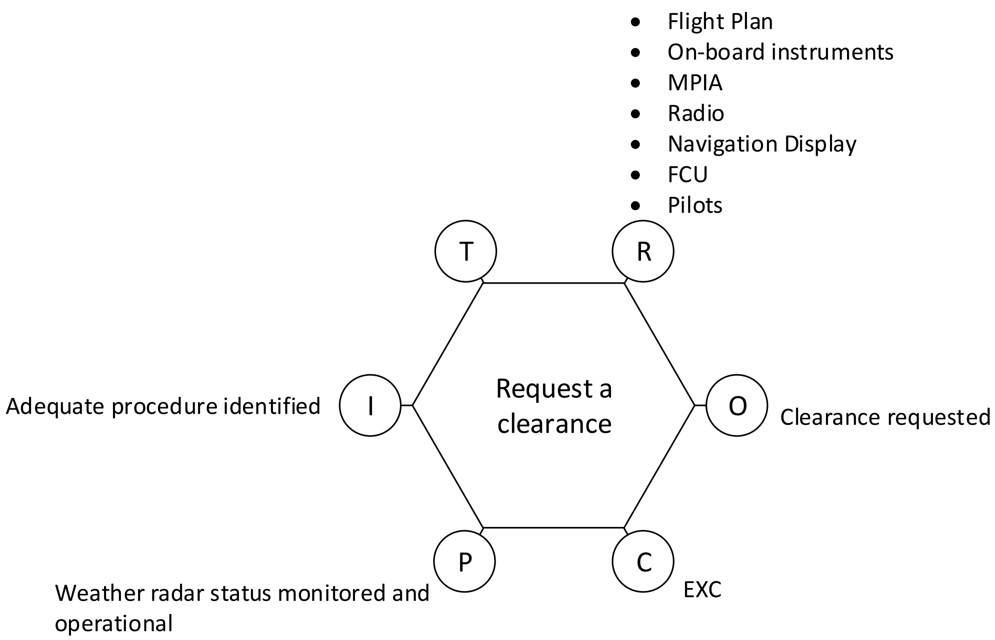


Figure 67: "Request a clearance" function as FRAM unit

Table 75 reports the explanation of Input, Output, Precondition, Resource, Control, and Time aspects.

TABLE 75: "REQUEST A CLEARANCE" FUNCTION FRAM TABLE

REQUEST A CLEARANCE	
Input(s)	• Adequate procedure identified
Output(s)	• Clearance requested
Precondition	• Weather radar status monitored and operational
Resource(s)	• Flight plan • On-board instruments • MPIA • Radio • Navigation Display • FCU • Pilots
Control(s)	• EXC
Time	*On demand*

Table 76 and

Table 77 report respectively the endogenous and the exogenous factors that can affect the variability of the described function. The impact of these factors can be positive, neutral or negative represented by a green, grey or red arrow respectively.

TABLE 76: ENDOGENOUS FACTORS OF VARIABILITY RELATED TO "REQUEST A CLEARANCE" FUNCTION

Experience	High (years …) =	Medium (years …) =	N/A (people have been trained and qualified)
Focus of attention	High (level …) =	Medium (level …) =	Low (level …) =
Situation Awareness	High (level …) =	Medium (level …) =	Low (level …) =
Workload	High (level …) =	Medium (level …) =	Low (level …) =
Stress	High (level …) =	Medium (level …) =	Low (level …) =
Complacency	High (level …) =	Medium (level …) =	Low (level …) =
Fatigue	High (level …) =	Medium (level …) =	Low (level …) =
Knowledge	High (No. of procedures/strategies) =	Medium (No. of procedures) =	N/A

TABLE 77: EXOGENOUS FACTORS OF VARIABILITY RELATED TO "REQUEST A CLEARANCE" FUNCTION

Weather perturbations	High (level …) =	Medium (level …) =	Low (level …) =

These factors can impact on the output variability in terms of time and precision as reported in Table 78.

TABLE 78: OUTPUT VARIABILITY OF "REQUEST A CLEARANCE" FUNCTION

Clearance requested		
Time	Too early	N/A
	On time	The pilot requests the clearance on time.
	Too late	The pilot requests the clearance too late for being managed by the EXC.
Precision	Imprecise	The pilot doesn't pay appropriate attention to the clearance request
	Acceptable	The pilot applies requests the clearance in an acceptable manner
	Precise	The pilot precisely requests the clearance

Figure 68 represents the "Coordinate with EXC" function as FRAM unit.

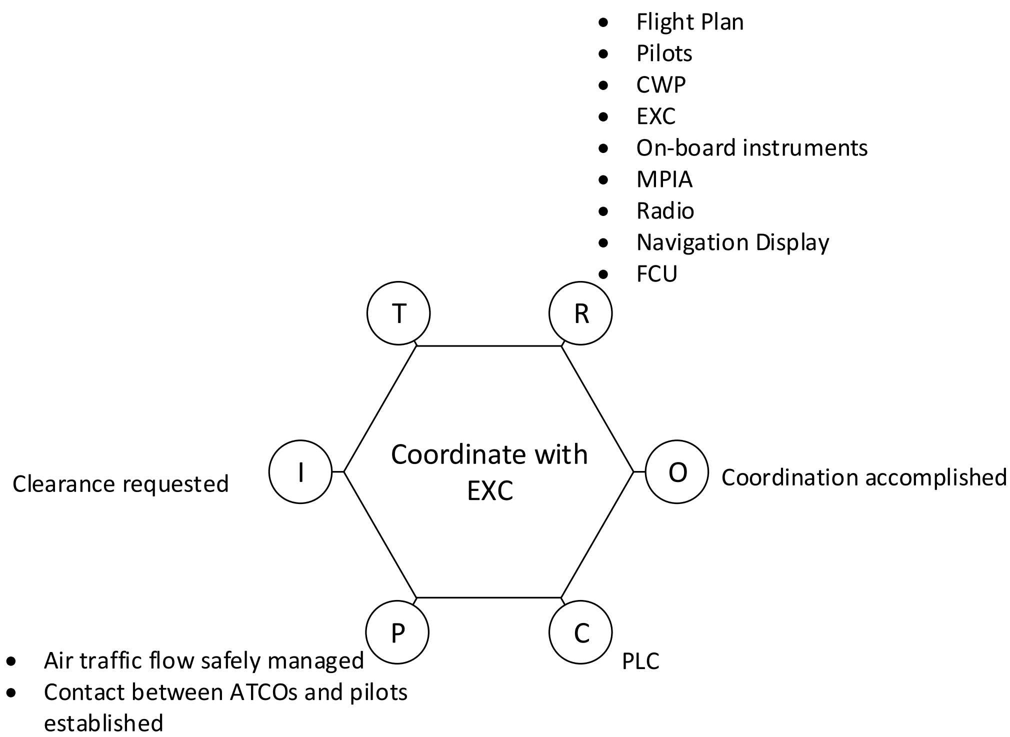


Figure 68: "Coordinate with EXC" function as FRAM unit

Table 79 reports the explanation of Input, Output, Precondition, Resource, Control, and Time aspects.

TABLE 79: "COORDINATE WITH EXC" FUNCTION FRAM TABLE

COORDINATE WITH EXC	
Input(s)	• Clearance requested
Output(s)	• Coordination accomplished
Precondition	• Air traffic flow safely managed • Contact between ATCOs and pilots established
Resource(s)	• Flight Plan • Pilots • CWP • EXC • On-board instruments • MPIA • Radio • Navigation Display • FCU
Control(s)	• PLC
Time	*Repeated function, every time the procedures require the coordination (aircraft assumptions/transfer – clearance)*

Table 80 and Table 81 report respectively the endogenous and the exogenous factors that can affect the variability of the described function. The impact of these factors can be positive, neutral or negative represented by a green, grey or red arrow respectively.

TABLE 80: ENDOGENOUS FACTORS OF VARIABILITY RELATED TO "COORDINATE WITH EXC" FUNCTION

Experience	High (years …) =	Medium (years …) =	N/A (people have been trained and qualified)
Focus of attention	High (level …) =	Medium (level …) =	Low (level …) =
Situation Awareness	High (level …) =	Medium (level …) =	Low (level …) =
Workload	High (level …) =	Medium (level …) =	Low (level …) =
Stress	High (level …) =	Medium (level …) =	Low (level …) =
Complacency	High (level …) =	Medium (level …) =	Low (level …) =
Fatigue	High (level …) =	Medium (level …) =	Low (level …) =
Knowledge	High (No. of procedures/strategies) =	Medium (No. of procedures) =	N/A

TABLE 81: EXOGENOUS FACTORS OF VARIABILITY RELATED TO "COORDINATE WITH EXC" FUNCTION

N° Aircraft	High (n …) = ↓	Medium (n …) = ↔	Low (n …) = ↑
Complexity of the surrounding air traffic	High (level …) = ↓	Medium (level …) = ↔	Low (level …) = ↑

These factors can impact on the output variability in terms of time and precision as reported in Table 82.

TABLE 82: OUTPUT VARIABILITY OF "REQUEST A CLEARANCE" FUNCTION

Coordination accomplished		
Time	Too early	N/A
	On time	The coordination between the EXC and the pilot is accomplished on time.
	Too late	The coordination between the EXC and the pilot is accomplished too late for the clearance acknowledging
Precision	Imprecise	The coordination between the EXC and the pilot is partially accomplished and the clearance acknowledged is imprecise
	Acceptable	The coordination between the EXC and the pilot is accomplished in an acceptable manner
	Precise	The coordination between the EXC and the pilot is precisely accomplished through clearance acknowledged

Figure 69 represents the "Execute the clearance" function as FRAM unit.

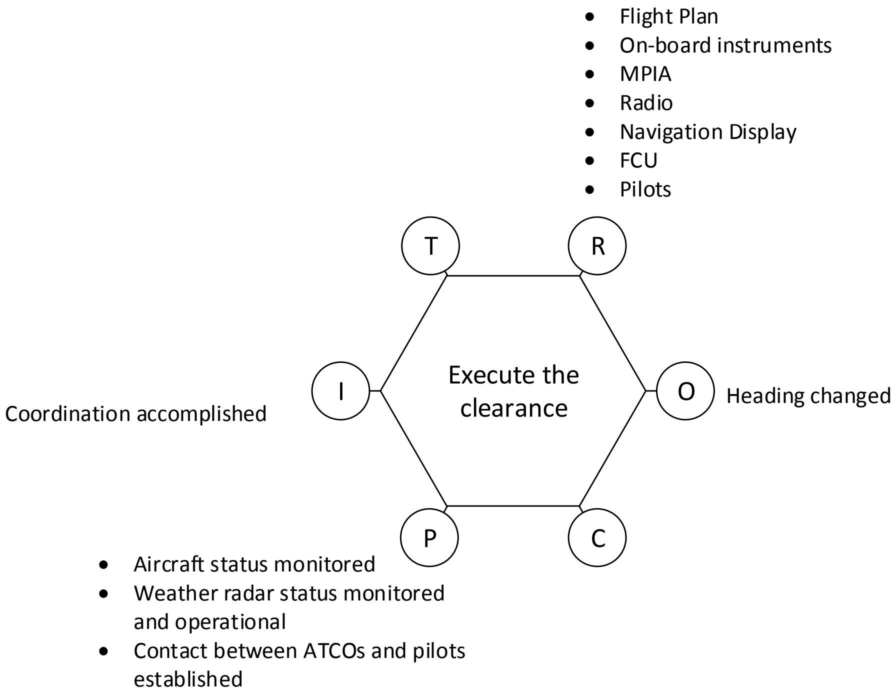


Figure 69: "Execute the clearance" function as FRAM unit

Table 83 reports the explanation of Input, Output, Precondition, Resource, Control, and Time aspects.

TABLE 83: "EXECUTE THE CLEARANCE" FUNCTION FRAM TABLE

EXECUTE THE CLEARANCE	
Input(s)	• Coordination accomplished
Output(s)	• Heading changed
Precondition	• Aircraft status monitored • Weather radar status monitored and operational • Contact between ATCOs and pilots established
Resource(s)	• Flight plan • On-board instruments • MPIA • Radio • Navigation Display • FCU • Pilots
Control(s)	• EXC
Time	*On demand – competing with other functions*

Table 84 and Table 85 report respectively the endogenous and the exogenous factors that can affect the variability of the described function. The impact of these factors can be positive, neutral or negative represented by a green, grey or red arrow respectively.

TABLE 84: ENDOGENOUS FACTORS OF VARIABILITY RELATED TO "EXECUTE THE CLEARANCE" FUNCTION

Experience	High (years …) = ↑	Medium (years …) = ↔	N/A (people have been trained and qualified)
Focus of attention	High (level …) = ↑	Medium (level …) = ↔	Low (level …) = ↓
Situation Awareness	High (level …) = ↑	Medium (level …) = ↔	Low (level …) = ↓
Workload	High (level …) = ↓	Medium (level …) = ↔	Low (level …) = ↑
Stress	High (level …) = ↓	Medium (level …) = ↔	Low (level …) = ↑
Complacency	High (level …) = ↓	Medium (level …) = ↔	Low (level …) = ↑
Fatigue	High (level …) = ↓	Medium (level …) = ↔	Low (level …) = ↑
System performance	High (level) = ↑	Medium (level …) = ↔	Low (level …) = ↓
Status of the system	High (level …) = ↑	Medium (level …) = ↔	Low (level …) = ↓
Knowledge	High (years …) = ↑	Medium (years …) = ↔	N/A

TABLE 85: EXOGENOUS FACTORS OF VARIABILITY RELATED TO "EXECUTE THE CLEARANCE" FUNCTION

Weather perturbations	High (level …) = ↓	Medium (level …) = ↔	Low (level …) = ↑

These factors can impact on the output variability in terms of time and precision as reported in Table 86.

TABLE 86: OUTPUT VARIABILITY OF "EXECUTE THE CLEARANCE" FUNCTION

Heading changed		
Time	Too early	N/A
	On time	The pilot changes the heading on time for avoiding the weather target
	Too late	The pilot changes the heading too late for avoiding the weather target.
Precision	Imprecise	The pilot doesn’t pay appropriate attention to the heading change
	Acceptable	The pilot changes the heading in an acceptable manner.
	Precise	The pilot takes care of ensuring that s/he changes the heading

Once all the identified functions have been represented as FRAM units, the endogenous and exogenous factors have been listed and their potential impact on the function output variability has been expressed in terms of time and precision, it is possible to translate the Timeline as FRAM instantiation. The instantiation supports in preliminary detecting the functional resonance between the functions.

Figure 70 illustrates the FRAM instantiation of the WXR case study. The instantiation shows how the functions are coupled between them. These couplings allow us to preliminary detect the functional resonance and how they can be affected from its propagation. In addition, Figure 71 and Figure 72 respectively illustrate the FRAM instantiations of the “Check weather conditions” which is a macro function consisting of a set of lower-functions and of the “Manage tilt angle" which is a lower-function of “Manage weather radar status” macro function. The instantiation shows how the functions are coupled between them. These couplings allow us to preliminary detect the functional resonance and how they can be affected from its propagation.

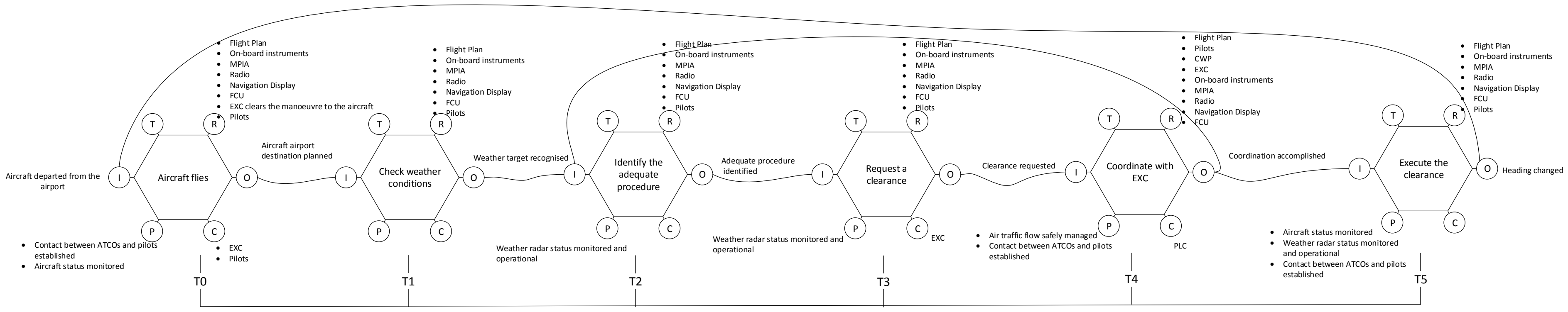


Figure 70: FRAM instantiation of the WXR case study

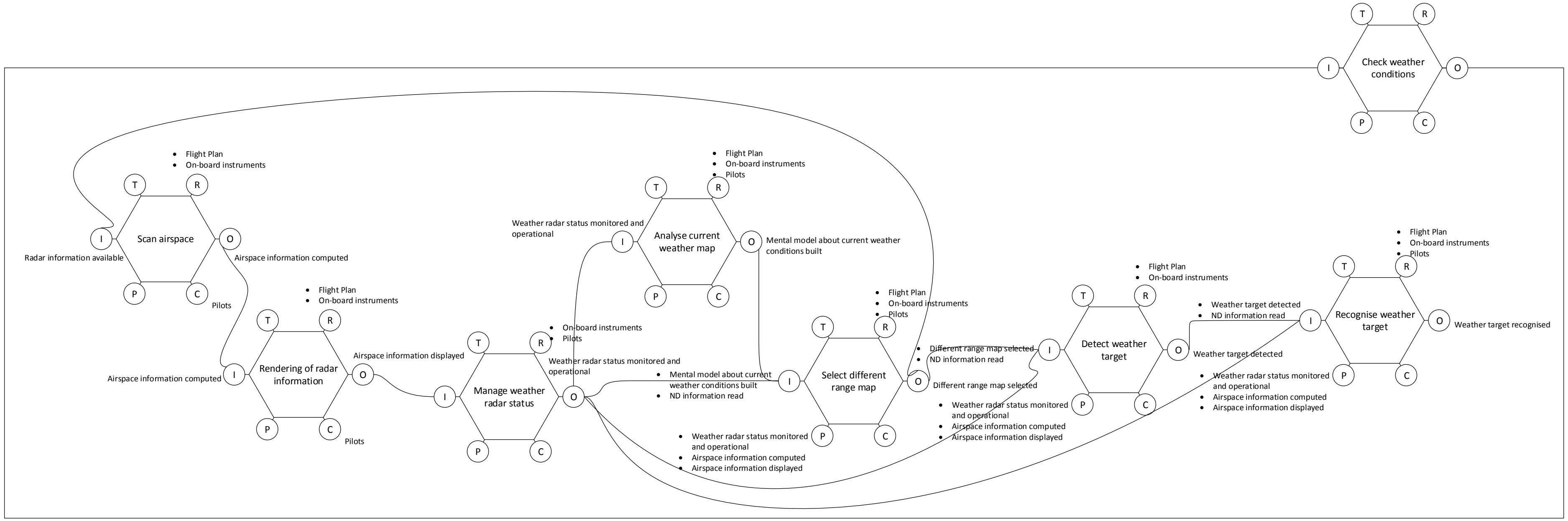


Figure 71: FRAM instantiation of the "Check weather conditions" macro function

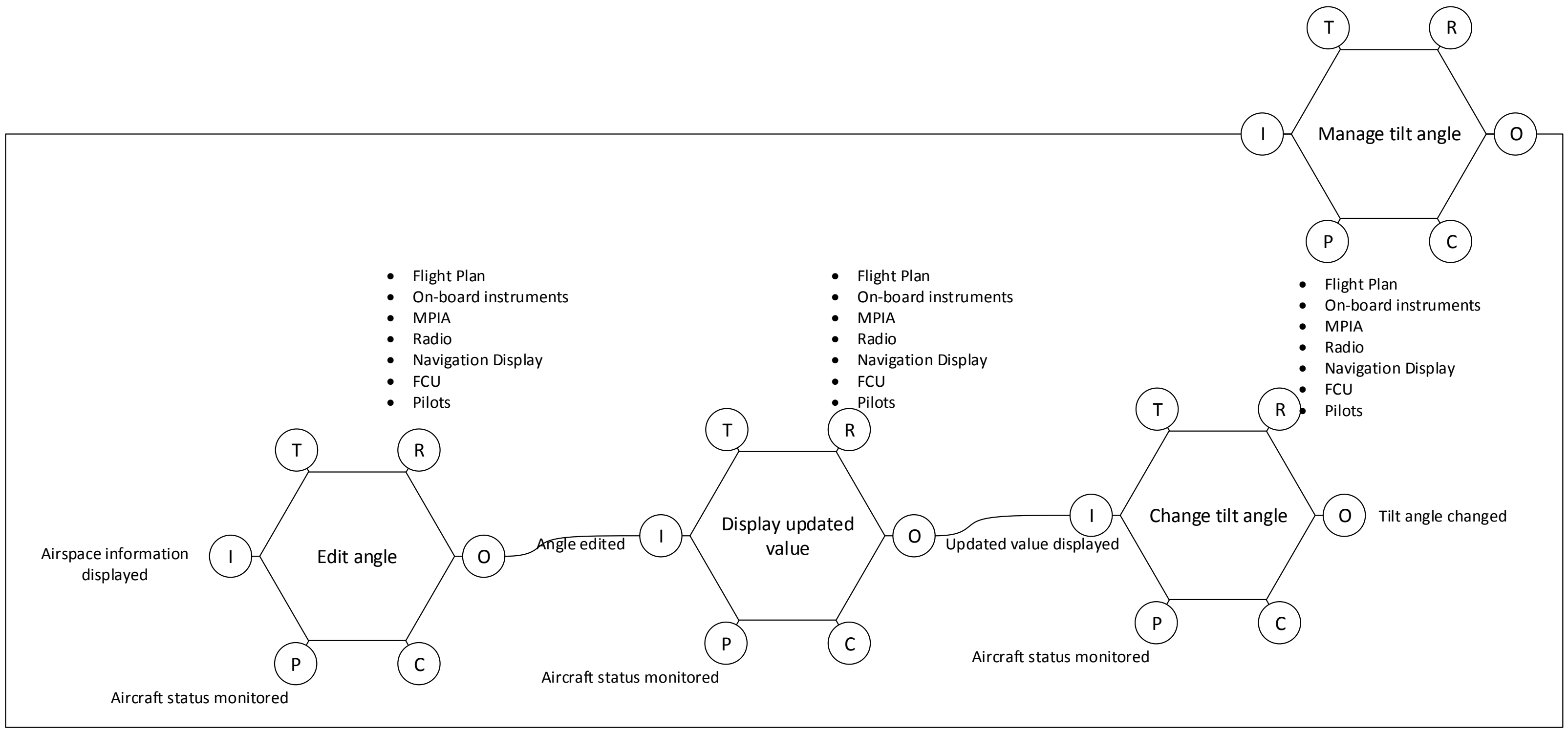


Figure 72: FRAM instantiation of the "Manage tilt angle" lower-function of “Manage weather radar status”

As described in Chapter 5 – Section 1.2.1, in parallel, for each identified type of function, the analyst will use dedicated modelling techniques for building the related models. According to Chapter 1 – Section 2.2, we selected HAMSTERS for modelling human functions. According to Chapter 2 – Section 2.2, we selected ICO&PetShop for modelling system functions. According to Chapter 2 – Section 5, we selected both modelling techniques for modelling interactive functions. This flow of the sub-phase outputs are “FRAM instantiations”, “HAMSTERS models”, and “ICO&PetShop models”.

For the WXR case study, Figure 73 illustrates the task model of the “Check weather conditions” main goal. According to the case study, one of the main goals of the pilot is “Check weather conditions” which includes several sub-routines such as “Manage weather status” that has to be performed iteratively (circular arrow symbol).

The task model of this sub-routine is illustrated in Figure 74 which represents the pilot activities performed in order to “Manage weather radar”. At the higher level of the tree, there is an abstract activity “Set up WXR” that is interrupted (operator [>) by a cognitive decision task “Decide WXR is ready”.

Connection between pilot’s activities and cockpit functions is made through interactive tasks (as input “Switch to WXON” and output “Check updated value”). The time required for performing the task heavily depends on the radar type. Such behavioural aspects of systems can be modelled using ICO notation and PetShop tool as detailed in Chapter 2 – Section 2.2. This task model corresponds to the manipulation of the user interface presented in Figure 52a). From these models we can see that the tasks to be performed in order to check weather conditions in a given direction are rather complex. The time required to perform them depends on 3 elements: the operator’s performance in terms of motor movements, perception and cognitive processing. Human performance models such as the one proposed in (Card, Moran, & Newell, 1986) can be used to assess difficulties and delays but the overall performance of the socio-technical system involves interaction and system execution times.

At the same time, tasks performed in a socio-technical system involve human interactions, such as the “Execute the clearance (change heading)” task. Figure 75 illustrates the task model of the “Execute the clearance (change heading)” task. At the higher level of the tree, there is an iterative activity “Manage change heading request” that is interrupted (operator [>) by a cognitive analysis task “Analyse that change heading clearance has been received” which can be allowed to the pilot to “Enter the new heading” abstract task. In the scenario under analysis in which the pilot can request a clearance at time, the iterative activity “Manage change heading request” has to be performed several times in order to find the appropriate clearance matching the pilot and the EXC needs.

However, as represented in the model, the pilot could make a choice (operator [] by “Request a clearance” abstract task): instead of following the communication/coordination protocol with the ATCOs requesting a clearance at time (“Decide about new heading” and “Ask for new heading clearance” tasks linked to the information “requested heading”), s/he can adopt a strategy (the StK “Communication/coordination with the ATCOs”). In this case, the pilot can “Decide about several possible headings” and “Ask for heading clearance with several options” providing a list of possible headings (information represented by a yellow box) to the EXC. This strategy could reduce the number of iterations between the pilot and the EXC. But this strategy is not the standard procedure and will be applied depending on the person who is in charge of the aircraft and on a common agreement with the remote EXC.

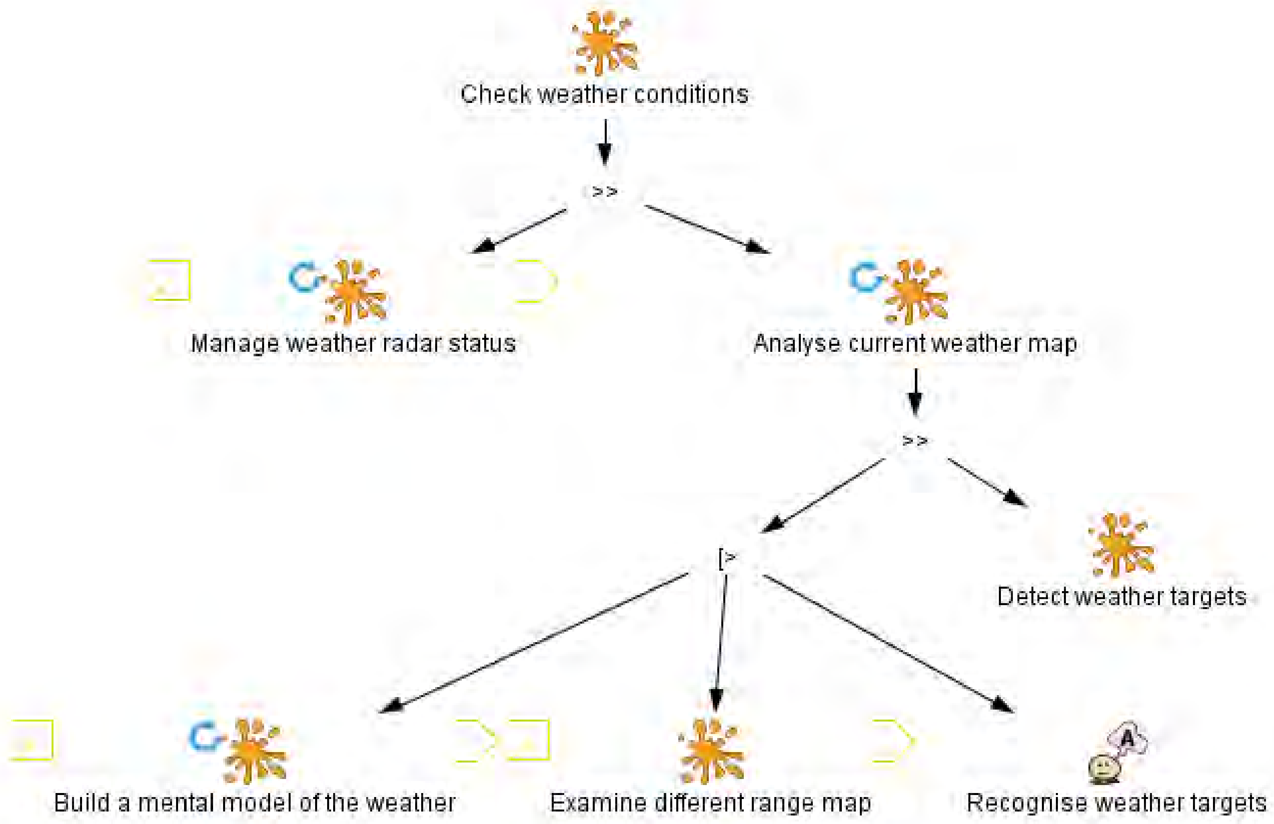


Figure 73: HAMSTERS 2.0 task model of the "Check weather conditions" main goal

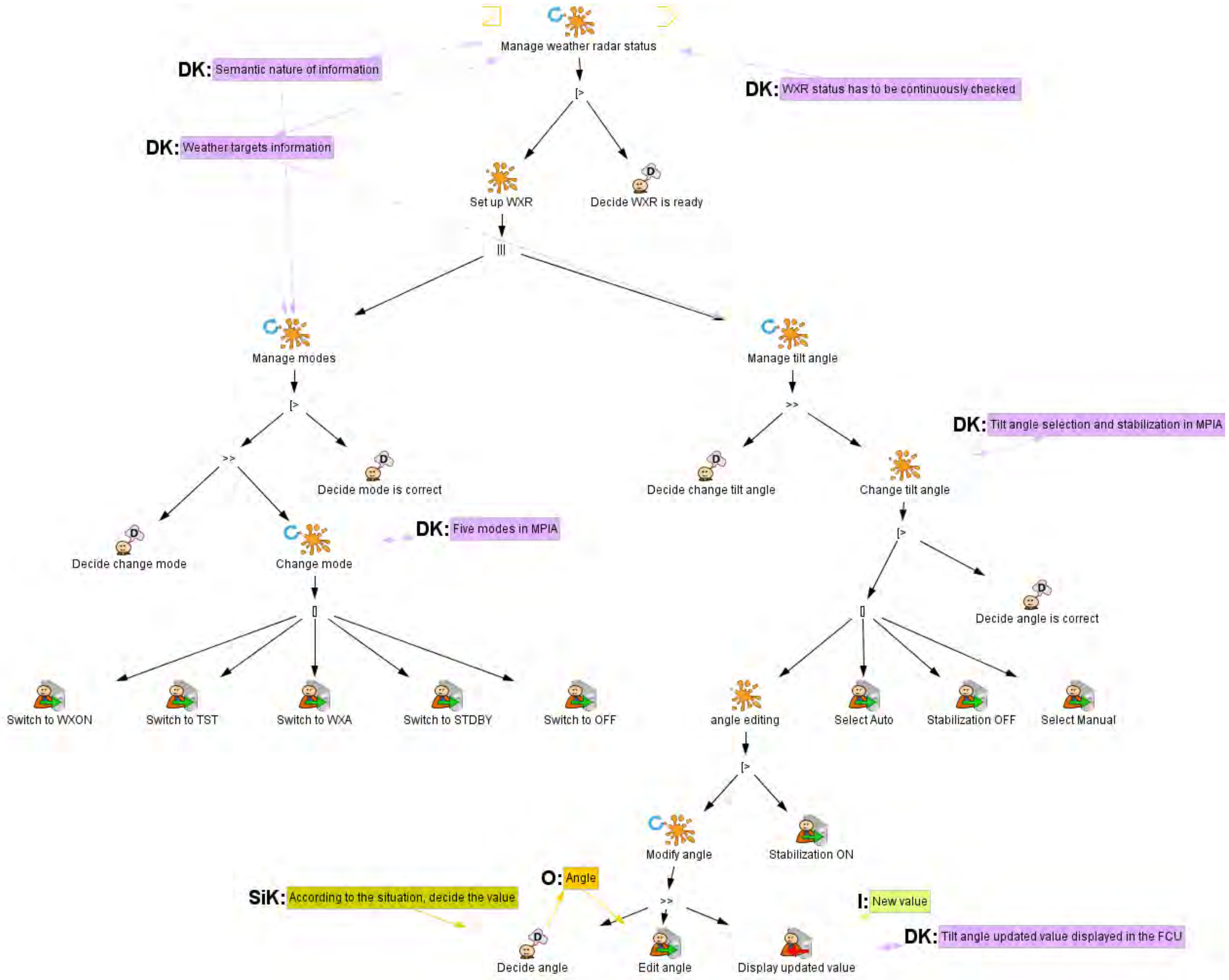


Figure 74: HAMSTERS 2.0 task model of the subroutine "Manage weather radar" task

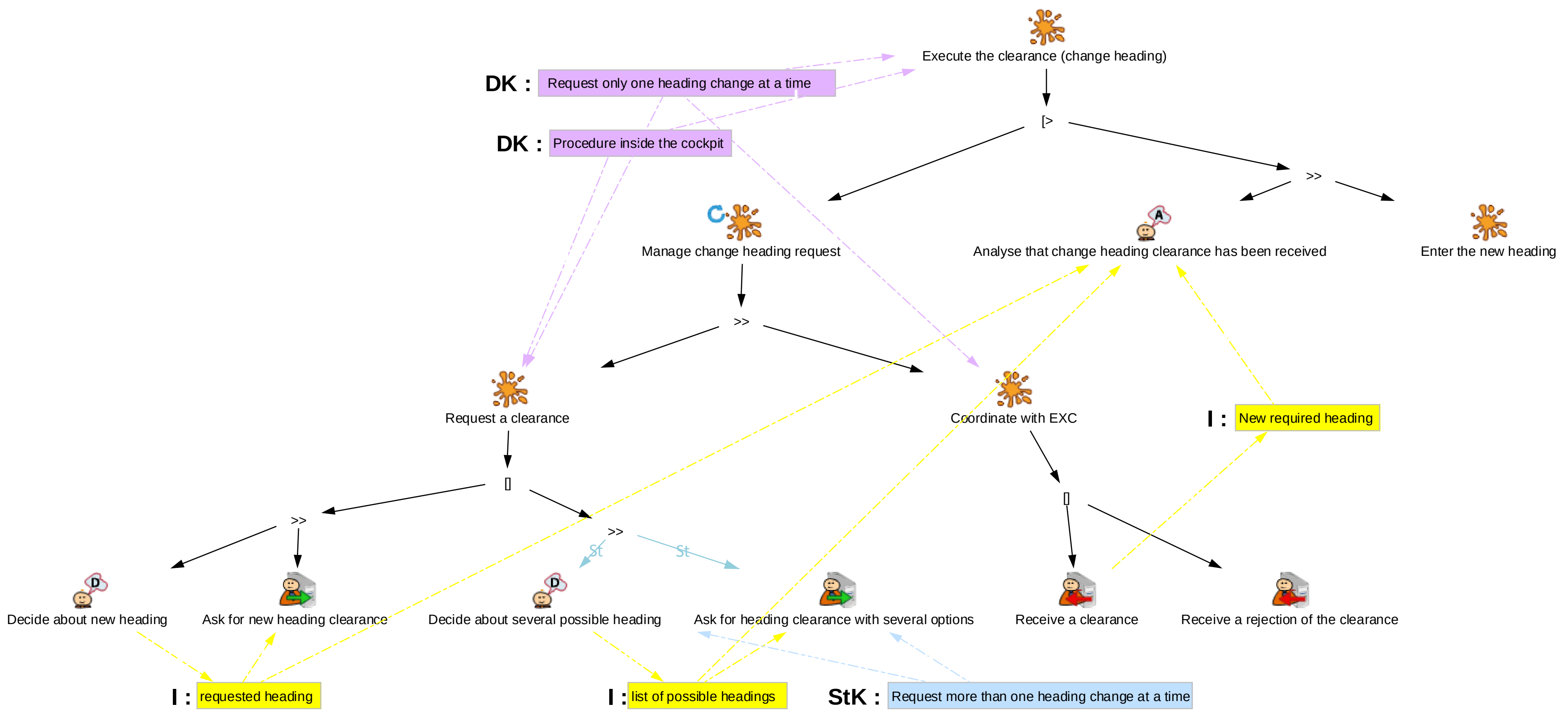


Figure 75: HAMSTERS 2.0 task model of the "Execute the clearance (change heading)" task

Regarding the system models, Figure 76 shows the ICO&PetShop model of mode selection and tilt angle setting WXR functionalities. As also represented through FRAM instantiation, the pilot has to continuously check the WXR status and select different range for detecting weather perturbations.

The ICO&PetShop model describes how it is possible to handle the weather radar configuration of both its mode and its tilt angle. Figure 76 shows the interactive means provided to the user to:

- Switch between the five available modes (upper part of the figure) using radio buttons (the five modes being WXON to activate the weather radar detection, OFF to switch it off, TST to trigger a hardware check-up, STDBY to switch it on for test only and WXA to focus detection on alerts).
- Select the tilt angle control mode (lower part of the figure) amongst three modes (fully automatic, manual with automatic stabilization and manual selection of the tilt angle).

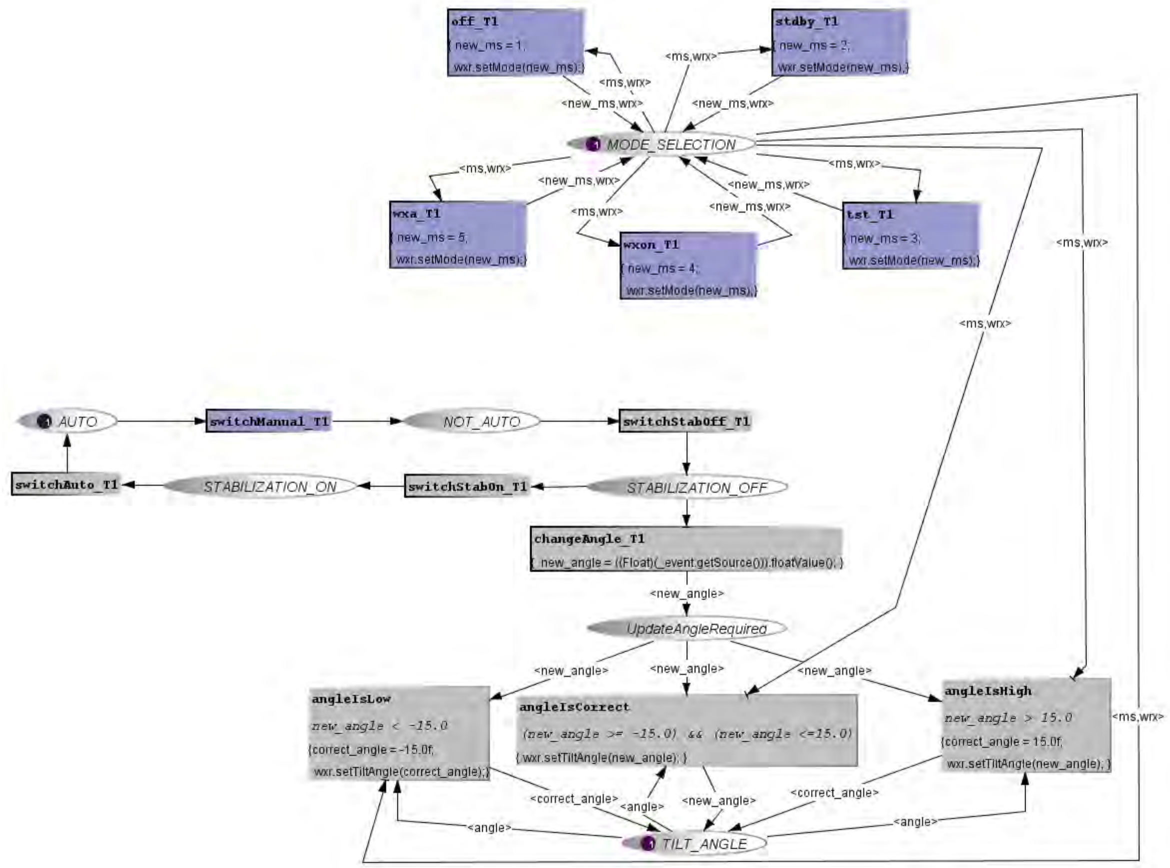


Figure 76: ICO&PetShop – Behaviour of the WRX mode selection and tilt angle setting

Figure 76 presents the description of the behaviour of this part of the interactive cockpit using the ICO formal description technique and may be divided into two parts.

- The Petri net in the upper part handles events received from the 5 radio buttons. The current selection (an integer value from 1 to 5) is carried by the token stored in MODE_SELECTION place and corresponds to one the possible radio buttons (OFF, STDBY, TST, WXON, WXA). The token is modified by the transitions (new_ms = 3 for instance) using variables on the incoming and outgoing arcs as formal parameters of the transitions. Each time the mode value is changed, the equipment part (represented by the variable wxr within the token) is set up accordingly.

- The Petri net in the lower part handles events from the four buttons and the text field (modify tilt angle). Interacting with these buttons changes the state of the application. In the current state, this part of the application is in the state fully automatic (a token is in AUTO place). To reach the state where the text field is available for the angle modification, it is necessary to bring the token to the place STABILIZATION_OFF by successively fire the two transitions switchManual_T1 and switchStabOff_T1 (by using the two buttons MANUAL and OFF represented by Figure 52), making transition change_Angle_T1 available. The selected angle must belong to the correct range (-15 to 15), controlled by the three transitions angleIsLow, angleIsCorrect and angleIsHigh. When checked, the wxr equipment tilt angle is modified, represented by the method called wxr.setTiltangle.

Figure 77 shows the ICO&PetShop model of range selection WXR functionality. As also represented through FRAM instantiation, the pilot has to select different range map for detecting weather perturbations.

The setting of the range detection of the weather radar is done using a FCU physical knob (please refer to Figure 52b) by switching between 6 values (from 1 to 6). Each time the value is set an event is raised (holding this value) by the knob and received by a dedicated part of the cockpit application. This part of the application is represented by the model of Figure 52b) Figure 77 that maps the value (form 1 to 6) into a range value that is sent to the WRX equipment.

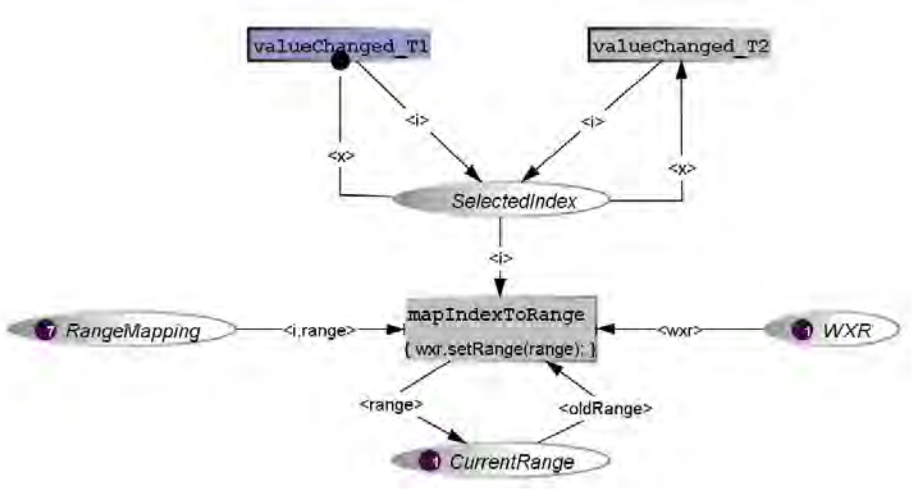


Figure 77: ICO&PetShop – Behaviour of the range selection

The event is received and the selected value is extracted by one of the two transitions called valueChanged_T1 and valueChanged_T2. The place RangeMapping contains the mapping between a value and the corresponding range (for instance 1 corresponds to range 10, 2 to 20…). Finally, the wxr equipment range is set with the selected range by the firing of transition mapIndexToRange.

In this IIa sub-phase, the other parallel flow is "Identify concepts", which will lead to the production of concept maps (for all sub-processes description please refer to Chapter 5 – Section 1.2.1). As demonstrated in PART II – Contributions - Chapter 4 , theoretically in Section 2 and practically in Section 3, concept maps provides a unified description of all the concepts (information, objects, knowledge and their relationships) and they can be used either to represent knowledge and/or to support automated systems for reasoning about the represented knowledge.

For the WXR case study, Figure 78 illustrates the concept map associated to HAMSTERS 2.0 task models for representing knowledge, information and objects (as explained theoretically in the first contribution of this thesis in Chapter 4 – Section 2) related to this case study.

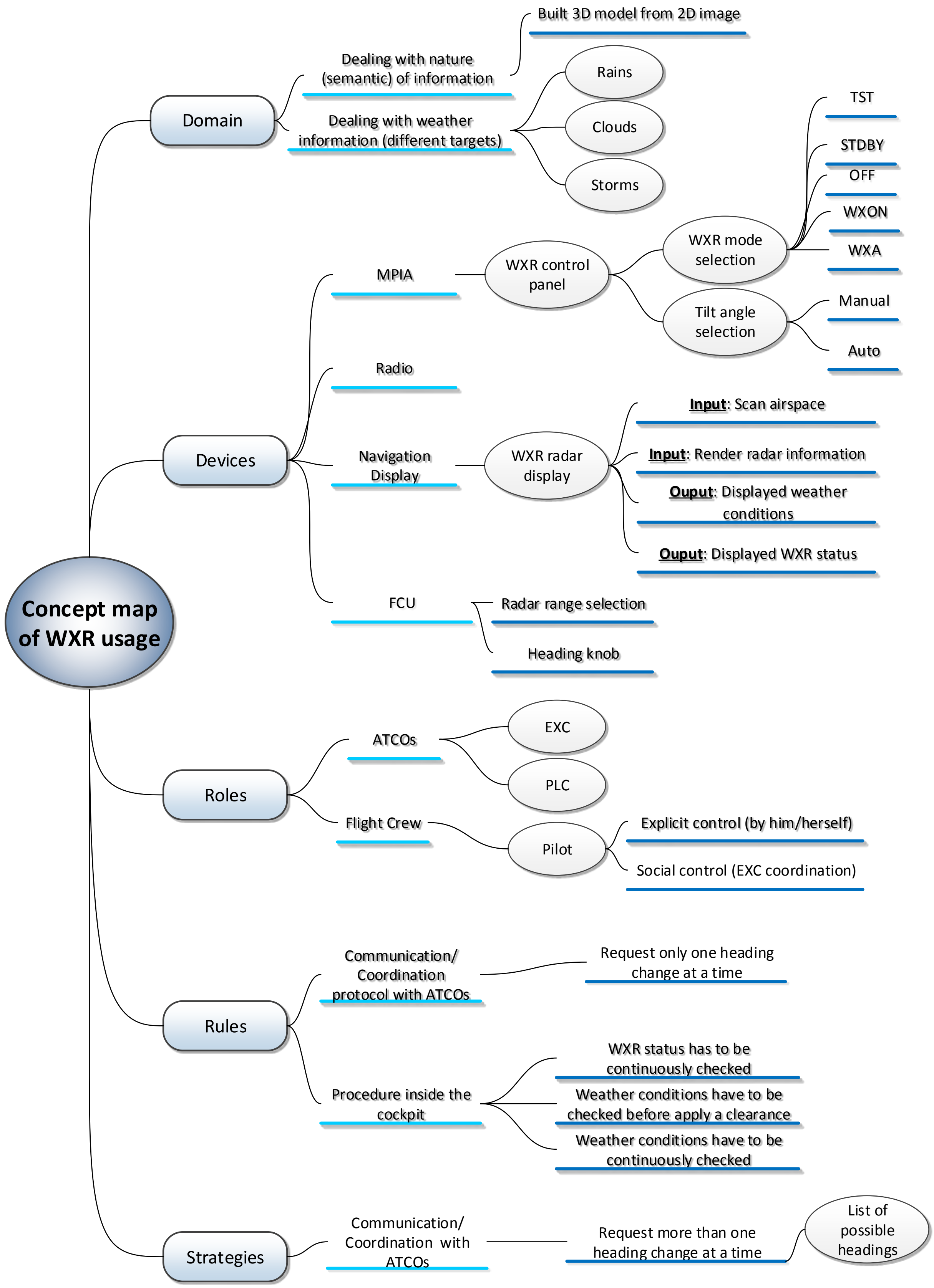


Figure 78: Concept map of WXR usage

"Figure 78: Concept map of WXR " represents required declarative knowledge necessary to be able to use the WXR. The concepts which are necessary or useful to perform actions during the flight phase are grouped and linked through a semantic network. It is made up of 5 key concepts or nodes using the recommended terminology explained in Section Chapter 42 of Chapter 4. These key concepts are Domain, Roles, Rules, Strategies and Devices. In this case study, these concepts are further detailed and applied to the specific context of the WXR. Structure from left to right shows the refinement of concepts, from abstract to concrete, as well as the instantiation. For example, the concept of "ATCOs" is instantiated twice: the "EXC" instance and the "PLC" instance. Lastly, relationships between refined concepts can also be represented in an explicit way, using links (as reported in the concept map of the Game of 15 in Figure 20). In order to illustrate the concept map in a seamless way and facilitate its reading, they are not shown here.

All the represented concepts should have a corresponding representation in one or several models produced with the task, system or organisation modelling techniques (as illustrated in the following picture).

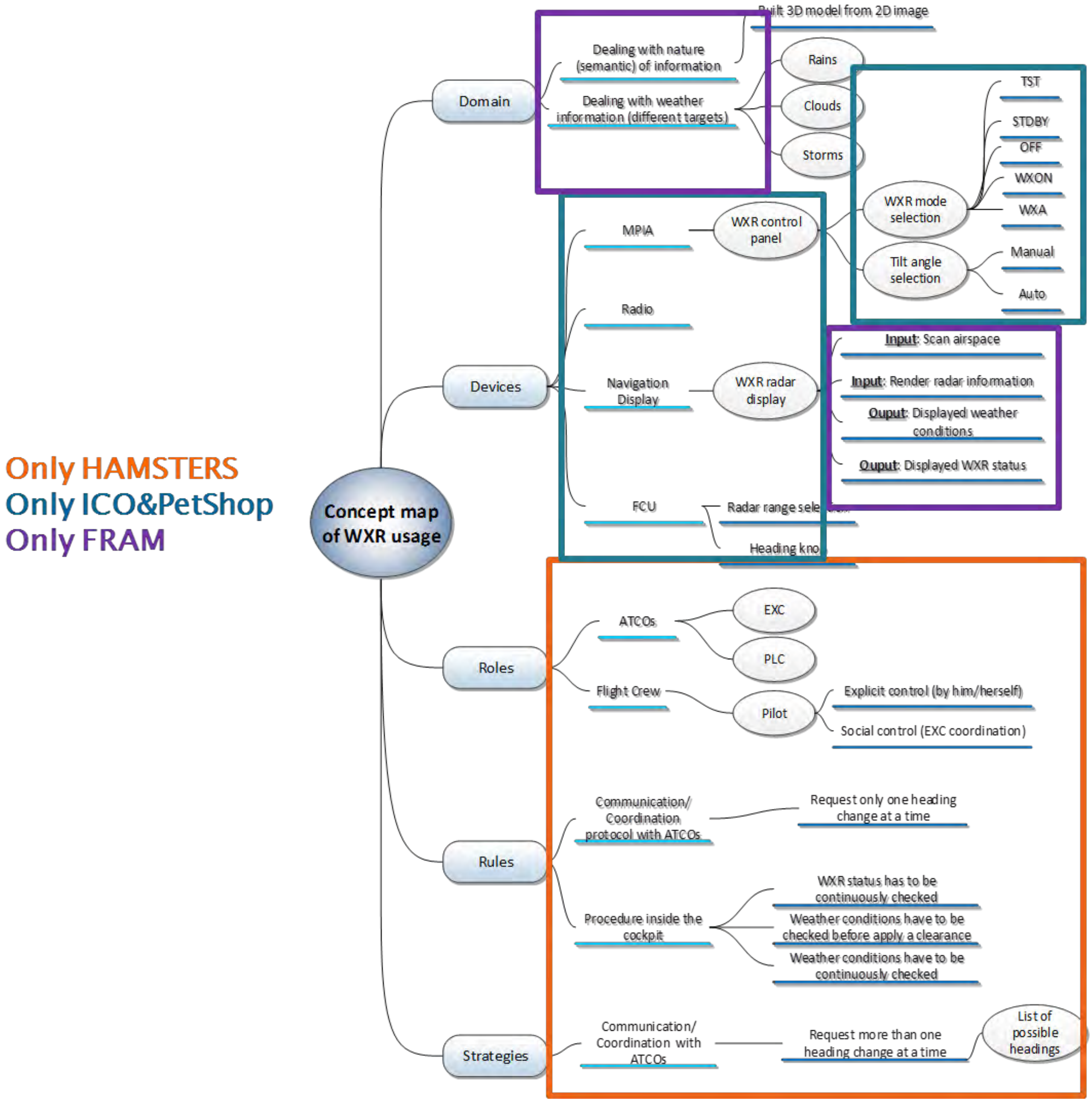


Figure 79: Concept map of WXR usage and concepts corresponding representation in one or several models

Moreover, to point out this correspondence between the WXR case study concept map and HAMSTERS 2.0, we can observe that the declarative knowledge violet box "Weather targets information" in "Figure 74: HAMSTERS 2.0 task model of the subroutine "Manage weather radar" task" corresponds to the bubble "Dealing with weather information (different targets)" in "Figure 78: Concept map of WXR ". These correspondences enable to ensure consistency between the different views needed to properly model and analyse a partly autonomous interactive system. The concept map can be integrated into the models, as presented in next section.

In order to proceed with the next sub-phase, the analyst has to check whether or not all the functions and all the concepts are represented in the models. If all of them are not represented ("No" back loop in Figure 24), the analyst has to re-perform some previous steps. Hence, s/he has to rework on identifying and describing functions and concepts (flow of steps starting by "Identify functions" and/or flow of steps starting by "Identify concepts" in Figure 24). Once all the functions and the concepts are represented in the models, s/he can carry on to the next sub-phase.

In this case, all the functions and the concepts are presented in the models. So, we can carry on the next sub-phase.

2.1.2 The Integration of the Models sub-phase applied to the WXR case study

As explained in Chapter 2 – Section 5, HAMSTERS and ICO&PetShop models can be integrated both at model and at tool level. This means that a variation of the system model that impacts a human task will trigger a change in the corresponding task model and vice versa. This helps in iteratively refining models of both types until all the corresponding models are consistent and coherent between each other. In addition, the integration of the system and task models with the FRAM instantiations, at the model level, provides a complementary view for assessing consistency and coherence between system and task models, as it enables to verify the sequence of function calls between models.

2.1.2.1 The correspondences assessment between models at tool level applied to the WXR case study

This section details the correspondences at tool level between HAMSTERS 2.0 models and ICO models regarding the case study of the WXR.

As reported in Table 87 and according to the corresponding models, the interactive input task "Change mode" (from the task model built through HAMSTERS 2.0 and shown in Figure 74) corresponds to an Event handler "ModeSelection" in the system model (which is built through ICO&PetShop and shown in Figure 76). Moreover, the principle of editing the correspondences between the two models is to put together system outputs Place Event "UpdateAngleRequired" in Place "AngleIsCorrect" (from the ICO&PetShop model) with Interactive output tasks "Display update value" (from the HAMSTERS 2.0 model).

TABLE 87: WXR CASE STUDY – HAMSTERS 2.0AND ICO&PETSHOP MODELS CORRESPONDENCES

Pilot		
Input correspondences table		
HAMSTERS	**ICO**	
Interactive input task	**Event handler**	
Switch to WXON	wxon_T1	
Switch to TST	tst_T1	
Switch to WXA	stdby_T1	
Switch to STDBY	off_T1	
Switch to OFF	wxa_T1	
Select Auto	switchAuto_T1	
Stabilization OFF	switchStabOff_T1	
Select Manual	switchManual_T1	
Edit angle	changeAngle_T1	
Stabilization ON	switchStabOn_T1	
Select range	mapIndexToRange	
Output correspondences table		
HAMSTERS	**ICO**	
Interactive output task	**Place**	**Place Event**
Display updated value	angleIsCorrect	UpdateAngleRequired

The synergistic use allows us in representing all the elements of a partly autonomous interactive system, and in this particular case, the user interface for the WXR (illustrated in Figure 52 of Chapter 6 – Section 1). Setting up this correspondence may show inconsistencies between the task and system model such as interactive tasks not supported by the system or rendering information not useful for the tasks performance.

This integration allows a real co-evolution of two models, as the execution of one model impacts the execution of the other related model. This integration can provide designers with shorter iterations in the task and system modelling process. It also represents an improvement for the end user as the execution of the system should support training and provide contextual help.

2.1.2.2 The correspondences assessment between models at model level applied to the WXR case study

This section describes the connections between FRAM and HAMSTERS 2.0 and ICO models applied to the WXR case study. As reported in Chapter 3 – Section 5, a support tool for FRAM is not available yet. However, this method can be extended and integrated with the others two modelling techniques at model level and it plays a fundamental role in assessing the models connections.

Indeed, FRAM (and its function instantiations) can be considered as a metamodel for connecting and representing the correspondences between HAMSTERS 2.0 and ICO models. It can be defined as a correspondence editor (please refer to Chapter 5 – Section 1.2.2.2).

The task models built through HAMSTERS 2.0 have their respective “Human functions” in the FRAM instantiations. This is also true for the system models built through PetShop which have the correspondent “System functions” in FRAM. In case of the interactive functions, at each variation on the task model, which impacts on system component, corresponds a change on the system model and vice versa. This means that in FRAM the output of the “Human function” (which corresponds to the HAMSTERS model) causes an input in the “Interactive function” generating a change in the correspondent system model (modelled through PetShop).

Figure 80 shows the role of FRAM as correspondence editor in the WXR case study. On the left side, an excerpt of the "HAMSTERS 2.0 task model of the subroutine "Manage weather radar" task" (please refer to Figure 74 for the whole representation). The task model has its correspondent function, named "Edit angle", in the FRAM instantiation.

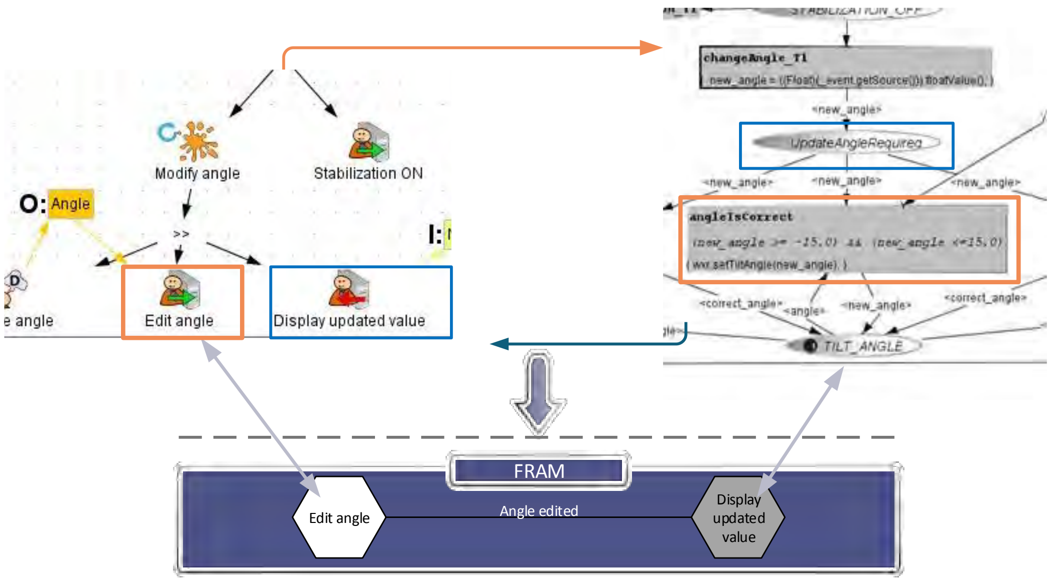


Figure 80: FRAM as correspondence editor in the WXR case study

For accomplishing this task, the pilot has to perform some consecutive tasks. One of these is to "Edit angle" (the orange box in the task model). This HAMSTERS 2.0 Interactive Input task has its correspondent in the PetShop model (the orange box in the system model). In FRAM, this correspondence is assessed through the output "Angle edited" of the function "Edit angle" which is the input for activating the "Display updated value" function. Consequently, this system function has its correspondent in the ICO model (the blue box in the system model) which has its correspondent in the HAMSTERS 2.0 model (the blue box in the task model).

In the WXR case study, all the correspondences have been assessed both at tool and also at model level. These assessments ensure that the models are consistent, coherent, and complete. Then, they have been integrated for properly representing the scenario and the related performance objectives. So, we can carry on to the Evaluation phase.

3 The Evaluation phase applied to the WXR case study

3.1.1 The Variability assessment sub-phase applied to the WXR case study

This phase starts with the step "Assess the variability" which consists of two complementary and parallel flows:

- "Perform the quantitative variability analysis" and
- "Perform the qualitative variability analysis".

The quantitative analysis flow provides an output which is the result of a refinement of the type of function variability. This can belong to human, to system and/or to interactive aspects. It results in a "Performance data of the scenario". To allow performance assessment we have to address timing issues at three levels: the operator side using the task models presented in Chapter 4 – Section 1, the system side exploiting the ICO behaviour models in Chapter 5 – Section 2.2, the interaction side related to the graphical interface described also in ICOs in Chapter 5 – Section 2.2.

With regard to the users' performance from task models, we first restricted the study to the interaction with the WXR interface (please refer to "Figure 52: Image of a) the weather radar control panel b) of the radar display manipulation" in Chapter 6 – Section 1). We have estimated the time it will take a user to point and click on a control of graphical interface. One of the evaluation approaches used in human factors domain is based on Fitts's law (Fitts, 1954) which is suitable for assessing motor movements. Fitts's law is presented in Formula (1) representing an index of difficulty for reaching a target (of a given size) from a given distance. Movement time (MT) for a user to access a target depends on width of the target (W) and the distance between the start point of the pointer and the centre of target (A).

$$MT = a + b\, log_2 \left(1 + \frac{2A}{W}\right) \quad (1)$$

For predicting movement time on the systems under consideration constants are set as follows: a=0 and b=100ms (mean value for users).

The following table presents the set of interactive widgets used within the weather radar control panel. For each widget, it provides a short name used for the following tables and the size used as the width for the Fitts's law (we use the minimum value between the width and the height to provide the assessment of the maximum difficulty to reach the considered widget).

TABLE 88: MIN WIDTH OF THE WIDGETS OF THE WXR USER INTERFACE

Interactive widgets	radio button off	radio button stdby	radio button tst	radio button wxon	radio button wxa	button Auto	button Manual	button ON	button OFF	text field angle
Short name	r1	r2	r3	r4	r5	b1	b2	b3	b4	t1
Min width (mm)	18	18	18	18	18	31	31	31	31	26

The following table provides the distances from the centre to each widget and between each widget. These distances are used to apply the Fitts's law when reaching a widget with a start point that can be the centre of the control panel or any widget.

TABLE 89: A) DISTANCE BETWEEN A WIDGET AND THE CONTROL PANEL CENTER IN THE WXR USER INTERFACE– B) TEMPORAL VALUES (IN MS) FOR USER INTERACTION USING FITTS'S

a)

Interactive widgets	r1	r2	r3	r4	r5	b1	b2	b3	b4	t1
Distance	104	130	115	100	87	77	17	128	104	132

b)

Interactive widgets	r1	r2	r3	r4	r5	b1	b2	b3	b4	t1
Temporal values (ms)	110	119	114	108	103	77	32	97	89	105

In addition to these motor values cognitive and perceptive values have to be used in order to cover all the elements of the task models. From (Card, Moran, & Newell, 1986) we know that the mean time for performing a comparison at the cognitive level is 100ms (ranging from 25ms to 170ms) while eye perception mean is 100ms too (ranging from 50ms to 200ms).

With regard to system models, in the ICO Petri net dialect, time is directly related to transition, which invokes services from the weather radar system (this is the case for transition off_T1 on Figure 76 which switches off the equipment). The duration of each invocation is presented in the following table (each value is coarse grain and depends on the type of weather radar). The 2000-4000ms value corresponds to the time required by the weather radar to scan the airspace in front of the aircraft (two or three scans are needed to get a reliable image).

TABLE 90: INTERACTION DELAYS INTRODUCED BY THE WXR APPLICATION

Model	Transition	Duration (ms)
WXR control panel model	Off_T1	500
	Stdby_T1	200
	Wxa_T1	500
	Wxon_T1	1000
	Tst_T1	1000
	angleIsLow	2000-4000
	angleIsCorrect	2000-4000
	angleIsHigh	2000-4000
Range selection model	mapIndexToRange	200

Using the HAMSTERS 2.0 task model in Figure 73 of Section 2.1.1 and the time estimations of the table above, we calculate that:

- the maximum time to configure the WXR interactive application for displaying weather conditions on the current route is 6688ms (change titl angle and change modes for ensuring that the WXR is properly work) while if s/he has to change 2 times the tilt angle the maximum time is 7376ms
- the minimum time to configure the WXR interactive application for displaying weather conditions on the current route is 4000ms (only test WXR appliacation),

In parallel to the quantitative analysis, we can perform the qualitative variability analysis. In particular, for the variability assessment of the usability performance objective (please refer to Chapter 6 – Section 1), an example of functional resonance can be between the two factors, time and precision, of the macro function "Check weather conditions". The results of the analysis are presented in a variability analysis table. This table synthesizes the downstream coupling between functions in order to identify functional resonance or dampening effects. Table 91 presents the variability analysis of the coupling factors of the function "Check weather conditions" and describes how they can have an impact on the output of the function, and consequently, on the input of the "Identify the adequate procedure" downstream function. Indeed, the output "Weather target recognised" of the "Check weather conditions" upstream function can have an impact on the input "Weather target recognised" of the "Identify the adequate procedure" downstream function with the three following different severity degrees:

- No impact (green box in the following table) which means that there is no impact on the output and on the downstream function,
- Medium impact (orange box in the following table) which means that the output presents a certain level of variability affecting the downstream function. However, this can be triggered by the output,
- High impact (red box in the following table) which means that the output presents a severe level of variability affecting the downstream function and this cannot be triggered.

TABLE 91: WXR CASE STUDY – TIME AND PRECISION COUPLING FACTORS OF "CHECK WEATHER CONDITIONS" MACRO FUNCTION

		Precision		
		Precise	Acceptable	Imprecise
Time	Too early	The pilot precisely recognises the weather target but too early. No impact on the output "Weather target recognised" and on the downstream function.	The pilot recognises the weather target in an acceptable manner but too early. No impact on the output "Weather target recognised" and on the downstream function.	The pilot partially recognises the weather target or doesn't pay appropriate attention to the recognition of the weather target. MEDIUM IMPACT (The adequate procedure could be applied because there is still time)
	On time	The pilot precisely recognises the weather target on time. No impact on the output "Weather target recognised" and on the downstream function.	The pilot recognises the weather target in an acceptable manner but on time. No impact on the output "Weather target recognised" and on the downstream function.	The pilot partially recognises the weather target or doesn't pay appropriate attention to the recognition of the weather target. MEDIUM IMPACT (The adequate procedure could be applied because s/he is on time).
	Too late	The pilot precisely recognises the weather target but too late. MEDIUM IMPACT (The adequate procedure could be still applied)	The pilot recognises the weather target in an acceptable manner but too late. HIGH IMPACT (The adequate procedure cannot be applied)	The pilot partially recognises the weather target or doesn't pay appropriate attention to the recognition of the weather target. HIGH IMPACT (The adequate procedure cannot be applied)

Moreover, these three different severity degrees can be represented in the FRAM instantiation as:

- No impact: the FRAM instantiation remains the same (as illustrated in "Figure 70: FRAM instantiation of the WXR case study")
- Medium impact: the affected function is marked with an orange wave representing this level of severity but the downstream function can be active (as illustrated in Figure 81),
- High impact: a red cross and a dotted line indicate that the output and the downstream function do not exist anymore (as illustrated in Figure 82).

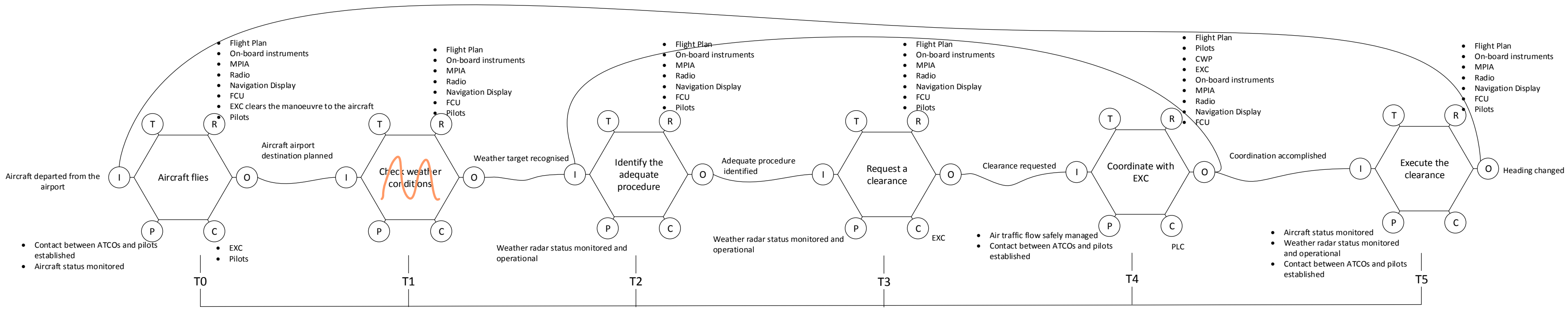


Figure 81: Medium impact of "Check weather conditions" macro function coupling factors – FRAM instantiation of the WXR case study

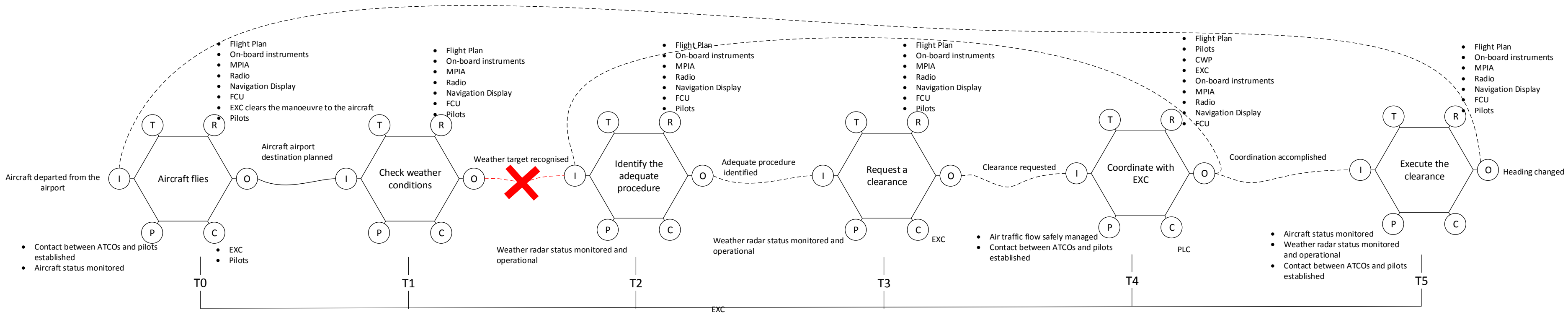


Figure 82: High impact of "Check weather conditions" macro function coupling factors – FRAM instantiation of the WXR case study

With regard to the variability assessment of the resilience performance objective (please refer to Chapter 6 – Section 1), an example of functional resonance can be between the two factors, time and precision, of the function "Coordinate with EXC". The results of the analysis are presented in a variability analysis table. This table synthesizes the downstream coupling between functions in order to identify functional resonance or dampening effects. Table 92 presents the variability analysis of the coupling factors of the function "Coordinate with EXC" and describes how they can have an impact on the output of the function, and consequently, on the input of the "Execute the clearance" and "Identify the adequate procedure" downstream functions. Indeed, the output "Coordination accomplished" of the "Coordinate with EXC" upstream function can have an impact on the input "Coordinate with EXC" of the "Execute the clearance" and of the "Identify the adequate procedure" downstream functions with the three following different severity degrees:

- No impact (green box in the following table) which means that there is no impact on the output and on the downstream function,
- Medium impact (orange box in the following table) which means that the output presents a certain level of variability affecting the downstream function. However, this can be triggered by the output,
- High impact (red box in the following table) which means that the output presents a severe level of variability affecting the downstream function and this cannot be triggered.

TABLE 92: WXR CASE STUDY – TIME AND PRECISION COUPLING FACTORS OF "COORDINATE WITH EXC" FUNCTION

		Precision		
		Precise	Acceptable	Imprecise
Time	Too early	The coordination between the EXC and the pilot is precisely accomplished and well in advance for cooperating in finding a possible clearance acknowledgment. No impact on the output "Coordination accomplished" and on the downstream functions.	The coordination between the EXC and the pilot is accomplished in an acceptable manner and well in advance for cooperating in finding a possible clearance acknowledgment. No impact on the output "Coordination accomplished" and on the downstream functions.	The coordination between the EXC and the pilot is imprecise, partially accomplished and too early. This allows to better understand the pilot's request. MEDIUM IMPACT (The "Execute the clearance" downstream function could be activated once the "Identify the adequate procedure" downstream function" has been re-performed)
	On time	The coordination between the EXC and the pilot is precisely accomplished on time. No impact on the output "Coordination accomplished" and on the downstream functions.	The coordination between the EXC and the pilot is accomplished in an acceptable manner on time. No impact on the output "Coordination accomplished" and on the downstream functions.	The coordination between the EXC and the pilot is imprecise, partially accomplished but on time. However, in order to better understand the pilot's request, it is needed additional time. HIGH IMPACT (The "Execute the clearance" downstream function cannot be activated and the "Identify the adequate procedure" downstream function" has to be re-performed)
	Too late	The coordination between the EXC and the pilot is precisely accomplished but too late for cooperating in finding a possible clearance acknowledgment. HIGH IMPACT (The "Execute the clearance" downstream function cannot be activated and the "Identify the adequate procedure" downstream function" has to be re-performed)	The coordination between the EXC and the pilot is accomplished in an acceptable manner but too late for cooperating in finding a possible clearance acknowledgment. HIGH IMPACT (The "Execute the clearance" downstream function cannot be activated and the "Identify the adequate procedure" downstream function" has to be re-performed)	The coordination between the EXC and the pilot is imprecise, partially accomplished and too late for cooperating in finding a possible clearance acknowledgment. HIGH IMPACT (The "Execute the clearance" downstream function cannot be activated and the "Identify the adequate procedure" downstream function" has to be re-performed)

Moreover, these three different severity degrees can be represented in the FRAM instantiation as:

- No impact: the FRAM instantiation remains the same (as illustrated in “Figure 70: FRAM instantiation of the WXR case study”)
- Medium impact: the affected function is marked with an orange wave representing this level of severity but the downstream function can be active (as illustrated in Figure 83),
- High impact: a red cross and a dotted line indicate that the output and the downstream function do not exist anymore (as illustrated in Figure 84).

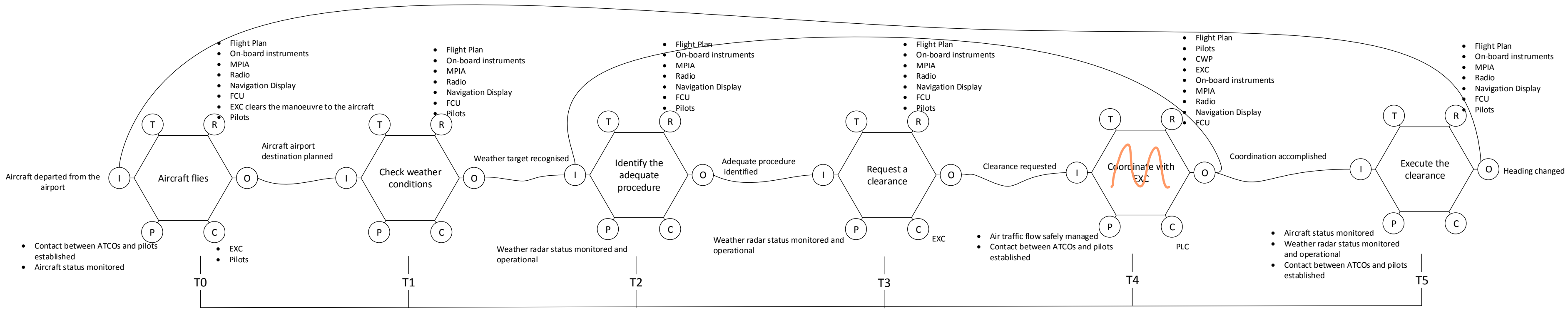


Figure 83: Medium impact of "Coordinate with EXC" function coupling factors – FRAM instantiation of the WXR case study

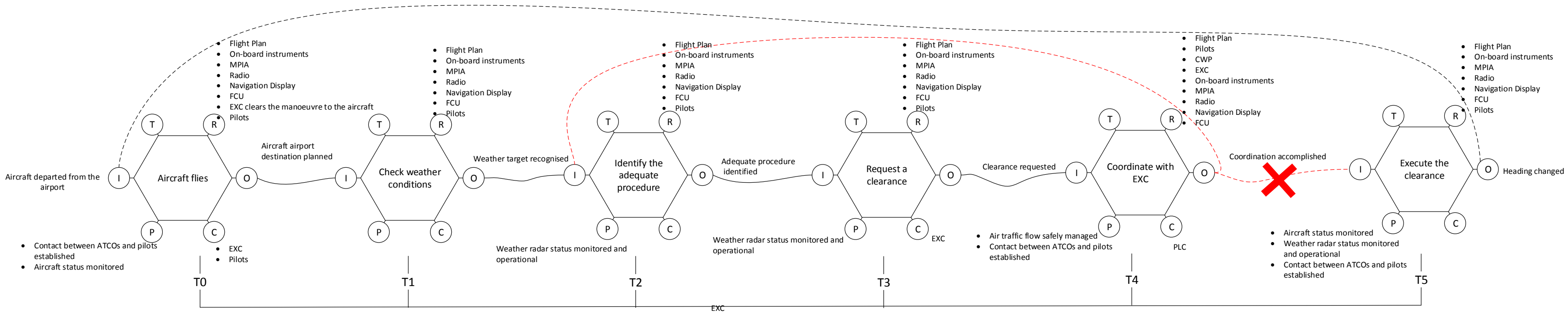


Figure 84: High impact of "Coordinate with EXC" function coupling factors – FRAM instantiation of the WXR case study

The variability analysis tables and the FRAM instantiations allow us to reason about the propagation of variability. With regard to the "Check weather conditions" function, it is a strong bottleneck and influences the entire STS under analysis because both the medium and also the high impact of its time variability, can strongly affect the downstream function and the entire scenario. With regard to the function "Coordinate with EXC", the provision of the clearance (and then its execution) could vary in terms of its timing aspect, i.e. it could be provided late (for instance according to the surrounding traffic or complex conflicts in the sector). In this case, there is less than adequate time available to execute the clearance. In such a situation, there may be a trade-off between efficiency and thoroughness in such a way that the weather check may be omitted (i.e. the precondition "Weather radar status monitored and operational") in order to save time. This would lead to an execution of the clearance, rather than to the more appropriate request for a new clearance. Furthermore, the functions "Request clearance" and "Coordinate with EXC" use a shared resource, i.e. on-board instruments such as the communication data link, which typically has limited bandwidth. A variation in the availability of this resource will again have implications for the timing of the functions. So, the variability assessment based on the variability analysis tables and the FRAM instantiations suggests that variability due to timing and resource aspects may lead to potentially hazardous situations.

In the WXR case study, both the qualitative and quantitative assessments of performance variability are detailed enough. So, we can carry on the next sub-phase.

3.1.2 The Coverage of the performance objectives sub-phase applied to the WXR case study

Once the variability assessment has been detailed enough, it is possible to start the "Coverage of the performance objectives" sub-phase (please refer to Chapter 5 – Section 1.3.2). The first step is to "Analyse the gathered data". Indeed, it is needed to analyse both qualitative and quantitative data. These data can be consolidated and verified through a comparison between them. This should ensure that all the performance objectives are covered.

For the WXR case study, the performance objectives defined in Section 1, have been covered both through the quantitative variability analysis and also through the qualitative one in Section 3.1.1. In particular,

- with regard to usability objective:
  - it has been fulfilled because the maximum time to configure the WXR interactive application for displaying weather conditions on the current route shall not exceed 10 seconds
- with regard to resilience objective:
  - it has not been fulfilled because the pilot could not be able to find an appropriate new route and to get approval to implement this new route whatever the surrounding traffic and weather conditions as reported in "Table 92: WXR case study – Time and Precision coupling factors of "Coordinate with EXC" function" and illustrated in "Figure 83: Medium impact of "Coordinate with EXC" function coupling factors – FRAM instantiation of the WXR case study" and "Figure 84: High impact of "Coordinate with EXC" function coupling factors – FRAM instantiation of the WXR case study". Indeed, in both FRAM variability analysis table and instantiations, we can observe that the clearance execution is strongly affected by the functional resonance of the "Coordinate with EXC" upstream function.

Now we have to understand why the resilience objective has not been fulfilled and, find a possible solution for achieving it (e.g. through a re-design or through further investigations). So, we can carry on the Analysis phase.

4 The Analysis phase applied to the WXR case study

4.1.1 The Identification of re-design opportunities sub-phase applied to the WXR case study

In the case of WXR case study, according to the results of the evaluation, the resilience objective is not achieved because the pilot could not be able to find an appropriate new route and to get approval to implement this new route whatever the surrounding traffic and weather conditions as reported in Table 92. Indeed, in this table the qualitative variability analysis between the coupling factors (Time and Precision) of the "Coordinate with EXC" function have shown that its impact, medium and high, can affect the downstream function and the entire system under analysis in term of clearance execution. Then, we can identify an opportunity for re-designing the system under analysis such as changing, on the human side, a task for allowing the pilot in requesting more than one clearance at a time to avoid delays in coordination exchanges. Thanks to the fact that we take into account the partly autonomous interactive system within its STS as a whole, we are able to indifferently re-design human or system aspects for improving the resilience of the whole STS. So, if in the illustrative example (of previous Chapter 5) we have improved it through the redesign of the interactive application, in this case, we here decide to improve it through the redesign of the human activities. Then, Figure 85 illustrates the revised HAMSTERS 2.0 model of the "Execute the clearance (change heading)" task. Differently from the previous task model (please refer to Figure 75), the strategic knowledge is integrated into the declarative knowledge "Request more than one heading change at a time" (highlighted by the orange box at the top of the figure). We choose to integrate this task as a regular procedure.

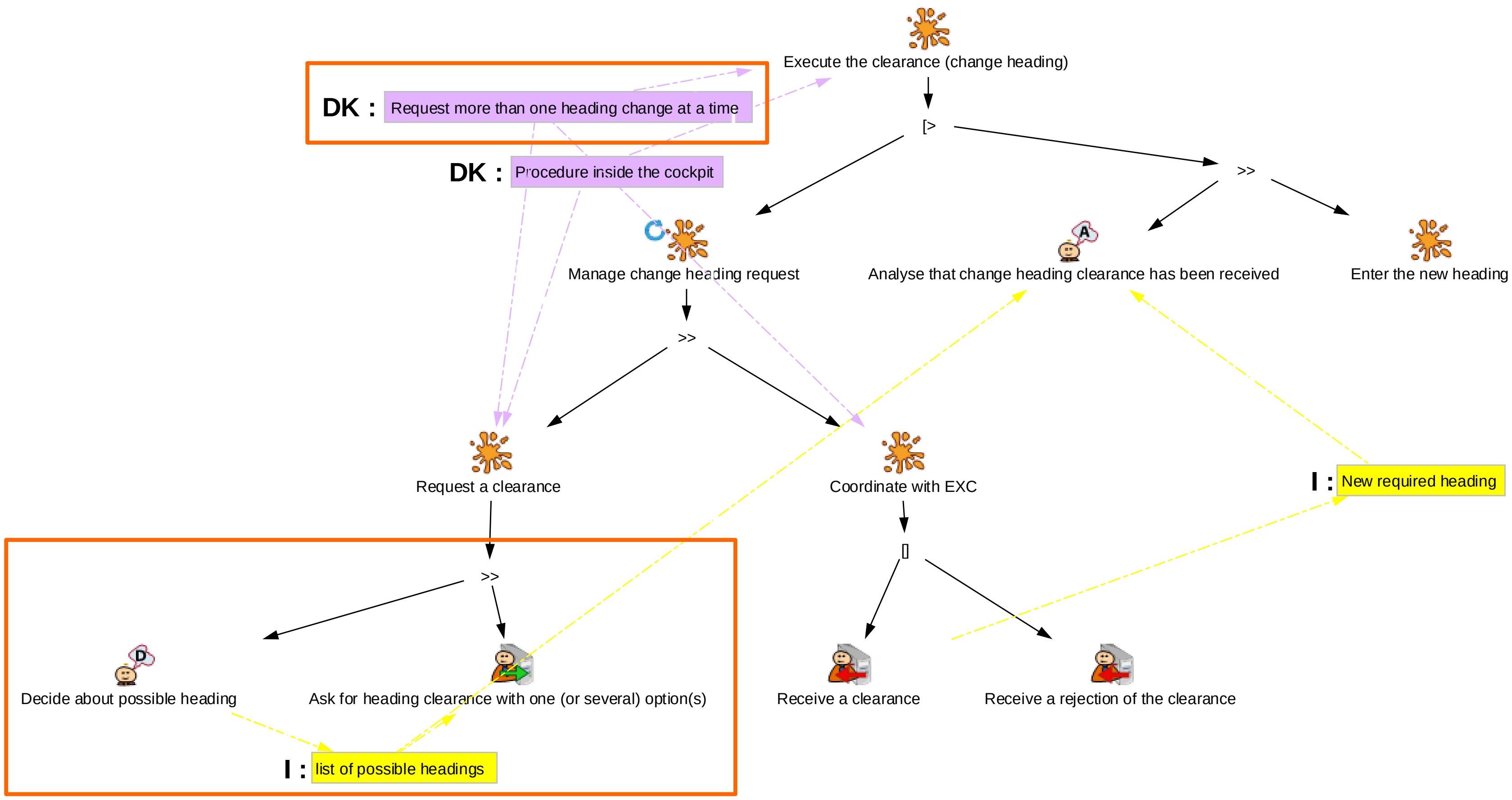


Figure 85: Revised HAMSTERS 2.0 task model of the "Execute the clearance (change heading)" task

This means that the pilot is allowed to provide at the same time a "list of possible headings" (Information yellow box in the model) to the ATCOs (highlighted by the orange box at the bottom of the figure). So, the EXC can evaluate simultaneously all of them and choice the one which best meets her/his needs. This reduces the coordination exchanges times.

Due to the fact that all the models have to be coherent, consistent and complete between them, the current change in the HAMSTERS 2.0 model, which involves the knowledge, should be reflected in the concept map. Differently from the previous concept map (please refer to Figure 78) and according to the revised HAMSTERS 2.0 model, Figure 86 shows the revised concept map of WXR usage in which there is no Strategy but this has been implemented in the Rules as "Communication/coordination protocol with ATCOs" (highlighted by the orange box at the bottom of the following figure).

This improvement on the human side could help in:

- improving the resilience of the PAIS within its STS by reducing the medium and high impact of the coupling factors for the "Coordinate with EXC" function presented in the Table 92 of the FRAM model and illustrated in the instantiations (Figure 83 and Figure 84 respectively)

Once the re-design opportunity has been identified, it is needed to come back to the modelling phase in order to integrate all the models and to re- perform all the required steps for evaluating, qualitatively and quantitatively, the whole variability of the revised partly autonomous interactive system (unfortunately due to space constraints, we cannot reperfom all the required steps). Furthermore, as argued in Chapter 2 – Section 5 and also demonstrated in the illustrative example and in this section, the developed multi-models approach can support automation design, as it enables the comparison between several design opportunities to allocate tasks and functions between users and systems.

So we can carry on to the next sub-phase.

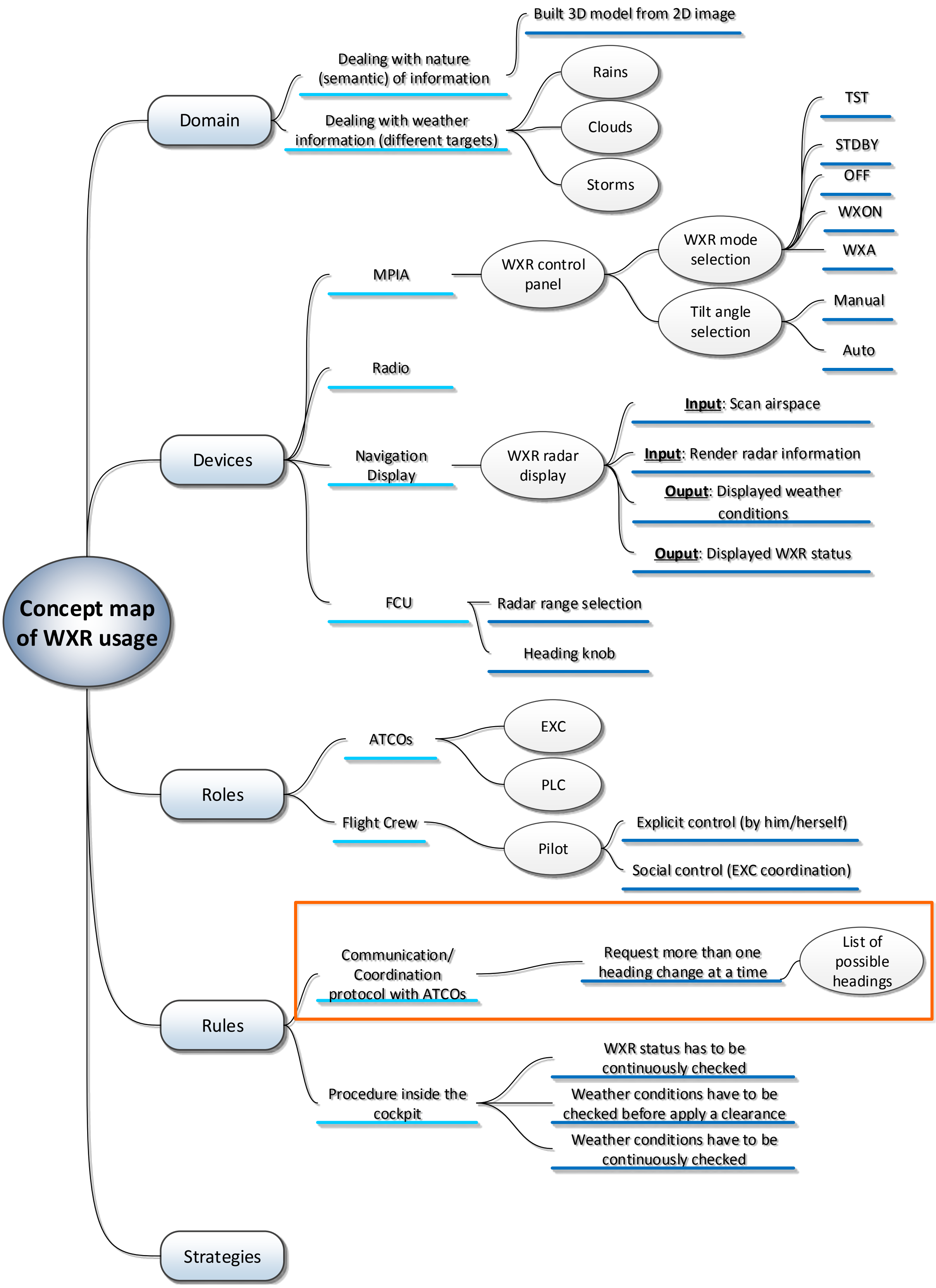


Figure 86: Revised concept map of WXR usage

4.1.2 The Identification of further investigations opportunities sub-phase applied to the WXR case study

Once that it has been ensured that it is not needed to re-design the PAIS, we can start with this sub-phase in order to focus on the scenarios and performance objectives and then, if it is required, to "Identify opportunities for changing scenarios/for creating new scenarios/for modifying objectives".

The analyst can identify opportunities for:

- changing scenarios,
- creating new scenarios, and
- modifying performances objectives.

If additional investigations are required, the analyst has to re-perform some previous steps. These can belong to the previous phase ("Yes" back loop to the "Variability assessment" of the "Evaluation phase" III labelled III in Figure 33) if the analyst needs to assess another level(s) of variability. The analyst may also need to change and/or define new scenarios and/or to modify the performance objectives ("Yes" back loop to the "Identify scenarios/performance objectives" of the "Definition phase" I labelled I in Figure 33). If further investigations are not required, the process can end.

For what concern the WXR case study, we decide that further investigations are not required. So the application of the process ends here.

5 Conclusion

This application of the proposed approach to the WXR case study demonstrates the scalability of the approach and highlights the ability of the integration of the models to cope with performance variability of the various parts of partly autonomous interactive system within its associated Socio-Technical System. So, the multi-models based approach provides a structured process for modelling and analysing a partly autonomous interactive system within its associated STS and provides support for taking into account usability and resilience properties. Then, the contribution is twofold by providing support for both modelling and analysis activities.

On the modelling side, this approach takes the best of each single modelling technique and through their integration into a multi-modelS based approach offers a holistic view on a partly autonomous interactive system within its associated STS. This approach consists in integrating the FRAM method with HAMSTERS 2.0 and ICO modelling techniques and tools. As demonstrated through the WXR case study:

- FRAM provides different levels of granularity for exploring the possible dependencies and situations of functional resonance within the partly autonomous interactive system under consideration. It can be considered as the right glue to fix this integration and to ensure consistency, coherence and completeness between the other two modelling techniques acting as a correspondence editor (as explained in detail in Chapter 5 – Section 1.2.2.2 and demonstrated in Section 2.1.2.2);
- The dependencies between the different elements of a STS have been further explored using HAMSTERS for human activities and ICO-PetShop for technical systems covering both interaction techniques on the user interfaces and the underlying hardware and software systems.

The integration of FRAM, HAMSTERS 2.0 and ICO leverages the high-level view on partly autonomous interactive systems, provided by FRAM, with the fine-grain view on human-system interaction, provided by HAMSTERS 2.0 and ICO. Consequently, the approach provides a framework allowing the modelling of partly autonomous interactive system performance variability under different conditions, with different levels of granularity.

On the analysis side, the main contribution is to associate qualitative performance variability analysis of the FRAM method with quantitative user and system performances evaluation support from HAMSTERS 2.0 and ICO (as explained in Chapter 5 – Section 1.3 and demonstrated in Section 3). This integration ensures consistency, coherence and completeness of the data collected during the modelling phase. Thanks to these qualitative and quantitative analyses, we can individuate areas for improving resilience and usability (as demonstrated through the WXR case study in Section 4). Consequently, the developed multi-models based approach can support automation design and task migration and/or functions allocation. Indeed, both require that humans improve their knowledge, learn how to interact and collaborate, between them and with the new technology, for successfully accomplishing tasks.

Conclusions

The research presented in this thesis concerns the increase of automation in Socio-Technical Systems (STS). It is clear, from interactions we experience on a daily basis that the continuous increase of automation (e.g. new high-tech smartphones that are getting smarter than the users) can lead to frustration and unintentional mistakes when interacting with devices. Such erroneous interactions in the context of STS can have a negative impact on the behaviour of the whole system and potentially jeopardise the resilience of the system.

When adding a new technology or an enhanced version of an existing technology in a STS, such as aircraft glass cockpits or Air Traffic Control stripless workstations equipped with Arrival MANager (AMAN) or Departure MANager (DMAN), the unpredictable nature of humans can bring human unreliability issues in addition to technical unreliability. On the industrial side, even though classical development processes are employed for the development of such systems, they are not able to guarantee both usability and resilience of the produced partly autonomous interactive systems. On the research side, reduced views are used for the development of partly autonomous interactive systems within a STS which usually focus on some specific aspects and hide away other ones due to STS complexity. These limited views allow providing local solutions to the development issues but their integration is usually claimed to be beyond the scope of the research.

The contribution of this thesis for addressing all the issues introduced above, it is a multi-models based approach for the modelling and the analysis of partly-autonomous interactive systems within a STS for assessing their resilience and usability.

The design driver of the contribution was twofold:

- Address all the aspects of the STS in a single integrated framework
- Address the entire development process of STS from design to evaluation with a special emphasis on automation dependability as this is key in Lisanne Bainbridge (Bainbridge, 1983) view on ironies of automation.

Addressing all the aspects of the STS is unachievable without tackling it at the right level of abstraction. Indeed, it would be very easy to get lost too early in low-level details or to miss important aspects by focussing too much on a specific part. For this reason one of the main hypotheses of this thesis is that using an adequate model allows to describe the STS at the adequate level both for design and analysis purposes. The other hypothesis is that due to the very different nature of the three parts of the STSs one model does not fit all but different models have to be used and, of course, that these various models have to be integrated to describe the entire relevant elements of the STS. While the thesis focuses on Air Traffic Management (ATM) systems, the definition of the models and the process for using them has been separated from the ATM case studies. This has demonstrated that the approach is not dedicated to ATM but is also suitable to other large scale STSs (even though some adjustments might be required to manage their idiosyncrasies).

Thus, the contribution of this thesis is first based on the identification of a set of requirements needed being able to model and analyse each of the STS elements. Some of these requirements were met by existing modelling techniques, others were reachable by extending and refining existing ones. This thesis proposes an approach which integrates three modelling techniques: FRAM (focused on organisational functions), HAMSTERS (centred on operator goals and activities) and ICO (dedicated to the modelling of interactive systems and supporting the development of dependable computing systems). The integration of models has been done at a detailed level through a dedicated process defining in a stepwise manner how to go from one model to another one but also how information represented in one model is reused and possibly refined in another one.

The principle of the multi-models approach has been illustrated on an example for carefully showing the extensions proposed to the selected modelling techniques and how they integrate together. A more complex case study from the ATM domain has been presented to demonstrate the scalability of the approach. This case study, dealing with aircraft route change due to bad weather conditions, highlights the ability of the integration of models to cope with performance variability of the various parts of the STS.

Chapter 1, Chapter 2, and Chapter 3 provided an overview on the modelling techniques and their current limitations in dealing with respectively human, system and organisation representation through models. Moreover, these chapters offered the opportunity to identify the most appropriate modelling techniques for modelling each element of the STS and to understand the requirements needed to improve and integrate them. In particular in Chapter 1, the description of existing task modelling techniques and methods has demonstrated the limited support for explicitly representing the various types of information, objects manipulated by the users and different types of knowledge required to interact with a partly autonomous interactive system. The highlighted limitations are a justified need for improving task modelling techniques and methods for which many other considerations much be taken into account to ensure efficient and successful human computer interaction particularly for partly autonomous interactive systems. In order to overcome these limitations, a set of requirements has been proposed. The identified requirements can guarantee a support for representing in a complete and systematic way information, objects and knowledge users need when accomplishing tasks while using partly autonomous interactive systems.

These aspects are detailed in Chapter 4 which proposed one of the contributions of this thesis; the needed extensions for representing in a complete and systematic way information, objects and knowledge (through the integration of concept maps) in the selecting task modelling technique, HAMSTERS. Moreover, all these improvements have been implemented in the tool and are show in an illustrative example. These HAMSTERS extensions used in a synergistic way with concept maps can provide useful support in the various phases of the development process of an interactive system (provided that it explicitly refers to a task analysis and task modelling phases such as the one presented in (Martinie, Palanque, Navarre, & Barboni, 2012)):

- Requirements elicitation by identifying what the users' needs in terms of knowledge, information and objects
- Design of the interactive system by making explicit the information that has to be provided to the user and the type of input that the users will need to perform
- Implementation of the interactive system by providing an informal but complete description of the data that the system will have to embed in order to allow performance of the user tasks
- Evaluation of the interactive system by providing an explicit representation of the information that will be manipulated and thus identifying candidate values for user testing.
- Training of the future operators of the interactive system by making explicit what knowledge they have to master in order to use the system.

Chapter 5 described in detail the developed multi-models based approach in which the selecting modelling techniques have been integrated and their links carefully explained. The approach provides a structured process for modelling and analysing a partly autonomous interactive system within its associated STS and provides support for taking into account usability and resilience properties.

Chapter 6 demonstrated the scalability of the approach and highlighted the ability of the integration of the models to cope with performance variability of the various parts of partly autonomous interactive system within its associated Socio-Technical System.

The proposed multi-modelS based approach for the modelling and the analysis of usable and resilient partly autonomous interactive systems address:

- all the aspects of the STS in a single integrated framework
- Address the entire development process of STS from design to evaluation.

The proposed approach has been partially applied to a WP-E project named “System Performances under Automation Degradation” (SPAD). The main goals of SPAD were a) to understand, model and estimate the propagation of automation degradation in ATM, and to evaluate the associated consequences on ATM performances, b) to validate the above results on a large ATM system with high degree of automation. To achieve these goals, part of the approach presented in the thesis (with the two modelling techniques, HAMSTERS and FRAM) has been applied to the “Arrival MANager” and “Remotely Piloted Aircraft Systems” case studies through different scenarios of automation degradation (with increasing level of severity). The application of the proposed approach to these scenarios helped in demonstrating the scalability of the approach and in demonstrating the suitability of the integration of the models to cope with higher level of automation and with performance variability of the ATM System.

However, there are some limitations that should be taken into account and addressed in the future. At the moment, the analysis remains human driven, which means, under the control and responsibility of the analyst and which requires a significant contribution from the analyst and the operational experts with a significant degree of experience. Finally, the multi-modelS based approach has been tested onlywithin the ATM domain and has not been tested within other different application domains.

Perspectives

Short-term perspectives can be derived from the limitations of modelling techniques (in particular for what concern HAMSTERS and FRAM as outlined in the conclusion of Chapter 1 and Chapter 3 respectively), but also from the results of the implementation of the developed approach on case studies.

With regard HAMSTERS, in addition to the extensions already presented in this thesis as first contribution (please refer to Chapter 4), it would be useful to model and analyse human errors and their impact on the usability and on the resilience of the partly autonomous interactive system within its associated STS. Indeed, human reliability is a relevant topic in the context of safety critical systems as highlighted in do Nascimento Neto et al. (do Nascimento Neto, Vieira, Santoni, & Scherer, 2009) and in Santoni et al. (Santoni, Vieira, Scaico, Vieira, & Pereira, 2004).

Furthermore, in order to better integrate concepts maps and representation of data within HAMSTERS, the HAMSTERS editor could be enhanced in order to embed a concept map editor whose data elements would be linked to data elements in task models.

With regard to FRAM, it would be useful to design and develop tools for supporting FRAM models edition and simulation as currently offered by PetShop and HAMSTERS. This is of critical importance if variability has to be assessed in a systematic way as due to propagation a lot of functions can be impacted by the evolution of a single source of variability in one function. Even if the purpose of FRAM is to represent the dynamics of the Socio-Technical System rather than to quantify the variability, in this thesis thanks to some improvements and FRAM integration in a multi models based approach, we have tried to overcome this issue by adding notation elements for characterising the interactive functions and for extending endogenous factors expressive power in order to highlight the explicit link between task, interactive system and organisational models However, we noticed that the measurement units proposed by the FRAM method for analysing the variability of endogenous and exogenous factors was often imprecise. Then, another perspective would be to refine the FRAM method so that it supports more precise description of the potential variability of endogenous and exogenous factors.

In addition, a minor contribution of this PhD thesis, has been to launch, encourage and supervise the design and development of a FRAM editor (still under development by the ICS team at IRIT Lab of Toulouse). Future work should aim at integrating this FRAM editor with the HAMSTERS and Petshop tools. The resulting integrated tool suit will aim at supporting the analyst during the modelling phase, and will aim at allowing the integration at tool level of FRAM models with other HAMSTERS task models and ICO interactive system models. This fully integrated tool suit would be a starting point for automating some steps of the quantitative variability analysis. In that case, the possibility of formalizing the interaction between the models at the different levels of analysis should be explored further. Currently, HAMSTERS and ICO models have precise connection points but FRAM and HAMSTERS models and FRAM and ICO models may be connected at several levels of abstraction, depending on the skill of the analyst. This may be necessary to for the analysis of whole STS encompassing several interactive systems, several operators and several organisations.

With regard to the multi-models based approach, future research should investigate the generalizability of this approach to other application domains not only the ATM but is also suitable to other large scale STSs (even though some adjustments might be required to manage their idiosyncrasies).

Lastly, variability and resonance aspects are of great importance when system failures and/or operators errors occur. Future work should cover the cases that have been dealt with in previous work which has been done in the area of systems reconfiguration (Navarre, Palanque, & Basnyat, 2008) and in the area of systematic account for human error using task models (Palanque & Basnyat, 2004).

Personal publications

Ragosta, M. (2015) **Observing Resilience: Air Traffic Control Centre contribution to everyday operations**, In International Symposium on Resilience Engineering (**REA Symposium 2015**), Lisbon, Portugal, June 22-25, 2015 (accepted)

Herrera, I.A., Pasquini, A., Ragosta, M., Vennesland A. (2014) **The SCALES framework for identifying and extracting resilience related indicators: preliminary findings of a go-around case study**, In SESAR Innovation Days 2014, Madrid, Spain, November 25-27, 2014. SESAR Work Package E.

Labunets, K., Massacci, F., Paci, F., Ragosta, M., Solhaug, B., Stølen, K., Tedeschi, A. (2014) **A first empirical evaluation framework for security risk assessment methods in the ATM domain,** In SESAR Innovation Days 2014, Madrid, Spain, November 25-27, 2014. SESAR Work Package E.

Martinie, C., Palanque, P., Pasquini, A., Ragosta, M., Silvagni, S., Sujan, M.A, Rigaud, E., Hollnagel, E. (2013). **Modelling of Automation Degradation: a Case Study**. In SESAR Innovation Days 2013. SESAR Work Package E.

Martinie, C., Palanque, P., Ragosta, M., Sujan, M. A., Navarre, D., (2013) **Understanding functional resonance through a federation of models: preliminary findings of an avionics case study**, In International Conference on Computer Safety, Reliability and Security (**SafeComp 2013**), 24 – 27 September, Toulouse, France.

Martinie, C., Palanque, P., Ragosta, M. (2013) **Extending Procedural Task Models by Explicit and Systematic Integration of Objects, Knowledge and Information**, In European Conference on Cognitive Ergonomics (**ECCE 2013**), 26 – 28 August, Toulouse, France.

Rigaud, E., Hollnagel, E., Martinie, C., Palanque, P., Pasquini, A., Ragosta, M., Silvagni, S., Sujan, M.A, (2013), **Modeling trade-offs consequences propagation and their impacts**, In International Symposium on Resilience Engineering (**REA Symposium 2013**), Soesterberg, The Netherlands, June 25-27, 2013.

Ragosta, M. (2013) **A multi-modelS based approach for the analysis and design of usable and resilient partly autonomous interactive systems**, In International Conference on Application and Theory of Automation in Command and Control Systems (**ATACCS 2013**), Naples, Italy, May 29-31, 2013.

Navarre, D., Martinie, C., Palanque, P., Ragosta, M., Pasquini, A. (2013), **Model-Based Dynamic Distribution of User Interfaces of Critical Interactive Systems**, In International Conference on Application and Theory of Automation in Command and Control Systems (**ATACCS 2013**), Naples, Italy, May 29-31, 2013.

Martinie, C., Palanque, P., Pasquini, A., Ragosta, M., Rigaud, E., Silvagni, S., Sujan, M.A. (2012) **A Framework for Modelling the Consequences of the Propagation of Automation Degradation: Application to Air Traffic Control Systems**. In SESAR Innovation Days (**SID 2012**), Braunschweig, Germany, November 27–29, 2012.

Martinie, C., Palanque, P., Pasquini, A., Ragosta, M., Rigaud, E., Silvagni, S.(2012) **Using Complementary ModelS-Based Approaches for Representing and Analysing ATM Systems' Variability**. In International Conference on Application and Theory of Automation in Command and Control Systems (**ATACCS 2012**), London, UK, May 28-31, 2012.

Ragosta, M. (2012) **Effective Automation in ATM: a new perspective on management and control of automated system**. In International Conference on Application and Theory of Automation in Command and Control Systems (**ATACCS 2012**), London, UK, May 28-31, 2012.

Martinie, C., Palanque, P., Ragosta, M. (2012) **Some Issues with Interaction Design and Implementation in the Context of Autonomous Interactive Critical Systems**. In CHI 2012 workshop: End-user interactions with intelligent and autonomous systems (**CHI 2012**), 05 - 12 May, Austin, Texas.

Hollnagel, E., Martinie, C., Palanque, P., Pasquini, A., Ragosta, M., Rigaud, E., Silvagni, S. (2011) **System Performances under Automation Degradation (SPAD)**, In SESAR Innovation Days (**SID 2011**), 29 November-1 December, Toulouse, France.

Barboni, E., Martinie, C., Palanque, P., Ragosta, M. (2011) **Task-Model Based Assessment of Automation Levels: Application to Space Ground Segments**, 2011 In IEEE International Conference on Systems, Man and Cybernetics (**SMC 2011**), 9-11 October, Anchorage, Alaska, IEEE Computer Society - Conference Publishing Services, 2011, pp. 3267-3273.

Ragosta, M. (2011), **Effective Automation in ATM: a new perspective on management and control of automated system**, In International Conference on Application and Theory of Automation in Command and Control Systems (**ATACCS 2011**), 26-27 May, Barcelona, Spain, IRIT Press, 2011, pp. 158-161.

Barboni, E., Lanzi, P., Martinie, C., Palanque, P., Pasquini, A., Ragosta, M. and Winckler, M. (2011), **Formal Tasks and Systems Models as a Tool for Specifying and Assessing Automation Designs**, In International Conference on Application and Theory of Automation in Command and Control Systems (**ATACCS 2011**), 26-27 May, Barcelona, Spain, IRIT Press, 2011, pp. 46-50.

References

Accot, J., & Zhai, S. (1997). Beyond Fitts' Law: Models for Trajectory-Based HCI Tasks. *Proceedings of CHI '97.* Atlanta, Georgia: ACM Press.

Adamski, A., & Westrum, R. (2003). Requisite imagination. The fine art of anticipating what might go wrong. In E. Hollnagel, *Handbook of cognitive task design* (pp. 193-220). Mahwah, NJ: Lawrence Erlbaum Associates.

Africha, H., Haffane, I., Lassalle, A., & Riegert, R. (2011). *Synopsat: interface de supervision satellite segment sol.* Toulouse, France.

AGILE. (2001). *Manifesto for Agile Software Development, http://agilemanifesto.org/.*

Aguinis, H., & Kraiger, K. (2009). Benefits of Training and Development for Individuals and Teams, Organizations, and Society. *Annual Review of Psychology*, pp. 451-475, vol. 60.

Airlines Electronic Engineering Committe. (2002, Avril 22). ARINC 661 specification: Cockpit Display System Interfaces To User Systems. Aeronautical Radio, Inc.

Amokrane, K., Lourdeaux, D., Burkhardt, J.-M., & Barthès, J.-P. (2008). An Intelligent Tutoring System for Training and Learning in a Virtual Environment for High-Risk Sites. *IEEE International Conference on Tools with Artificial Intelligence, ICTAI '08.* Dayton, Ohio: IEEE.

Anderson, J. R., John, B. E., Just, M. A., Carpenter, P. A., Kieras, D. E., & Meyer, D. E. (1995). Production system models of complex cognition. *Proc. of the Seventeenth Annual Conference of the Cognitive Science Society.* Hillsdale: Lawrence Erlbaum Associates.

Anderson, J., Matessa, M., & Douglass, S. (1995). The ACT-R theory and visual attention. *Seventeenth Annual Conference of the Cognitive Science Society* (pp. 61-65). Hillsdale, NJ: Lawrence Erlbaum Associates.

Anett, J. (2004). Hierarchical Task Analysis. In S. N. Diaper Dan, *The Handbook of Task Analysis for Human-Computer Interaction* (pp. pp. 67-82). Mahwah, New Jersey: Lawrence Erlbaum Associates.

Annett, J. (2004). Hierarchical Task Analysis. In D. Diaper, & N. Stanton, *The Handbook of Task Analysis for Human-Computer Interaction.* Lawrence Erlbaum Associates.

Apitz, G., Guimbretière, F., & Zhai, S. (2008, May). Foundations for designing and evaluating user interfaces based on the crossing paradigm. *ACM Transactions on Computer Human Interactions*.

Avizienis, A., Laprie, J.-C., Randell, B., & Landwehr, C. (2004). Basic Concepts and Taxonomy of Dependable and Secure Computing. *IEEE Transactions on dependable and secure computing, 1*(1).

Azzouzi, Y., André, F., & Hoarau, R. (2009). *Edition centrée utilisateur de modèle: application au Modèle de tâches.* Toulouse.

Bailey, G. (1993). Iterative methodology and designer training in human-computer interface design. *INTERACT and CHI conference on Human factors in computing systems.* Amsterdam: ACM.

Bainbridge, L. (1983). Ironies of automation. *Automatica, 19*(6), 775-779.

Balbo, S., Ozkan, N., & Paris, C. (2004). Choosing the Right Task Modeling Notation: A Taxonomy. In D. D. N., *The Handbook of Task Analysis for Human-Computer Interaction* (pp. 445-465). Mahwah, New Jersey: Lawrence Erlbaum Associates.

Barboni, E., Bastide, R., Lacaze, X., Navarre, D., & Palanque, P. (2003). Petri Net Centered versus User Centered Petri Nets Tools. *10th Workshop Algorithms and Tools for Petri Nets, AWPN 2003.* Eichstätt, Allemagne: Springer.

Barboni, E., Conversy, S., Navarre, D., & Palanque, P. (2006). Model-Based Engineering of Widgets, User Applications and Servers Compliant with ARINC 661 Specification. *DSVIS 2006*, (pp. 25-38).

Barboni, E., Ladry, J.-F., Navarre, D., Palanque, P., & M., W. (2010). Beyond Modelling: An Integrated Environment Supporting Co-Execution of Tasks and Systems Models. *ACM SIGCHI conference Engineering Interactive Computing Systems (EICS 2010).* Berlin, Allemagne: ACM SIGCHI.

Barboni, E., Navarre, D., Palanque, P., & Bazalgette, D. (2006). PetShop: A Model Based Tool for the Formal Modelling and Simulation of Interactive Safety Critical Embedded Systems. *International Conference on Human Computer Interaction in Aeronautics, HCI Aero.* Seattle: ACM.

Barnard, P., & Teasdale, J. (1991). Interacting cognitive subsystems: A systemic approach to cognitive-affective interaction and change. *Cognition and Emotion, vol. 5*, pp. 1-39.

Barthet, M.-F., & Tarby, J. C. (1996). The Diane+ method. *Computer-aided design of user interfaces*, (pp. 95-120).

Basnyat, S. (2006). A generic integrated modelling framework for the analysis, design and validation of interactive safety-critical and error-tolerant systems. *PhD Thesis*.

Basnyat, S., Palanque, P., Bernhaupt, R., & Poupart, E. (2008). Formal Modeling of Incidents and Accidents as a Means for Enriching Training Material for Satellite Control Operations. *Joint ESREL 2008 and 17th SRA-Europe Conference. 22 - 25 September, Valencia.*

Bass, L., Little, R., Pellegrino, R., Reed, S., Seacord, R., Sheppard, S., & Szezur, M. (1991). The Arch model: Seeheim revisited. *User Interface Developpers' workshop.*

Bastide, R., & Palanque, P. (1995). A Petri Net Based Environment for the Design of Event-Driven Interfaces. *16th International Conference on Application and theory of Petri Nets (ATPN'95).* Turin, Italie: Springer Verlag.

Bastide, R., & Palanque, P. (2001). Modeling a Groupware Editing Tool with Cooperative Objects. In G. Agha, & F. De Cindio, *Concurrent Object-Oriented Programming and Petri Nets.* Springer-Verlag.

Bastide, R., & Palanque, P. (2003). UML for Interactive Systems: What is missing. *Workshop on software engineering and HCI, INTERACT 2003.* Zurich, Suisse: Springer verlag LNCS.

Bastide, R., Navarre, D., Palanque, P., Schyn, A., & Dragicevic, P. (2004). A model-based approach for real-time embedded multimodals ystems in military aircraft. *Proceedings of the sixth international conference on multimodal interfaces.* Pennsylvania State University, USA: ACM Press.

Bastien, C., & Scapin, D. (1993). *Ergonomic Criteria for the Evaluation of Human-Computer interfaces, rapport technique n°156.* Rocquencourt, France: Institut National de Recherche en Informatique et en Automatique.

Baudisch, P., Cutrell, E., Robbins, D., Czerwinski, M., Tand-ler, P., Berderson, B., & Zierlinger, A. (2003). Drag-and-Pop and Drag-and-Pick: Techniques for Accessing Remote Screen Content on Touch-and-Pen-operated Systems. *IFIP TC 13 International Conference on Human Computer Interactions (INTERACT 2003)* (pp. 57-64). Zurich, Suisse: IOS Press.

Baumeister, L., John, B., & Byrne, M. (2000). A Comparison of Tools Building GOMS Models Tools for Design. *ACM International Conference on Human Factors in Computing Systems (CHI).* La Hague: ACM Press.

Baxter, G., & Sommerville, I. (2011). Socio-technical systems: From design methods to systems engineering. *Interacting with Computers, 23*(1), 4-17.

Beaudouin-Lafon, M. (1997). *Ingénierie des Systèmes Interactifs*. Consulté le septembre 14, 2011, sur Michel Beaudouin-Lafon: http://www.lri.fr/~mbl/ENS/IHM/ecole-in2p3/Cours/cours1.html

Beck, K. (1999). *Extreme Programming Explained.* Palo Alto, CA: Addison-Wesley.

Bedwell, W., & Salas, E. (2010, September). Computer-based training: capitalizing on lessons learned, vol. 14, issue 3. *International Journal of Training and Development*, pp. 239-249.

Berge, T., & Van Hezewijk, R. (1999). Procedural and Declarative Knowledge: An Evolutionary Perspective. *Theory and Psychology, 9*(5), 605-624.

Bernhaupt, R., Navarre, D., Palanque, P., & Winckler, M. (2007). Model-Based Evaluation: A New Way to Support Usability Evaluation of Multimodal Interactive Applications. In T. H. Law E., *Maturing Usability: Quality in Software, Interaction and Quality (chapter 5).* Springer Verlag.

Bertin, J. (1967). *Sémiologie Graphique - Les diagrammes - les réseaux - les cartes.* Gauthier-Villars et Mouton & Cie.

Bias, R., & Mayhew, D. (2005). *Cost-Justifying Usability, Second Edition: An Update for the Internet Age (Interactive Technologies).* Morgan Kaufmann.

Bindewald, J. M., Miller, M. E., & Peterson, G. L. (2014). A function-to-task process model for adaptive automation system design. *International Journal of Human-Computer Studies, 72*(12), 822-834.

Blanch, R., & Ortega, M. (2011). Benchmarking pointing techniques with distractors: adding a density factor to Fitt's pointing paradigm. *International conference on human factors in computing systems (CHI'11)* (pp. 1269-1638). Vancouver, Canada: ACM.

Blanch, R., Guiard, Y., & Beaudoin-Lafon, M. (2004). Semantic pointing: improving target acquisition with control-display ratio adaptation. *International Conference on Human Factors in Computing Systems* (pp. 519-526). Vienne, Autriche: ACM.

Blumendorf, M., Lehman, G., Feuerstack, S., & Albayrak, S. (2008). Executable Models for Human-Computer Interaction. *DSV-IS. 5136.* Kingston, Canada: Springer.

Bodart, F., Hennebert, A., Leheureux, J., & Vanderdonkt, J. (1994). Towards a dynamic strategy for computer-aided visual placement. *Workshop on Advanced Visual interfaces.* Bari, Italie.

Boehm, B. (1986). A spiral model of software development and enhancement. *SIGSOFT Software Engineering Notes, XI*(4, pp.14-24).

Bouti, A., & Kadi, D. A. (1994). A state-of-the-art review of FMEA/FMECA . *International Journal of reliability, quality and safety engineering, 1*(04), 515-543.

Branson, R., & Grow, G. (1987). Instructional systems development. In R. Gagné, *Instructional Technologies: Foundations* (pp. 397-428). Hillsdale, NJ: Lawrence Erlbaum.

Branson, R.K., Rayner, G.T., Cox, J.L., Furman, J.P., King, F.J., & Hannum, W.H. (1975). *Interservice procedures for instructional systems development. Phases I, II, III, IV, V, and executive summary. (TRADOC Pamphlet 350-30).* Fort Monroe, VA: U.S. Army Training and Doctrine Command.

Breedvelt, I., Paterno, F., & Severiins, C. (1997). Reusable structures in task models. *DSVIS* (pp. 251-265). Grenade, Espagne: Springer-Verlag.

British Standards Institution. (2002). *Hazard and operability studies (HAZOP studies)- Application Guide.* British Standard.

Brown, M., & Leveson, N. (1998). Modeling controller tasks for safety analysis. *International Workshop on Human Error and System Development.* Seattle.

Bye, A., Hollnagel, E., & Brendeford, T. S. (1999). Human–machine function allocation: a functional modelling approach. *Reliability Engineering and System Safety, 64*, 291–300.

Cacciabue, P. (2004). Human error risk management for engineering systems: a methodology for design, safety assessment, accident investigation and training. *Reliability Engineering and System Safety*(83, pp. 229–240).

Caffiau, S., Scapin, D., Girard, P., Baron, M., & Jambon, F. (2010). Increasing the expressive power of task analysis: Systematic comparison and empirical assessment of tool-supported task models. *Interacting with Computers 22*, pp. 569-593.

Calvary, G., Coutaz, J., Thévenin, D., Bouillon, L., Florins, L., Limbourg, Q., . . . Santoro, C. (2002). *The CAMELEON Reference framework, R&D Project IST 2000 30104.*

Calvary, G., Coutaz, J., Thévenin, D., Limbourg, Q., Bouillon, L., & Vanderdonckt, J. (2003). A unifying reference framework for multi-target user interfaces. *Interacting with Computers, 15*(3).

Cañas, A. J., Carff, R., Hill, G., Carvalho, M., Arguedas, M., Eskridge, T., . . . Carvajal, R. (2005). Concept maps: Integrating knowledge and information visualization. In S.-O. Tergan, & T. Keller, *Knowledge and information visualization: Searching for synergies* (pp. 205-2019). Heidelberg, NY: Springer Lecture Notes in Computer Science.

Cañas, A. J., Hill, G., & Lott, J. (2004). *Support for constructing knowledge models in CmapTool.* Pensacola, FL: Institute for Human and Machine Cognition.

Card, S. K., Newell, A., & Moran, T. P. (1983). *The Psychology of Human-Computer Interaction.*

Card, S., Moran, T., & Newell, A. (1986). The Model Human Processor: An Engineering Model of Human Performance. *Handbook of Perception and Human Performance*, 1-35.

Card, S., Moran, T., & Newell, A. (1986). *The Model Human Processor: An Engineering Model of Human Performance. Handbook of Perception and Human Performance.*

Chen, P. (1976). The Entity-Relationship Model - Toward a Unified View of Data. *ACM Transactions on Database Systems, 1*(1).

Clark, D. R. (2004). *Instructional System Design Concept Map.* Consulté le juillet 2011, sur http://nwlink.com/~donclark/hrd/ahold/isd.html

Cockburn, A. (2002). *Agile Software Development.* Boston, MA: Addison-Wesley.

Coffey, J. W., Carnot, M. J., Feltovich, P. J., Feltovich, J., Hoffman, R. R., Cañas, A. J., & Novak, J. D. (2003). *A Summary of Literature Pertaining to the Use of Concept Mapping Techniques and Technologies for Education and Performance Support* . Pensacola, FL: Institute for Human and Machine Cognition.

Collins, A. M., & Quillian, M. R. (1969). Retrieval time from semantic memory. *Journal of verbal learning and verbal behavior, 8*(2), pp. 240-247.

Collins, A. M., & Quillian, M. R. (1969). Retrieval time from semantic memory. *Journal of verbal learning and verbal behavior, 8*(2), 240-247.

Collins, D. (1995). *Designing Object-Oriented user interfaces.* Readwoods City, CA: Benjamin/Cummings Publishing, Inc.

Coutaz, J., Nigay, L., Salbert, D., Blandford, A., May, J., & Young, R. (1995). Four Easy Pieces for Assessing the Usability of Multimodal in Interaction : the CARE Properties. *Human Computer Interaction, INTERACT'95.* Lillehammer: Chapman & Hall (IFIP).

Coutaz, J., Paterno, P., Faconti, G., & Nigay, L. (1993). A comparison of approaches for specifying multimodal interactive systems. *Proceedings of ERCIM Workshop on Multimodal Human-Computer Interaction*, (pp. 165-174). Nancy, France.

Curtis, B., & Hefley, B. (1994). A WIMP no more: the maturing of user interface engineering. *Interactions, 1*(1).

David, H. (2000). Measures of stress/strain on air traffic controllers in simulated air traffic control. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* (pp. 680-680). SAGE Publications.

de Jong, T., & Ferguson-Hessler, M. G. (1996). Types and qualities of knowledge. *Educational Psychologist, 31*, 105-113.

Dennis, A., & Valachich, J. (1993). Computer Brainstorms: More Heads are Better than One. *Journal of Applied Psychology, 78*(4).

Diaper, D. (2001). Task analysis for knowledge descriptions (TAKD): A requiem for a method. *Behaviour and Information Technology, 20*(3), pp. 199-212.

Diaper, D. (2004). Understanding task analysis for Human-Computer Interaction. In D. Diaper, & N. Stanton, *The handbook of task analysis for human-computer interaction.* Lawrence Erlbaum Associates.

Diaper, D., & Stanton, N. (2004). *The handbook of task analysis for human-computer interaction.* Lawrence Erlbaum Associates.

Diaper, D., & Stanton, N. (2004). *The Handbook of Task Analysis for Human-Computer Interaction.* Lawrence Erlbaum Associates.

Diehl, M., & Stroebe, W. (1987). Productivity Loss in Brainstorming Groups: Towards a Solution of a Riddle. *Journal of Personality and Social Psychology, 53*(3).

Dix, A. (1995). Formal methods: an introduction to and overview of the use of formal methods within HCI. In M. A., & G. N., *Perspectives on HCI: diverse approaches.* Academic Press.

Dix, A., Finlay, J., Abowd, G., & Beale, R. (2003, 3rd edition). *Human-Computer Interaction.* Harlow: Pearson Education.

do Nascimento Neto, J. A., Vieira, M. F., Santoni, C., & Scherer, D. (2009). Proposing Strategies to Prevent the Human Error in Automated Industrial Environments. *Lecture Notes in Artificial Intelligence, 5638*, 279-288.

Dragicevic, P., & Fekete, J. (2001). Input Device Selection and Interaction Configuration with ICON. *People and Computer XV - Interaction without frontier (Joint proceedings of HCI 2001 and IHM 2001)* (pp. 543-558). Spring Verlag.

DREAMER. (s.d.). *DREAM*. Récupéré sur http://www.irit.fr/recherches/ICS/softwares/dream/

Duke, D., & Harrison, M. (1994). *Connections: From A(V) to Z.* Cambridge: ESPRIT BRA 7040 Amodeus-2.

Duke, D., Faconti, G., Harrison, M., & Paternò, F. (1994). Unifying views of interactors. *Proceedings of the Workshop on Advanced Visual Interfaces* (pp. 143-152). Bari, Italy: ACM Press.

Duncan, J., & Annett, K. (1967). Task Analysis and Training Design. *Journal of Occupational Psychology, 41*, 211-221.

El-Chaar, J., Boer, C., Pedrazzoli, P., Mazzola, S., & Dal Maso, G. (2011). Interactive 3D Virtual Environments for Industrial Operation Training and Maintenance. *International Conference on Reliability, Maintainability and Safety, ICRMS'2011* (pp. 1376-1381). Guiyang: IEEE.

Emery, F., & Trist, E. (1960). Management Science Models and Techniques, vol. 2. In C. Churchman, & M. Verhulst, *Socio-technical systems. :* (pp. 83–97). Pergamon, Oxford, UK;.

Endsley, M., & Kaber, D. (1999). Level of automation effects on performance, situation awareness and workload in a dynamic control task. *Ergonomics, 42*, 462– 492.

Ericson, C. (1999). Fault Tree Analysis - A History. *Proceedings of the 17th International Systems Safety Conference.*

Esteban, O., Chatty, S., & Palanque, P. (1995). Whizz'ed: a Visual Environment for building Highly Interactive Software. *IFIP International Conference on Human-Computer Interaction (Interact'95)* (pp. 121-127). Lillehammer, Norvège: Springer.

EUROCONTROL. (2003, Novembre 6). *Eurocontrol Safety Regulatory Requirement (ESARR) 6, Software in ATM Systems*.

EUROCONTROL. (2003). European Air Traffic Management Program. *ATCO Basic Training - Training plans*. European Organisation for the Safety of Air Navigation.

EUROCONTROL. (2004). European Air Traffic Management Program. *EATM Training Progression and Concepts, HRS/TSP-006-GUI-07*. European Organisation for the Safety of Air Navigation.

EUROCONTROL. (2004). *Review of Techniques to support the EATMP Safety Assessment Methodology* (Vol. V(1)). EEC Note No. 01/04.

EUROCONTROL. (2009). *A white paper on Resilience Engineering for ATM.*

European Aviation Safety Agency. (2007, septembre 19). *Certification Specifications for Large Aeroplanes, CS-25, Amendment 3*. Cologne, Allemagne: EASA.

European Cooperation for Space Standardization. (2008). Space Engineering, Ground Systems and Operations, ECSS-E-70C.

European Organisation for Civil Aviation Equipment. (1992). DO-178B, Software Consideration in Airborne Systems and Equipment Certification. EUROCAE.

European Space Agency. (1994). ESA PSS-05-0. Software Engineering Standards, Issue 2, Revision 1. Noordvijk, The Netherlands: ESA Publications Division.

Feary, M., Sherry, L., Polson, P., & Palmer, E. (2000). Evaluation Of A Formal Methodology For Developing Aircraft Vertical Flight Guidance Training Materia. *International Conference on Human-Computer Interaction in Aeronautics.* Toulouse, France.

Federal Aviation Administration. (2011, Septembre). *Training*. Récupéré sur FAA Human Factors: http://www.hf.faa.gov/webtraining/Training/Training1.htm

Fitts, P. M. (1954). The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, 381-391.

Fu, W.-T., Bothell, D., Douglass, S., Haimson, C., Sohn, M.-H., & Anderson, J. (2006). Toward a real-time model-based training system. *Interacting with Computers, vol. 18*, pp. 1215-1241.

Gaffar, A., Sinnig, D., Seffah, A., & Forbrig, P. (2004). Modeling patterns for task models. *International Conference on TAsk MOdels and DIAgrams* (pp. 99-104). New-York, NY: Springer-Verlag.

Gagné, R. (1985). *The Conditions of Learning and the Theory of Instruction.* New York: Holt, Rinehart, and Winston.

Gagné, R. (1987). *Instructional Technology: Foundations.* Hillsdale, NJ: Lawrence Erlbaum.

Gagné, R., Wager, W., Golas, K., & Keller, J. (2005). *Principles of Instructional Design.* Florence, KY: Thomson Wadsworth.

Genrich, H. (1991). Predicate/transitions nets. In K. Jensen, & G. Rozenberg (Éd.), *High-levels Petri nets:theory and application* (pp. 3-43). Springer Verlag.

Giese, M., Mistrzik, T., Pfau, A., Szwillus, G., & von Detten, M. (2008). AMBOSS: A Task Modelling Approach for Safety-Critical Systems. *HCSE/TAMODIA. 5247, pp. 98-109.* Pise, Italie: Springer.

Gore, B., Hooey, B., Haan, N., Bakowski, D., & Mahlstedt, E. (2011). A Methodical Approach to Build Valid Human Performance Models of Flight Deck Operations. *International Conference in Human Computer Interaction, HCII 2011.* Orlando, Florida.

Graham, C., & Cockton, G. (1996). *Design principles for Interactive Software.* London: Chapman & Hall.

Greenberg, S. (2004). Working through Task-Centered System Design. In D. Diaper, & N. Stanton, *The Handbook of Task Analysis for Human-Computer Interaction.* Lawrence Erlbaum Associates.

Groeneweg, J. (2002). *Controlling The Controllable: Preventing Business Upsets. TRIPOD business management series 1.* Global safety group.

Grossman, T., & Balakrishnan, R. (2005). The bubbe cursor: enhancing target acquisition by dynamic resizing of the cursor's activation area. *International Conference on Human Factors in Computing Systems (CHI'05)* (pp. 281-290). Portland, Oregon: ACM.

Gulliksen, J., & Goransson, B. (2003). Usability Design: Integrating User Centered System Design in the Software Development Process. *IFIP International Conference on Human-Computer Interaction, INTERACT.* Zurich, Suisse: Springer.

HALA! SESAR Research Network. (2012). *HALA! Position paper.* 2.04.

Hamon, A., Palanque, P., Silva, J. L., Deleris, Y., & Barboni, E. (2013). . Formal description of multi-touch interactions. *Proceedings of the 5th ACM SIGCHI symposium on Engineering interactive computing systems* (pp. 207-216). ACM.

Harel, D. (1987). StateCharts: A Visual Formalism for Complex Systems. *Science of Computer Programming, vol. 8, n. 3*, pp. 231-274.

Harrison, M., Johnson, P., & Wright, P. (2002). Automating functions in multi-agent control systems: supporting the decision process. In F. Redmill, & T. Anderson (Éd.), *Proceedings of the Tenth safety-critical system symposium* (pp. 93-106). Southampton: Springer.

Hart, S. G. (2006). NASA-Task Load Index (NASA-TLX); 20 Years Later. *Proceedings of the Human Factors and Ergonomics Society 50th Annual Meeting* (pp. 904-908). Santa Monica, CA: Human Factors & Ergonomics Society.

Hartson, H., & Hix, D. (1989). Human-computer interface development: concepts and systems for its management. *ACM Computing Surveys, 21*(1).

Hassenzahl, M. (2003). The thing and I: understanding the relationship between user and product. In M. Blythe, C. Overbeeke, A. Monk, & P. Wright, *Funology: From Usability to Enjoy-ment* (pp. 31-42). Dordrecht: Kluwer.

Haugeneder, H., & Gehrke, M. (1986). A user friendly ATN programming environment (APE). *Coling '86 [8]*, (pp. 399–401).

Henry, N., & Fekete, J.-D. (2006). MatrixExplorer: a Dual-Representation System to Explore Social Networks. *IEEE Transactions on Visualization and Computer Graphics (Proceedings Visualization / Information Visualization), 12*(5).

Herrera, I., Pasquini, A., Ragosta, M., & Vennesland, A. (2014). *SCALES D1.1 - Preliminary specifications and guidelines.* Edition 00.01.00 (under SJU approval). Available online: http://www.hala-sesar.net/projects/blog/scales.

Hewett, Baecker, Card, Carey, Gasen, Mantei, . . . Verplank. (1996). *ACM SIGCHI Curricula for Human-Computer Interaction.* Consulté le 2011, sur ACM SIGCHI: http://old.sigchi.org/cdg/index.html

Hilliard, R. (2000). Ieee-std-1471-2000 recommended practice for architectural description of software-intensive systems. *IEEE*, 16-20.

Hinckley, K., Czerwinski, M., & Sinclair, M. (1998). Interaction and Modeling Techniques for Desktop Two Handed Input. *Association for Computing Machinery, Inc.*

Hix, D., & Harston, H. (1993). *Developing User Interfaces: ensuring usability through product and process.* New York: Wiley & Sons.

Hofer, E., & Ruggiero, F. (1990). Computer-Based Flight Training Evaluation. In D. Norrie, & H.-W. Six, *Computer Assisted Learning* (pp. 309-320). Springer Berlin / Heidelberg.

Hoffman, R. R., Coffey, J. W., Novak, J. D., & Cañas, A. J. (2005). Application of concept maps to web design and web work. In R. W. Proctor, & K.-P. L. Vu, *Handbook of human factors in web design* (pp. 156-175). Mawah, NJ: Lawrence Erlbaum Associates.

Hollnagel, E. (1998). *Cognitive reliability and error analysis method.* Oxford, UK: Elsevier Science Ltd.

Hollnagel, E. (1998). *Cognitive reliability and error analysis method.* Oxford, UK: Elsevier Science Ltd.

Hollnagel, E. (2004). *Barriers and accident prevention.* Aldershot, UK: Ashgate.

Hollnagel, E. (2006). Task Analysis: The Why, What and How. In G. Salvendy, *Handbook of Human Factors and Ergonomics (3rd Edition)* (pp. 373-383). New York: Wiley.

Hollnagel, E. (2008). From protection to resilience: Changing views on how to achieve safety. *8th International Symposium of the Australian Aviation Psychology Association.* Sydney, Australia.

Hollnagel, E. (2008). The changing nature of risks. *Ergonomics Australia, 22*(1), 33–46.

Hollnagel, E. (2011). Prologue: The Scope of Resilience Engineering. In E. Hollnagel, J. Paries, D. Woods, & J. Wreathall, *Resilience engineering in Practice* . Aldershot, UK: Ashgate.

Hollnagel, E. (2011). *Resilience Engineering in Practice.* Surrey: Ashgate Publishing Limited.

Hollnagel, E. (2012). *FRAM: The functional resonance analysis method for modelling complex socio-technical systems.* Farnham, UK: Ashgate.

Hollnagel, E. (2014). *Safety-I and Safety-II. The past and the future of safety management.* Ashgate.

Hollnagel, E., & Speziali, J. (2008). *Study on developments in accident investigation methods: A survey of the 'State-of-the-Art'".* Sophia Antinopolis Cedex: SKI.

Hollnagel, E., & Woods, D. (2006). *Joint Cognitive Systems: Foundations of Cognitive Systems Engineering.* Boca Raton, FL: CRC Press/Taylor & Francis Group;.

Hollnagel, E., Martinie, C., Palanque, P., Pasquini, A., Ragosta, M., Rigaud, E., & Silvagni, S. (2011). System Performances under Automation Degradation (SPAD). *The First SESAR Innovation Days (SID 2011).* Toulouse, France.

Hollnagel, E., Woods, D. D., & Leveson, N. (2005). *Resilience Engineering: Concepts and Precepts.* Brookfield, VT: Ashgate Publishing Company.

Höök, K., Isbister, K., Westerman, S., Gardner, P., Sutherland, E., Vasalou, A., . . . Petta, P. (2011). Evaluation of Affective Interactive Applications. In *Emotion-Oriented Systems.* Berlin: Springer.

Hopcroft, J., & Ullman, J. (1990). *Introduction To Automata Theory, Languages, And Computation.* Boston, MA, USA: Addison-Wesley Longman Publishing.

Hussman, H., Meixner, G., & Zuehlke, D. (2011). *Model-Driven Development of Advanced User Interfaces.* Berlin: Springer-Verlag.

International Atomic Energy Agency. (1998). *Experience in the use of Systematic Approach to Training (SAT) for Nuclear Power Plant Personnel, Technical Report, IAEA-TECDOC-1057.*

International Atomic Energy Agency. (2009). IAEA Nuclear Energy Series. *Managing Human Resources in the Field of Nuclear Energy, NG-G-2.1*. Vienna, Austria: IAEA.

International Electrotechnical Commission (IEC). (2006). *Fault Tree Analysis. Edition 2.0.006.* International Electrotechnical Commission.

International Standard Organisation. (2000). ISO TR 18529:2000 Ergonomics of human system interaction: Human-centred lifecycle process description.

International Standard Organisation. (2001). ISO IEC 9126: Software engineering -- Product quality.

International Standard Organisation. (2008). ISO/IEC 12207:2008(E), Systems and Software Engineering - Software Life Cycle Processes. ISO/IEC-IEEE.

International Standard Organization. (1996). *DIS 9241-11: Ergonomic requirements for office work with visual display terminals (VDT) - Part 11 Guidance on Usability*.

Isaac, A., Shorrock, S., & Kirwan, B. (2002). Human error in European air traffic management: The HERA project. *Reliability Engineering and System Safety, 75*(2), 257-272.

ISO. (1989). ISO 8807 Information Processing Systems - Open Systems Interconnection - LOTOS - A Formal Description Technique Based on temporal Ordering of Observational Behaviour. Genève, Suisse: ISO.

ISO. (2011). *42010:2011 Systems and software engineering - Architecture description.*

JAA. (2006). *JAR - FCL 1 - Flight Crew Licensing (Aeroplane)*. Joint Aviation Authorities.

Jackson, M. (1983). *System Development.* Englewoog Cliffs, NJ: Prentice Hall.

Jacob, R. (1999). A Software Model and Specification Language for Non-WIMP User Interfaces. *ACM Transactions on Human-Computer Interactions, vol. 6, n. 1*, pp. 1-46.

Jacob, R. J. (1986). A specification language for direct manipulation user interfaces. *ACM Transition Graphics, 5*(4), 283-317.

Jacob, R. J., Deligiannidis, L., & Morrison, S. (1999). A software model and specification language for non-WIMP user interfaces. *ACM Transactions on Computer-Human Interaction (TOCHI)*, *6(1)*, pp. 1-46.

Jameson, A. (2001). Modelling both the Context and the User. *Personal Ubiquitous Comput., 5*(1), 29-33.

Johnson, C. (1995). Using Z to support the design of interactive safety-critical systems in Software Engineering Journal. (march).

Johnson, P., & Johnson, H. (1989). Knowledge Analysis of Task; Tast Analysis and Specification for Human-Computer Systems. pp. 119-144.

Johnson, P., Johnson, H., Waddington, R., & Shouls, A. (1989). Task related knowledge structures: Analysis, modelling and application. In D. Jones, & R. Winder, *From Research to Implementation, People and Computers, IV.* Cambridge, UK: Cambridge University Press.

Johnson, P., Wilson, S., Markopoulos, P., & Pycock, Y. (1993). ADEPT-Advanced Design Environment for Prototyping with Task Models. *Proceedings of ACM Conference on Human Aspects in Computing Systems InterCHI'93* (p. 66). New York, NY, USA: ACM Press.

Jordan, N. (1963). Allocation of functions between man and machines in automated systems. *Journal of Applied Psychology, 47*, 161-165.

Jourde, F., Laurillau, Y., & Nigay, L. (2010). COMM Notation for Specifying Collaborative and MultiModal Interactive Systems. *ACM SIGCHI Symposium on Engineering Interactive Computing Systems, EICS 2010* (pp. 125-134). Berlin, Germany: ACM.

Kaber, D. B., & Endsley, M. R. (2004). The effects of level of automation and adaptive automation on human performance, situation awareness and workload in a dynamic control task. *Theoretical Issues in Ergonomics Science, 5*(2), 113-153.

Kenney, P., & Loftin, R. (1994). Virtual Environments in Training: NASA's Hubble Space Telescope Mission. *International Conference on Interservice/Industry Training Systems & Education.* Orlando, Florida.

Kincaid, J., & Westerlund, K. (2009). Simulation in Education and Training. *IEEE International Conference on Winter Simulation.* Austin, Texas: IEEE.

Kirkpatrick, D., & Kirkpatrick, J. (2006). *Evaluating Training Programs: The Four Levels, 3rd Edition.* Berrett-Koehler.

Koedinger, K. R., & Corbett, A. (2006). *Technology Bringing Learning Sciences to the Classroom. The Cambridge Handbook of the Learning Sciences.* Cambridge University Press.

Komatsu, L. (1992). Recent Views of Conceptual Structure. *Psychological Bulletin, 112*(4), pp. 500-526.

Komatsu, L. (1992). Recent Views of Conceptual Structure. *Psychological Bulletin, 112*(4), pp. 500-526.

Kruchten, P. (2004). *The Rational Unified Process: an Introduction* (Vol. 3rd edition). Boston, MA: Pearson Education, Inc.

Lacaze, X. (2005). La conception rationalisée pour les systèmes interactifs. *PhD Thesis*.

Lacaze, X., Palanque, P., Barboni, E., Bastide, R., & Navarre, D. (2006). From DREAM to Reality : Specificities of Interactive Systems Development With Respect to Rationale Management. Springer Verlag.

Ladry, J.-F. (2010). *Une notion et un processus outillé pour le développement de systèmes interactifs multimodaux critiques.* Toulouse: Université de Toulouse.

Ladry, J.-F., Navarre, D., & Palanque, P. (2009). Formal Description Techniques to Support the Design, Construction and Evaluation of Fusion Engines for SURE (Safe Usable, Reliable and Evolvable) Multimodal Interfaces. *International Conference on Multimodal Interfaces.* 2009: ACM.

Lakos, C. (1991). *Language for Object-Oriented Petri Nets, #91-1.* University of Tasmania: Department of Computer Science.

Lakos, C., & Christensen, S. (1994). A General Systematic Approach to Arc Extensions for Coloured Petri Nets. *15th International Conference on Application and Theory of Petri Nets. LNCS no. 815.* Berlin: Springer.

Law, E. L.-C., Scapin, D., Cockton, G., Springett, C., Stary, C., & Winckler, M. (2009). *Maturation of Usability Evaluation Methods: Retrospect and Prospect.*

Leake, D. B., Maguitman, A., & Reichherzer, T. (2004). Understanding knowledge models: Modeling assessment of concept importance in concept maps. In R. Alterman, & D. Kirsch (Éd.), *Proceedings of the twenty-sixth annual conference of the cognitive science society* (pp. 795-800). Mahwah, NJ: Lawrence Erlbaum.

Lee, G., Howard, J., & Anderson, P. (2002). Safety-critical requirements specification and analysis using SpecTRM. *Proceedings of the 2nd Meeting of the US Software System Safety Working Group* .

Leveson, N. G. (2004). A New Accident Model for Engineering Safer Systems. *Science, 42*(4), 237-270.

Leveson, N., & Brown, M. (1998). Modeling Controller Tasks for Safety Analysis. *HESSD 98.*

Leveson, N., Dulac, N., Zipkin, D., Cutcher-Gershenfeld, J., Carroll, J., & Barrett, B. (2007). Engineering resilience into safety-critical systems. In E. Hollnagel, D. D. Woods, & N. & Leveson, *Resilience Engineering–Concepts and Precepts* (pp. 95-123). Ashgate Aldershot.

Lim, K., Long, J., & Silcock, N. (1992). Integrating human factors with the Jackson System Development method: an illustrated overview. *Ergonomics, 35*(10).

Limbourg, Q., & Vanderdonckt, J. (2004). Comparing Task Models for User Interface Design. In D. D. N., *The Handbook of Task Analysis for Human-Computer Interaction* (pp. 135-154). Mahwah, New Jersey: Lawrence Erlbaum Associates.

Limbourg, Q., Pribeanu, C., & Vanderdonckt, J. (2001). Towards Uniformed Task Models in a Model-Based Approach. *DSV-IS '01*, (pp. 164-182).

Limbourg, Q., Vanderdonckt, J., Michotter, M., Bouillon, L., & Lopez-Jaquero, V. (2004). USIXML: A Language Supporting Multi-path Development of User Interfaces. *In proceedings of EHCI-DSVIS 2004.* LNCS.

Loftin, R., & Kenney, P. (1994). Virtual Environments in Training: NASA's hubble space telescope mission. *International Conference on Interservice/Industry Training Systems & Education Conference.* Orlando, Florida: NTSA.

Lu, S., Paris, C., & Vander Linde, K. (1998). Towards the automatic generation of task models from object oriented diagrams. *IFIP Working Conference on Engineering for Human-Computer Interaction.* Crète, Grèce: Kluwer academic publishers.

Lu, S., Paris, C., Vander Linde, K., & Colineau, N. (2003). Generating UML Diagrams From Task Models. *CHINZ.* Dunedin, Nouvelle Zélande.

Macchi, L. (2010). A Resilience Engineering approach for the evaluation of performance variability: development and application of the Functional Resonance Analysis Method for air traffic management safety assessment. Doctoral dissertation, Paris, ENMP.

Macchi, L., Hollnagel, E., & Leonhard, J. (2009). Resilience Engineering Approach to Safety Assessment: An Application of FRAM for the MSAW system. *Proceedings of Eurocontrol Safety R&D Seminar*, *2*. Munich.

MacLean, A., Young, R. M., Bellotti, V. M., & Moran, T. P. (1991). Questions, options, and criteria: elements of design space analysis. *Hum.-Comput. Interact., 6*(3), 201-250.

Marquardt, N., Jota, R., Greenberg, S., & Jorge, J. (2011). The Continuous Interaction Space: Interaction Techniques Unifying Touch and Gesture On and Above a Digital Surface. *International Conference on Human Computer Interaction (INTERACT'11)* (pp. 461-476, vol. 6948). Lisbon: Springer.

Martensen, A. L., & Butler, R. W. (1987). *The Fault-Tree Compiler.* Hampton, VA, United States: NASA Langley Research Center.

Martinie, C. (2011). Une approche à base de modèles synergiques pour la prise en compte simultanée de l'utilisabilité, la fiabilité et l'opérabilité des systèmes interactifs critiques. Toulouse, France: PhD thesis.

Martinie, C., & Palanque, P. (2011). A Multi-Models Based Development Process for Critical Interactive Systems Integrating Formal and Informal Approaches. *Workshop on Combining Models and Design for interactive systems (ComDeisMoto 2011), IFIP TC 13 conference on Human Computer Interaction (INTERACT 2011).* Lisbonne, Portugal.

Martinie, C., Palanque, P., & Ragosta, M. (2013). Extending Procedural Task Models by Explicit and Systematic Integration of Objects, Knowledge and Information. *The 31st European Conference on Cognitive Ergonomics (ECCE 2013).* Toulouse, France.

Martinie, C., Palanque, P., & Winckler, M. (2011). Structuring and Composition Mechanism to Address Scalability Issues in Task Models. In P. Campos, T. Graham, J. Jorge, N. Nunes, P. Palanque, & M. Winckler (Éd.), *IFIP TC13 Conference on Human-Computer Interaction (INTERACT 2011). LNCS 6948*, pp. 589-609. Lisbon, Portugal: Springer Verlag.

Martinie, C., Palanque, P., Barboni, E., & Ragosta, M. (2011). Task-Model Based Assessment of Automation Levels: Application to Space Ground Segments . *IEEE International Conference on System, Man and Cybernetics, SMC 2011.* Anchorage, Alaska: IEEE.

Martinie, C., Palanque, P., Barboni, E., Winckler, M., Ragosta, M., Pasquini, A., & Lanzi, P. (2011). Formal Tasks and Systems Models as a Tool for Specifying and Assessing Automation Designs . *International Conference on Application and Theory on Automation in Command and Control Systems, ATACCS 2011.* Barcelone, Espagne: IRIT Press.

Martinie, C., Palanque, P., Navarre, D., & Barboni, E. (2012). A Development Process for Usable Large Scale Interactive Critical Systems: Application to Satellite Ground Segments. *IFIP WG13.2 conf. on Human Centric Software Engineering (HCSE 2012)* (pp. 72-93). Springer.

Martinie, C., Palanque, P., Navarre, D., & Barboni, E. (2012). A Development Process for Usable Large Scale Interactive Critical Systems: Application to Satellite Ground Segments. *IFIP WG13.2 conf. on Human Centric Software Engineering (HCSE 2012)* (pp. 72-93). Springer.

Martinie, C., Palanque, P., Navarre, D., & Winckler, M. (2010). A formal approach supporting effective and efficient training program for improving operators' reliability. *European Safety and Reliability Conference.* Rhodes, Grèce: ESRA.

Martinie, C., Palanque, P., Navarre, D., Winckler, M., & Poupart, E. (2011). Model-based Training: An Approach Supporting Operability of Critical Interactive Systems: Application to Satellite Ground Segments. *ACM SIGCHI International Conferences on Engineering Interactive Computing Systems.* Pise, Italie: ACM Press.

Martinie, C., Palanque, P., Winckler, M., & Conversy, S. (2010). DREAMER: a Design Rationale Environment for Argumentation, Modelling and Engineering Requirements. *ACM International Conference on Design of Communication, SIGDOC.* Sao Paulo, Brésil: ACM Press.

Mavin, A., & Maiden, N. (2003). Determining Socio-Technical Systems Requirements: Experiences with Generating and Walking Through Scenarios. *11th International Conference on Requirements Engineering.* Monterey Bay, CA, USA: IEEE Computer Society Press.

McConnell, S. (1996). *Rapid Development: Taming Wild Software. Schedules.* Redmond, WA: Microsoft Press.

McDermid, J., & Ripken, K. (1983). Life cycle support in the Ada environment. *ACM SIGAda Ada Letters, III*(1).

Memon, A., & Xie, Q. (2005). Studying the Fault-Detection Effectiveness of GUI Test Cases for Rapidly Evolving Software. *IEEE Transactions on Software Engineering, vol. 31, no. 10*, pp. 884-896.

Metzger, U., & Parasuraman, R. (2001). Conflict detection aids for air traffic controllers in free flight: Effects of reliable and failure modes on performance and eye movements. *Proceedings of the 11th International Symposium on Aviation Psychology.* Columbus, OH: Ohio State University.

Meyer, D. E., Annett, J., & Duncan, K. D. (1967). Task analysis and training design. *Journal of Occupational Psychology, 41*.

Meyer, D., Annett, J., & Duncan, K. D. (1967). Task analysis and training design. *Journal of Occupational Psychology*(41).

Miller, S., Tribble, A., Whalen, M., & Heimdal, M. (2006). Proving the shalls - Early validation of requirements through formal methods. *Software Tools for Technology Transfer, 8*(4).

Mollard, R., Cabon, P., Bourgeois-Bougrine, S., & David, H. (2000). Measurement of fatigue and sleepiness in ATC simulation. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* (pp. 208-211). SAGE Publications.

Myers, P., Watson, B., & Watson, M. (2008). Effective Training Programs Using Instructional Systems Design and E-Learning. *Process Safety Progress, 27*(2).

Narumi, T., Nishizaka, S., Kajinami, T., Tanikawa, T., & Hirose, M. (2011). Augmented reality flavors: gustatory display based on edible marker and cross-modal interaction. *International Conference on Human Factors in Computing Systems (CHI'11)* (pp. 93-102). Vancouver: ACM.

Natvig, M. K., & Westerheim, H. (2008). Refinement and evaluation of the ARKTRANS framework through use in travel information services. *15th World Congress on Intelligent Transport Systems and ITS America's Annual Meeting.*

Navarre, D., Palanque, P., & Basnyat, S. (2008). A Formal Approach for User Interaction Re-configuration of Safety Critical Interactive Systems. *In Proceedings of SAFECOMP '08* (pp. 373-386). Berlin, Heidelberg: Springer-Verlag.

Navarre, D., Palanque, P., & Bastide, R. (2004). A formal description technique for the behavioural description of interactive applications compliant with ARINC 661 specifications. *HCI-Aero'04.* Toulouse.

Navarre, D., Palanque, P., & Bastide, R. A. (2001). Tool-support design framework for safety critica linteractive systems in interacting with computers. *Elsevier, 15*(3), 309–328.

Navarre, D., Palanque, P., & Winckler, M. (2009). Task Models and System Models as a Bridge between HCI and Sofware Engineering. In *Human-Centered Software Engineering Software Engineering Models, Patterns and Architectures for HCI.* Springer.

Navarre, D., Palanque, P., Barboni, E., & Mistrzik, T. (2007). On the Benefit of Synergistic Model-Based Approach for Safety Critical Interactive System Testing. In Springer (Éd.), *TAMODIA*, *LNCS 4849.* Toulouse, France.

Navarre, D., Palanque, P., Barboni, E., & Mistrzyk, T. (2007). On the Benefit of Synergistic Model-based Approach for Safety Critical Interactive System Testing. *TAMODIA2007.*

Navarre, D., Palanque, P., Barboni, E., Ladry, J.-F., & Martinie, C. (2011, Janvier). Designing for resilience to hardware failures in interactive systems: A model and simulation-based approach. *Elsevier's Safety Science, Special issue : Reliability Engineering and System Safety, Vol. 96*, pp. 38-52.

Navarre, D., Palanque, P., Ladry, J.-F., & Barboni, E. (2009). ICOs: a Model-Based User Interface Description Technique dedicated to Interactive Systems Addressing Usability, Reliability and Scalability. *Transactions on Computer-Human Interaction, 16*(4).

Navarre, D., Palanque, P., Martinie, C., & Steere, S. (2011). Formal Description Techniques for Human-Machine Interfaces - ModelS-Based Approaches for the Design and Evaluation of Dependable Usable Interactive Systems. In G. A. Boy (Éd.), *The Handbook of Human-Machine Interaction. A Human-Centered Approach* (Vol. ISBN 978-0-7546-7580-8). Ashgate.

Neitzel, D. (2006, May, june). How to develop an effective training program. *IEEE Industry Applications Magazine*(May-June).

Nielsen, J. (1994). Heuristic evaluation. In J. Nielsen, & R. Mack, *Usability Inspection Methods.* New York, NY: John Wiley and Sons.

Norman, D. A. (1998). *The Invisible Computer.* MIT Press.

Norman, D. A. (2002). *The design of everyday things.* Basic Books.

Norman, D., & Drapper, S. (1986). *User Centred System Design.* U.S.: L. Erlbaum.

Novak, J. D., & Cañas, A. J. (2008). *The Theory Underlying Concept Maps and How to Construct and Use Them.* Technical Report, Florida: Institute for Human and Machine Cognition.

Nunes, N., & Cunha, J. (2000). Towards an UML profile for interactive systems development: the Wisdom approach. *UML conference.* Kent, UK: Springer Verlag LNCS.

Object Management Group. (2011). OMG Unified Modeling Language (OMG UML), Infrastructure, Version 2.4.

Okazaki, T., Muromura, M., & Kaynano, J. (2011). A Study on Evaluating Maneuvering skill and Developing Support Tool for Marine Pilot Trainees Berthing a Ship. *IEEE International Conference on Systems, Man and Cybernetics, SMC'2010* (pp. 3087-3093). Istanbul, Turquie: IEEE.

Older, M., Waterson, P., & Clegg, C. (1997). A critical assessment of task allocation methods and their applicability. *Ergonomics, 40*(2), 151–171.

Palanque, P. (1992). *Modélisation par objets coopératifs interactifs d'interfaces homme-machine dirigées par l'utilisateur.* Toulouse: Université Toulouse 1.

Palanque, P. (2011). Cours sur l'ingénierie des systèmes interactifs.

Palanque, P., & Basnyat, S. (2004). Task Patterns for taking into account in an efficient and systematic way both standard and erroneous user behaviours. *HESSD 2004 6th Int. Working Conference on Human Error, Safety and System Development*, (pp. 109-130).

Palanque, P., & Basnyat, S. (2004). Task Patterns for Taking Into Account in an Efficient and Systematic Way Both Standard and Erroneous User Behaviours. *International Conference on Human Error, Safety and Systems Development, HESSD 2004* (pp. 109-130). Toulouse, France: Springer.

Palanque, P., & Bastide, R. (1994). A Formalism for Reliable User Interfaces. *Workshop on Software Engineering and Human Computer Interaction associated with the 16th IEEE ICSE conference.* Sorrento, Italy.

Palanque, P., & Bastide, R. (1997). Synergistic Modelling of Tasks, Users and Systems Using Formal Specification Techniques. *Interacting With Computers, 9*(2).

Palanque, P., & Lacaze, X. (2007). DREAM-TEAM: A Tool and a Notation Supporting Exploration of Options and Traceability of Choices for Safety Critical Interactive Systems. *INTERACT.* Rio, Brésil: Springer Verlag.

Palanque, P., & Martinie, C. (2011). Contextual Help for Supporting Critical Systems' Operators: application to space ground segments. . *International workshop and academic/industrial consortium on Activity Context Representation: Techniques and Langages, (AAAI 2011).* San Francisco, USA.

Palanque, P., Barboni, E., Martinie De Almeida, C., Navarre, D., & Winckler, M. (2011). A Tool Supported Model-based Approach for Engineering Usability Evaluation of Interaction Techniques. *EICS.* Pisa, Italy: ACM.

Palanque, P., Basnyat, S., Blandford, A., Bernhaupt, R., Boring, R., Johnson, C., & Johnson, P. (2007). Beyond usability for safety critical systems: How to be SURE (safe, usable, reliable, and evolvable). *International Conference on Human Factors in Computing Systems (CHI'07)* (pp. 2133-2136). San Jose, California: ACM.

Palanque, P., Bastide, R., & Dourte, L. (1993). Contextual Help for Free with Formal Dialogue Design. *HCII 93.* Orlando.

Palanque, P., Bastide, R., & Paternò, F. (1997). Formal Specification As a Tool for the Objective Assessment of Safety Critical Interactive Systems. *INTERACT.* Sydney, Autralie: Chapman et Hall.

Palanque, P., Bastide, R., & Sengès, V. (1995). Validating Interactive System Design Through the Verification of Formal Task and System Models. *IFIP Working Conference on Engineering for Human-Computer Interaction.* Grand Targhee Resort, Wyoming, USA: Chapman et Hall.

Palanque, P., Ladry J-F., Navarre, D., & Barboni, E. (2009). High-Fidelity Prototyping of Interactive Systems can be Formal too. *13th International Conference on Human-Computer Interaction.* San Diego, CA, USA.

Palanque, P., Long, J., Tarby, J., Barthet, M., & Lim, K. (1994). Conceptions d'applications ergonomiques: une méthode pour informaticiens et une méthode pour ergonomes. *Ergonomie et Informatique Avancée.* Biarritz.

Palanque, P., Navarre, D., & Gaspard-Boulinc, H. (2000). *MEFISTO method version 1, ESPRIT Reactive LTR 24963, WP2-7.*

Paradies, M., & Unger, L. (2000). *TapRooT® - The system for Root Cause Analysis, Problem investigation, and proactive improvement.* Knoxville, Tennessee: System Improvements Inc.

Parasuraman, R., & Riley, V. (1997). Humans and automation: Use, misuse, disuse, abuse. *Human Factors: The Journal of the Human Factors and Ergonomics Society, 39*(2), 230-253.

Parasuraman, R., Sheridan, T., & Wickens, C. (2000, May). A model for types and levels of human interaction with automation. *IEEE Transactions on Systems, Mans and Cybernetics, Part A: Systems and Humans, vol. 30, n.3*, pp. 286-297.

Paris, C., Tarby, J., & Vander Linden, K. (2001). A Flexible Environment for Building Task Models. *Proceedings of the ICM-HCI 2001*, (pp. 313-330). Lille, France.

Paris, C., Vander Linden, K., & Lu, S. (1992). Automated Knowledge Acquisition for Instructional Text Generation. *Proceedings of SIGDOC '02* (pp. 142-151). ACM Press.

Parnas, D. L. (1969). On the use of transition diagrams in the design of a user interface for an interactive computer system. *Proceedings 24th National ACM Conference*, (pp. 379-385).

Paternò, F. (1999). *Model-Based Design and Evaluation of Interactive Application.* Berlin: Springer Verlag.

Paternò, F. (2004). ConcurTaskTree: An engineered notation for task models. In D. Diaper, & N. Stanton, *The handbook of task analysis for human-computer interaction.* Lawrence Erlbaum Associates.

Paternò, F. (2004). ConcurTaskTrees: An Engineered Notation for Task Models. In D. Diaper, & N. Stanton, *The Handbook of Task Analysis for Human Computer Interaction.* Lawrence Erlbaum Associates.

Paternò, F., & Santoro, C. (2000). Integrating model checking and HCI tools to help designers verify user interface properties. In P. Palanque, & Paternò (Éd.), *Proceedings of the 7th international conference on Design, specification, and verification of interactive systems (DSV-IS'00)* (pp. 135-150). Berlin, Heidelberg: Springer-Verlag.

Paterno, F., & Zini, E. (2004). Applying information visualization techniques to visual representations of task models. *TAMODIA.* New York, NY, USA: ACM.

Paternò, F., Breedvelt-Schouten, I. M., & de Koning, N. (1999). Deriving Presentations from Task Models. *Proceedings of the IFIP TC2/TC13 WG2.7/WG13.4 Seventh Working Conference on Engineering for Human-Computer Interaction*, (pp. 319-337).

Paternò, F., Mancini, C., & Meniconi, S. (1997). ConcurTaskTrees: A Diagrammatic Notation for Specifying Task Models. *INTERACT.* Sidney, Autralie: Chapman & Hall.

Patrick, J. (1992). *Training: Research and Practice.* San Diego, CA: Academic Press.

Petri, C. A. (1962). Kommunikation mit Automaten. *PhD Thesis*. Technical University Darmstadt.

Puerta, A., & Eisenstein, J. (1999). Towards a general computational framework for model-based interface development systems. *IUI.* Los Angeles, CA, USA.

Puerta, A., & Eisenstein, J. (2002). A common representation for interaction data. *In Proc. Of the 7th International Conference on Intelligent User Interfaces.* Santa Fe: ACM press.

Rajeev, A. (1999). Timed Automata. *11th International Conference on Computer Aided Verification.* London, UK: Springer-Verlag.

Ramamoorthy, C., & Ho, G. (1980). Performance Evaluation of Asynchronous Concurrent Systems. *IEEE Transactions of Software Engineering, 6*(5).

Rasmussen, J. (1983). Skills, rules, and knowledge; signals, signs, and symbols, and other distinctions in human performance models. *IEEE Transactions on Systems, Man and Cybernetics, SMC-13*(3), pp. 257-266.

Rasmussen, J. (1985). The role of hierarchical knowledge representation in decision making and system management. *IEEE Transactions on Systems, Man and Cybernetics, SMC-15*(2), pp. 234-243.

Rasmussen, J., & Vicente, K. (1989). Coping with human errors through system design: Implications for ecological interface design. *International Journal of Man-Machine Studies*(31, pp. 517-534).

Rational Software Corporation. (1997). *UML Notation Guide. 1.1.*

Rauterberg, M. (1992). An Iterative-Cyclic Software Process Model. *International Conference on Software Engineering and Knowledge Engineering.* Capri, Italie: IEEE.

Reason, J. (1990). *Human error.* Cambridge: Cambridge University Press.

Reason, J. T. (1990). *Human Error.* Cambridge University Press.

Reichart, D., Dittmar, A., Forbrig, P., & Wurdel, M. (2008). Tool Support for Representing Task Models, Dialog Models and User-Interface Specifications. *DSVIS. LNCS 5136.* Kingston, Canada: Springer.

Reiser, R. (1987). History. In R. Gagné, *Instructional Technology: Foundations.* Hillsdale, NJ: Lawrence Erlbaum.

Reiser, R. (2001). A History of Instructional Design and Technology: Part II: A History of Instructional Design In Educational Technology Research and Development. *49*(2).

Rollenhagen, C. (1995). *MTO – En Introduktion: Sambandet Människa, Teknik och Organisation.* Lund, Sweden: Studentlitteratur.

Rosson, M. B., & Carroll, J. M. (2001). *Usability engineering: scenario-based development of human-computer interaction.* Elsevier.

Royce, W. (1970). *Managing the development of large software systems: Concepts and techniques.* WESCON technical paper.

Royer, J. M., Cisero, C. A., & Carlo, M. S. (1993). Techniques and procedures for assessing cognitive. *Review of Educational Research Summer, 63*, 201-243.

Royer, J. M., Cisero, C. A., & Carlo, M. S. (1993). Techniques and procedures for assessing cognitive. *Review of Educational Research Summer* , 201-243 .

Rumelhart, D. E., Hinton, G. E., & McClelland, J. L. (1987). A general framework for parallel distributed processing. In D. E. Rumelhart, & J. L. McClelland, *Parallel distributed processing: Explorations in the microstructure of cognition.* MIT Press.

Salas, E., & Cannon-Bowers, J. (2001). The Science of Training: A Decade of Progress. *Annual Review of Psychology*, pp. 471-499.

Santoni, C., Vieira, M. F., Scaico, A., Vieira, F. A., & Pereira, M. R. (2004). Reviewing Human Errors in Industrial Incidents for Safety Design. *The society for Modelling and Simulation Intrenational. Proceedings of the 2004 Summer Computer Simulation Conference.* (pp. 69-75). San Diego, CA: Bruzzone, A.G.; Williams, E.;.

Santoro, C. (2005). Chapter 6 - Using Task Models to Verify Properties of an Interactive System. In C. Santoro, *A Task-Model Based Approach for the Design and Evaluation of Innovative User Interfaces.* Louvain-la-Neuve: Presses Universitaires de Louvain.

Save, L., & Feuerberg, B. (2012). Designing Human-Automation Interaction: a new level of Automation Taxonomy. In D. De Waard, K. Brookhuis, F. Dehais, C. Weikert, S. Röttger, D. Manzey, . . . P. Terrier (Éd.), *Human Factors of Systems and Technology (on the occasion of the Human Factors and Ergonomics Society Europe Chapter Annual Meeting).* Toulouse, France.

Sawyer, J., Minsk, B., & Bisantz, A. (1996). Coupling User Models and System Models: A Modeling Framework for Fault Diagnosis in Complex Systems. *Interacting with computer*.

Scapin, D. L. (2007). K-MADe. *COST294-MAUSE 3rd International Workshop, Review, Report and Refine Usability Evaluation Methods (R3 UEMs).* Athens.

Scapin, D., & Pierret-Golbreich, C. (1989). Towards a method for task description: MAD. In L. Berlinguet, & D. Berthelette (Éd.), *Work with Display Units WWU'89*, (pp. 27–34).

Schank, R. C., & Abelson, R. P. (1995). Knowledge and memory: The real story. In R. S. Wyer, *Advances in social cognition.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Schmidt, D. (2006). Model-Driven Engineering. *IEEE Computer, 39*(2).

Scott, D. (2009). Growing a Training System and Culture for the Ares I Upper Stage Project. *Aerospace Conference.* Big Sky, MT: IEEE.

Seamster, T., Boehm-Davis, D., Holt, R., & Schultz, K. (1998, August 1). *Developping Advanced Crew Resource Management (ACRM) Training: A Training Manual*. Washington DC, USA: Federal Aviation Administration.

Sebillotte, S. (1988). Hierarchical planning as method for task analysis: the example of office task analysis. *Behaviour & Information Technology, 7*(3).

Selby, R. (2009). Synthesis, Analysis, and Modeling of Large-Scale Mission-Critical Embedded Software Systems. *International Conference on Software Process: Trustworthy Software Development Process.* Vancouver: Springer-Verlag.

SESAR Joint Undertaking. (2013). *Guidance material for HP Automation Support. WP 16.05.01 Deliverable 04.* .

Shavelson, R. J., Ruiz-Primo, M. A., & Wiley, E. (2005). Windows into the mind. *Higher Education, 49*(4), 413-430.

Shavelson, R. J., Ruiz-Primo, M. A., & Wiley, E. (2005). Windows into the mind. *Higher Education, 49*(4), 413-430.

Shepherd, A. (1985). Hierarchical task analysis and training decisions. *Programmed Learning and Educational Technology*(22).

Sheridan, T. B., & Verplank, W. (1978). *Human and Computer Control of Undersea Teleoperators.* Cambridge, MA: Man-Machine Systems Laboratory, Department of Mechanical Engineering, MIT.

Sheridan, T. B., & Verplank, W. (1978). *Human and Computer Control of Undersea Teleoperators.* Cambridge, MA: Man-Machine Systems Laboratory, Department of Mechanical Engineering, MIT.

Shwaber, K., & Beedle, M. (2002). *Agile Software Development with SCRUM.* Upper Saddle River, NJ: Prentice-Hall.

Sinnig, D., Forbrig, P., Wurdel, M., Chalin, P., & Khendek, F. (2007). Practical extensions for Task models. *International Conference on TAsk MOdels and DIAgrams, TAMODIA* (pp. 42-55 (vol. 4849)). Heidelberg: Springer.

Sommerville, I. (2006). *Software engineering* (éd. 7th). Addison-Wesley.

Sowa, J. F. (1992). Semantic networks. In J. W. Sons (Éd.), *Encyclopedia of Artificial Intelligence.* New York: Shapiro, S. C.;.

Sowa, J. F. (1992). Semantic networks. In S. C. Shapiro, *Encyclopedia of Artificial Intelligence.* New York, NY, USA: John Wiley & Sons.

Sparx Systems. (2010). *Enterprise Architect.*

Stanton, N. A., Stewart, R., Harris, D., Houghton, R. J., Baber, C., McMaster, R., . . . Green, D. (2006). Distributed situation awareness in dynamic systems: theoretical development and application of an ergonomics methodology. *Ergonomics, 49*(12-13), pp. 1288-1311.

Sternberg, R. (2008). *Cognitive Psychology.* New York, NY: Cengage Learning.

Storey, N. (1996). *Safety-critical computer systems.* Addison-Wesley.

Stuart, J., & Penn, R. (2004). TaskArchitect: Taking the work out of task analysis. *Proceedings of the 3rd annual conference on Task models and diagrams. 86*, pp. 145-154. New York, NY, USA: ACM Press.

Sukaviriya, P., Kovacevic, S., Foley, J., Myers, B., Olsen Jr., D., & Schneider-Hufschmidt, M. (1994). Model-Based User Interfaces: What Are They and Why Should We Care? *ACM Symposium on User Interface Software and Technology* (pp. 133-135). Marina del Rey, CA, USA: ACM.

Svensson, O. (2001). Accident and Incident Analysis Based on the Accident Evolution and Barrier Function (AEB) Model. *Cognition, Technology & Work, 3*(1), 42-52.

Sweet, W. (1995). The glass cockpit [flight deck automation]. *Spectrum, IEEE, 32*(9), 30-38.

Tankeu-Choitat, A., Fabre, J.-C., Palanque, P., Navarre, D., Deleris, Y., & Fayollas, C. (2011). Self-Checking Components for Dependable Interactive Cockpits using Formal Description Techniques. *17th Pacific Rim Dependable Computing Conference (PRDC 2011).* Pasadena, US.

Tarby, J., & Barthet, M. (1996). The Diane+ method. *2nd International Workshop on Computer-Aided Design of User Interfaces.* Presses Universitaires de Namur, Vanderdonckt, J.

Tardieu, H., Rochfeld, A., & Colletti, R. (1986). *La méthode MERISE, principes et outils.* Edition d'Organisation.

Thimbleby, H. (2007). User-centered methods are insufficient for safety-critical systems. *Usability and HCI for medicine and healthcare. lncs 4799.* Graz: Springer.

Turban, E., & Aronson, J. (1988). *Decision Support Systems and Intelligent Systems.* NJ: Prentice Hall, Inc. Upper Saddle River.

U.S. Army Field Artillery School. (1984). *A System Approach To Training (Course Student textbook). ST - 5K061FD92 .*

U.S. Department of Defense. (1949). *Procedures for Performing a Failure Mode, Effects and Criticality Analysis.*

U.S. Department of Defense. (1980). *Procedures for Performing a Failure Mode, Effects and Criticaility Analysis.* Available: http://upload.wikimedia.org/wikipedia/commons/6/60/MIL-STD-1629RevA.pdf.

Van Der Veer, G. C., Lenting, V. F., & Bergevoet, B. A. (1996). Gta: Groupware task analysis - modeling complexity. *Acta Psychologica, 91*, 297-322.

van Welie, M., van der Veer, G. C., & Eliëns, A. (1998). Euterpe - Tool support for analyzing cooperative environments. *9th European Conference on Cognitive Ergonomics.* Limerick, Ireland.

Vanderdonckt, J., & Bodart, F. (1993). Encapsulating Knowledge for Intelligent Automatic Interaction Objects Selection. *INTERCHI '93*, (pp. 424-429). Amsterdam.

Vicente, K. (1999). *Cognitive work analysis: Toward safe, productive, and healthy computer-based work.* Mahwah, New Jersey: Lawrence Erlbaum Associates.

Villaren, T., Coppin, G., & Leal, A. (2012). Modeling task transitions to help designing for better situation awareness. In S. Diniz Junqueira Barbosa, J. Creissac Campos, R. Kazman, P. Palanque, M. Harrison, & S. Reeves (Éd.), *In Proceedings of the 4th ACM SIGCHI symposium on Engineering interactive computing systems (EICS '12)* (pp. 195-204). New York, NY, USA: ACM Press.

Wherry, E. (2003). Scroll ring performance evaluation. *International Conference on Human Factors in Computing Systems (CHI'03).* Fort Lauderdale: ACM.

Wickens, C., Mavor, A., Parasuraman, R., & McGee, J. (1998). *Commission on behavioural and social sciences and education in The Future of Air Traffic Control: Human operators and automation.* Washington DC: National Academy Press.

Wieting, R. (1996). Hybrid high-level nets. (J. M. Charnes, D. J. Morrice, & D. T. Brunne, Éds.) *Proceedings of the 1996 Winter Simulation Conference*, 848–855.

Willans, J., & Harrison, M. (2001). Prototyping preimplementation designs of virtual environment behaviour. *8th IFIP Working conference on engineering for human-computer interaction (EHCI'01).* Lecture notes in computer science.

Williams, C. G. (1998). Using Concept Maps to Assess Conceptual Knowledge of Function. *Journal for Research in Mathematics Education, 29*(4), 414-421.

Williams, C. G. (1998). Using Concept Maps to Assess Conceptual Knowledge of Function. *Journal for Research in Mathematics Education, 29*(4), 414-421.

Williams, K. (2005). Computer-Aided GOMS: A Description and Evaluation of a Tool That Integrates Existing Research for Modeling Human-Computer Interaction. *International Journal of Human-Computer Interaction, 18*(1), 39-58.

Wilson, P., Dell, L. D., & Anderson, G. F. (1993). *Root Cause Analysis: A Tool for Total Quality Management.* Milwaukee, WI: Quality Press.

Winckler, M., Vanderdonckt, J., Trindade, F., Stan, & Stanciulescu, A. (2008). Cascading Dialog Modeling with UsiXML. *DSVIS. LNCS 5136.* Kingston, Canada: Springer.

Wise, J. A., Hopkin, V. D., & Garland, D. J. (2009). *Handbook of aviation human factors.* CRC Press.

Wood, W. (1970). Transition network grammars for natural language analysis. *Communications of the ACM, 13*(10).

Young, R. M. (2001). Production Systems in Cognitive Psychology. In N. J. Smelser, & P. B. Baltes, *International Encyclopaedia of the Social and Behavioural.* Oxford: Elsevier Science.

Abstract

The current European Air Traffic Management (ATM) System needs to be improved for coping with the growth in air traffic forecasted for next years. It has been broadly recognised that the future ATM capacity and safety objectives can only be achieved by an intense enhancement of integrated automation support. However, increase of automation might come along with an increase of performance variability of the whole ATM System especially in case of automation degradation.

ATM systems are considered complex as they encompass interactions involving humans and machines deeply influenced by environmental aspects (i.e. weather, organizational structure) making them belong to the class of Socio-Technical Systems (STS) (Emery & Trist, 1960). Due to this complexity, the interactions between the STS elements (human, system and organisational) can be partly linear and partly non-linear making its performance evolution complex and hardly predictable. Within such STS, interactive systems have to be usable i.e. enabling users to perform their tasks efficiently and effectively while ensuring a certain level of operator satisfaction. Besides, the STS has to be resilient to adverse events including potential automation degradation issues but also interaction problems between their interactive systems and the operators. These issues may affect several STS aspects such as resources, time in tasks performance, ability to adjust to environment, etc. In order to be able to analyse the impact of these perturbations and to assess the potential performance variability of a STS, dedicated techniques and methods are required. These techniques and methods have to provide support for modelling and analysing in a systematic way usability and resilience of interactive systems featuring partly autonomous behaviours. They also have to provide support for describing and structuring a large amount of information and to be able to address the variability of each of STS elements as well as the variability related to their interrelations. Current techniques, methods and processes do not enable to model a STS as a whole and to analyse both usability and resilience properties. Also, they do not embed all the elements that are required to describe and analyse each part of the STS (such as knowledge of different types which is needed by a user for accomplishing tasks or for interacting with dedicated technologies). Lastly, they do not provide means for analysing task migrations when a new technology is introduced or for analysing performance variability in case of degradation of the newly introduced automation. Such statements are argued in this thesis by a detailed analysis of existing modelling techniques and associated methods highlighting their advantages and limitations.

This thesis proposes a multi-models based approach for the modelling and the analysis of partly-autonomous interactive systems for assessing their resilience and usability. The contribution is based on the identification of a set of requirements needed being able to model and analyse each of the STS elements. Some of these requirements were met by existing modelling techniques, others were reachable by extending and refining existing ones. This thesis proposes an approach which integrates 3 modelling techniques: FRAM (focused on organisational functions), HAMSTERS (centred on human goals and activities) and ICO (dedicated to the modelling of interactive systems).

The principles of the multi-models approach is illustrated on an example for carefully showing the extensions proposed to the selected modelling techniques and how they integrate together. A more complex case study from the ATM domain is then presented to demonstrate the scalability of the approach. This case study, dealing with aircraft route change due to bad weather conditions, highlights the ability of the integration of models to cope with performance variability of the various parts of the STS.

Keywords: multi-models approach, ATM, usability, resilience, automation, interactive systems

Résumé

La croissance prévisionnelle du trafic aérien est telle que les moyens de gestion actuels doivent évoluer et être améliorés. Les objectifs de capacité et sécurité pourront être atteints en augmentant le niveau d'automatisation des systèmes. Cependant, cette augmentation pourrait entraîner un accroissement de la variabilité de la performance de l'ensemble des moyens de gestion du trafic aérien, en particulier en cas de dégradation de l'automatisation.

Les systèmes de gestion du trafic aérien sont considérés comme complexes car ils impliquent de nombreuses interactions entre humains et systèmes, et peuvent être profondément influencés par les aspects environnementaux (météorologie, organisation…). Ces systèmes appartiennent à la catégorie des Systèmes Sociotechniques (STS) (Emery & Trist, 1960). A cause de leur complexité, les interactions entre les différents éléments (humains, systèmes et organisations) de ces STS peuvent être linéaires et partiellement non linéaires, ce qui rend l'évolution de leur performance difficilement prévisible. Au sein de ces STS, les systèmes interactifs doivent être utilisables, i.e. permettre à leurs utilisateurs d'accomplir leurs tâches de manière efficace et efficiente. Un STS doit aussi être résilient aux événements adverses tels que les potentielles dégradations de l'automatisation ou les problèmes d'interaction entre les systèmes et leurs opérateurs. Ces problèmes peuvent affecter plusieurs aspects des systèmes sociotechniques comme les ressources, le temps d'exécution d'une tâche, la capacité à d'adaptation à l'environnement… Afin de pouvoir analyser l'impact de ces perturbations et d'évaluer la variabilité de la performance d'un STS, des techniques et méthodes dédiées sont requises. Elles doivent fournir un support à la modélisation et à l'analyse systématique de l'utilisabilité et de la résilience de systèmes interactifs aux comportements partiellement autonomes. Elles doivent aussi permettre de décrire et de structurer un grand nombre d'informations, ainsi que de traiter la variabilité de chaque élément du STS et la variabilité liée à leurs interrelations. Les techniques et méthodes existantes ne permettent actuellement ni de modéliser un STS dans son ensemble, ni d'en analyser les propriétés d'utilisabilité et de résilience. De plus, elles ne contiennent pas tous les éléments requis pour décrire et analyser chaque partie d'un STS (comme par exemple les différents types de connaissances nécessaires à un utilisateur pour accomplir une tâche). Enfin, elles ne fournissent pas les moyens d'analyser la migration de tâches suite à l'introduction d'une nouvelle technologie ou d'analyser la variabilité de la performance en cas de dégradation de fonctions récemment automatisées. Ces arguments sont développés dans la thèse et appuyés par une analyse détaillée des techniques de modélisation existantes et des méthodes qui leurs sont associées.

La contribution présentée est basée sur l'identification d'un ensemble d'exigences requises pour pouvoir modéliser et analyser chacun des éléments d'un STS. Certaines de ces exigences ont été remplies grâce à l'utilisation de techniques de modélisation existantes, les autres grâce à l'extension et au raffinement d'autres techniques. Cette thèse propose une approche qui intègre 3 techniques en particulier : FRAM (centrée sur les fonctions organisationnelles), HAMSTERS (centrée les objectifs et activités humaines) et ICO (dédiée à la modélisation du comportement des systèmes interactifs).

Cette approche est illustrée par un exemple mettant en œuvre les extensions proposées et l'intégration des modèles. Une étude de cas plus complexe sur la gestion du trafic aérien (changement de route d'un avion en cas de mauvaises conditions météorologiques) est ensuite présentée pour montrer le passage à l'échelle de l'approche. Elle met en avant les bénéfices de l'intégration des modèles pour la prise en compte la variabilité de la performance des différents éléments d'un STS.

Mots-clés: approche basée modèles, gestion du trafic aérien, automatisation, systèmes interactifs

Annex

The Annex describes the multi-models based approach applied to the first scenario idendified for the WXR case study. The main phases, sub-phases and related outputs are briefly reported.

The Definition phase applied to the WXR scenario

Extract from Scenario:

> *"While the aircraft is flying, the pilot continuously checks weather conditions in order to detect as soon as possible perturbations. For accomplishing this task, the pilot verifies through the mode selection functionality of weather radar the information about the status of the radar for being sure that the weather radar has been set up correctly and it is properly working. Then, through the second functionality which is dedicated to the adjustment of the weather radar orientation, the pilot checks if there are weather perturbations on the flight path. Turning the knob, s/he can set up different range scale. After having setup the range to 160 NM, s/he sees on the display that there is a perturbation. The pilot recognises a storm cloud and remembers that the planned procedure is to change the current route of the aircraft to avoid passing in this storm cloud. Then, the pilot decides to request a new heading for avoiding the weather perturbation. S/he contacts the executive air traffic controller (EXC) in charge of the sector, and requests to turn left heading 320. When the EXC receives the request, s/he deals with it in order to safely manage the current air traffic flow and coordinate with the pilot the manoeuvre. The EXC analyses that the requested heading clearance is compliant with the trajectories of all the other aircraft of her/his area. The EXC then gives her/his acknowledgement through the clearance turn left heading 320, and the pilot can execute it. The pilot enters the new heading through the FCU and the weather perturbation is avoided."*

Perfomance objectives:

> *"With regard to usability:*
>
> - *The maximum time to configure the WXR interactive application for displaying weather conditions on the current route shall not exceed 10 seconds.*
>
> *With regard to resilience:*
>
> - *The pilot should be able to find an appropriate new route and to get approval to implement this new route whatever the surrounding traffic and weather conditions.*

The Modelling phase applied to the WXR scenario

The Identification/Description/Modelling sub-phase applied to the WXR scenario

The main identified functions are:

- Aircraft flies
- Check weather conditions
    - Scan airspace
    - Rendering of radar information

- Manage weather radar status
  - Manage mode
  - Manage tilt angle
    - Edit angle
    - Display updated value
    - Change tilt angle
- Analyse current weather map
- Select different range map
- Detect weather target
- Recognize weather target

- Identify the adequate procedure
- Request a clearance
- Coordinate with EXC
- Execute the clearance

These functions can be represented at higher levels of abstraction or with greater detail as required by the analyst according to her/his interest. For example, "Check weather conditions" function is an abstraction of several lower-level functions such as "Read information" and "Select range map" and so on. This "macro" function includes system, human and interactive functions. Such an abstraction provides support for the representation of a larger number of functions while keeping the graphical model representation understandable.

Once identified the functions, these have to be described according to their type (human, system and interactive). The following table reports the type of the listed functions.

TABLE 93: SCENARIO 1 – FUNCTIONS TYPE DESCRIPTION – WXR CASE STUDY

Function	**Type of function**		
	Human	*System*	*Interactive*
Aircraft flies			x
Check weather conditions			
Rendering of radar information		x	
Scan airspace		x	
Manage weather radar status			x
Manage modes			x
Manage tilt angle			x
Edit tilt angle	x		
Display updated value		x	
Change tilt angle			x
Analyse current weather map	x		
Select different range map			x
Detect weather target		x	
Recognize weather target	x		
Identify the adequate procedure	x		
Request a clearance	x		
Coordinate with EXC	x		
Execute the clearance			x

Then, according to the scenario under analysis, we can build a "Timeline" which represents a sequence of related events arranged in chronological order and displayed along a line. As illustrated in Figure 87, at time 0 (T0) the aircraft is flying. In the meantime, the pilot checks the weather conditions (T1). In performing this task, s/he identifies a perturbation (a weather target) and needs to apply the adequate procedure (T2). So, the pilot requests a clearance to the EXC (T3). This means that the pilot has to make coordination with the EXC who is in charge for the sector (T4). Once the EXC provides the confirmation for the clearance, this can be executed (T5) and, then, implemented as change heading in the Flight Management System of the aircraft.

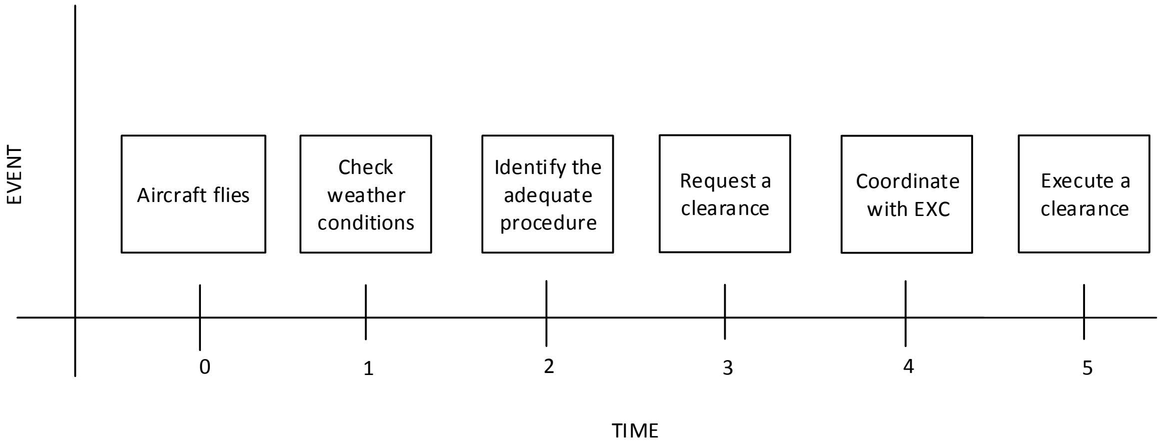


Figure 87: Scenario 1 – the WXR case study timeline

Once the functions have been listed and the timeline has been described, it is needed to "Characterise potential variability through FRAM" (as described in Chapter 3 – Section 2.2). The following steps are performed on the WXR case study functions:

- All the functions are described as FRAM units with detailing the six basic aspects (Input, Output, Pre-conditions, Resources, Time, and Control) (please refer to Chapter 3 – Section 2.2);
- The endogenous factors are listed. Endogenous source of variability is related to the internal variability of the PAIS within STS (human, system or their interactions) that performs the function (please refer to Chapter 3 – Section 2.2);
- The potential impact of endogenous factors on the functions output variability is described. Variability of output can be described with a set of dimensions such as timing and precision (please refer to Chapter 3 – Section 2.2).

The produced outputs (FRAM units, endogenous factors table and output variability tables) are needed to create a "FRAM instantiation" (please refer to Chapter 5 – Section 1.2.1).

For the WXR case study, the following figure represents the function "Aircraft flies" as FRAM unit.

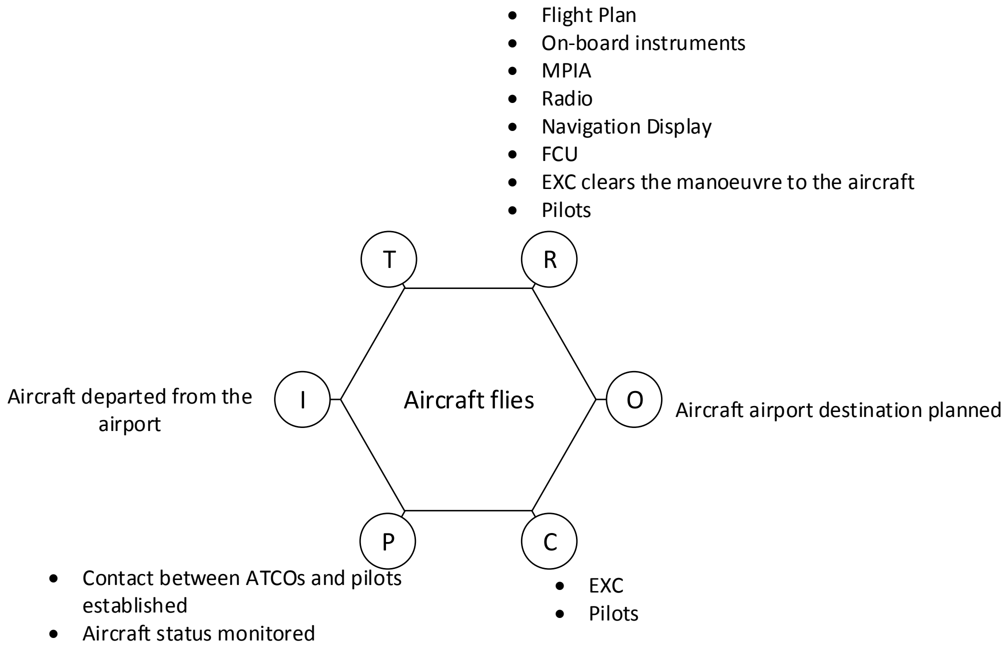


Figure 88: "Aircraft flies" function as FRAM unit

Moreover, for each FRAM unit, we describe the six aspects through a dedicated table. Some aspects may be marked as N/A ("Not Applicable") because of they are not relevant for the specific function. Indeed, the following table reports the explanation of Input, Output, Precondition, Resource, Control, and Time aspects.

TABLE 94: "AIRCRAFT FLIES" FUNCTION FRAM TABLE

AIRCRAFT FLIES	
Input(s)	• Aircraft departed from the airport
Output(s)	• Aircraft airport destination planned
Precondition	• Contact between ATCOs and Flight Crew established • Aircraft status monitored
Resource(s)	• Flight Plan • On-board instruments • MPIA • Radio • Navigation Display • FCU • EXC clears the manoeuvre to the aircraft • Pilots
Control(s)	• EXC • Pilots
Time	Time related to speed and airports distance

Then, it is needed to identify all the endogenous factors. As described in Chapter 3 – Section 2.2, this description is supported by a table (Table 95 in the current case study) in which are listed the endogenous factors that can affect the variability of the described FRAM function. In the WXR case study, we can have also exogenous factors (e.g. the number of aircraft in the same airspace and the weather conditions) that can affect the variability of the described FRAM function (these are reported in Table 96). The impact of both these factors can be positive, neutral or negative represented by a green, grey or red arrow respectively.

TABLE 95: ENDOGENOUS FACTORS OF VARIABILITY RELATED TO "AIRCRAFT FLIES" FUNCTION

Experience	High (years …) = ↑	Medium (years …) = ↔	N/A (people have been trained and qualified)
Focus of attention	High (level …) = ↑	Medium (level …) = ↔	Low (level …) = ↓
Situation Awareness	High (level …) = ↑	Medium (level …) = ↔	Low (level …) = ↓
Workload	High (level …) = ↓	Medium (level …) = ↔	Low (level …) = ↑
Stress	High (level …) = ↓	Medium (level …) = ↔	Low (level …) = ↑
Complacency	High (level …) = ↓	Medium (level …) = ↔	Low (level …) = ↑
Fatigue	High (level …) = ↓	Medium (level …) = ↔	Low (level …) = ↑
System performance	High (level) = ↑	Medium (level …) = ↔	Low (level …) = ↓
Status of the system	High (level …) = ↑	Medium (level …) = ↔	Low (level …) = ↓
Knowledge	High (No. of procedures/strategies) = ↑	Medium (No. of procedures) = ↔	N/A

TABLE 96: EXOGENOUS FACTORS OF VARIABILITY RELATED TO "AIRCRAFT FLIES" FUNCTION

N° Aircraft	High (n …) = ↓	Medium (n …) = ↔	Low (n …) = ↑
Weather perturbations	High (level …) = ↓	Medium (level …) = ↔	Low (level …) = ↑

The endogenous and exogenous factors can have an impact on the output variability in terms of time and precision as reported in Table 97. With regard to time, we can have 3 possible conditions that are "Too early", "On time" and "Too late". With regard to precision, we can have 3 possible conditions that are "Imprecise", "Acceptable" and "Precise". The impact of each condition is explained in the table and it can affect a linked downstream function with different severity degree.

TABLE 97: OUTPUT VARIABILITY OF "AIRCRAFT FLIES" FUNCTION

Aircraft airport destination planned		
Time	Too early	The aircraft airport destination is planned well in advance with respect to the regulation.
	On time	The aircraft airport destination is planned on time with respect to the regulation.
	Too late	N/A
Precision	Imprecise	N/A
	Acceptable	N/A
	Precise	The aircraft airport destination is planned and well defined.

These sub-steps have to be accomplished for each function. In this case study the next function is "Check weather conditions" that is a "macro" function. As explained before, this means that it can be decomposed into several lower-level functions as "Scan airspace". This is represented as a FRAM unit in the following figure.

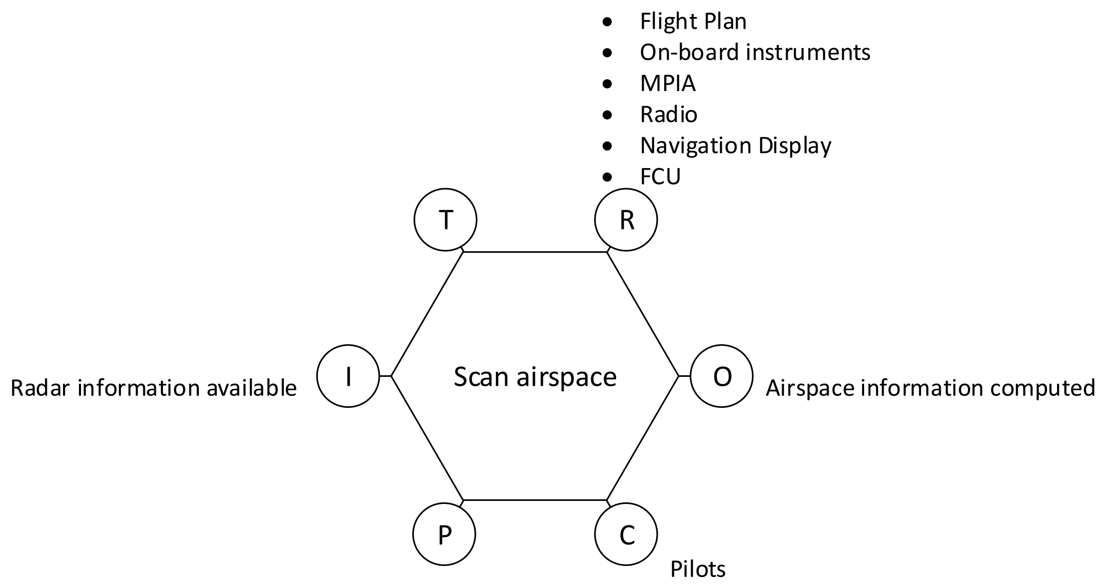


Figure 89: "Scan airspace" function as FRAM unit

The following table reports the explanation of Input, Output, Precondition, Resource, Control, and Time aspects.

TABLE 98: "SCAN AIRSPACE" FUNCTION FRAM TABLE

SCAN AIRSPACE	
Input(s)	• Radar information available
Output(s)	• Airspace information computed
Precondition	N/A
Resource(s)	• Flight plan • On-board instruments • MPIA • Radio • Navigation Display • FCU
Control(s)	• Pilots
Time	*Continuous Function – in parallel with the "Rendering of radar information"*

Table 99 and Table 100 report respectively the endogenous and the exogenous factors that can affect the variability of the described function. The impact of these factors can be positive, neutral or negative represented by a green, grey or red arrow respectively.

TABLE 99: ENDOGENOUS FACTORS OF VARIABILITY RELATED TO "SCAN AIRSPACE" FUNCTION

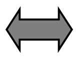
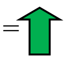
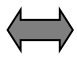
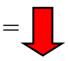
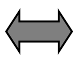
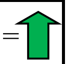
System performance	High (level) = 	Medium (level …) = 	Low (level …) = 
Status of the system	High (level …) = 	Medium (level …) = 	Low (level …) = 

TABLE 100: EXOGENOUS FACTORS OF VARIABILITY RELATED TO "SCAN AIRSPACE" FUNCTION

Weather perturbations	High (level …) = 	Medium (level …) = 	Low (level …) = 

These factors can impact on the output variability in terms of time and precision as reported in the following table.

TABLE 101: OUTPUT VARIABILITY OF "SCAN AIRSPACE" FUNCTION

Airspace information computed		
Time	Too early	N/A
	On time	The airspace information are successfully computed
	Too late	The airspace information are not computed in time
Precision	Imprecise	The airspace information are not correctly computed
	Acceptable	The airspace information are correctly computed
	Precise	The airspace information are correctly computed

The third identified function is "Rendering of radar information". This is represented as a FRAM unit in the following figure.

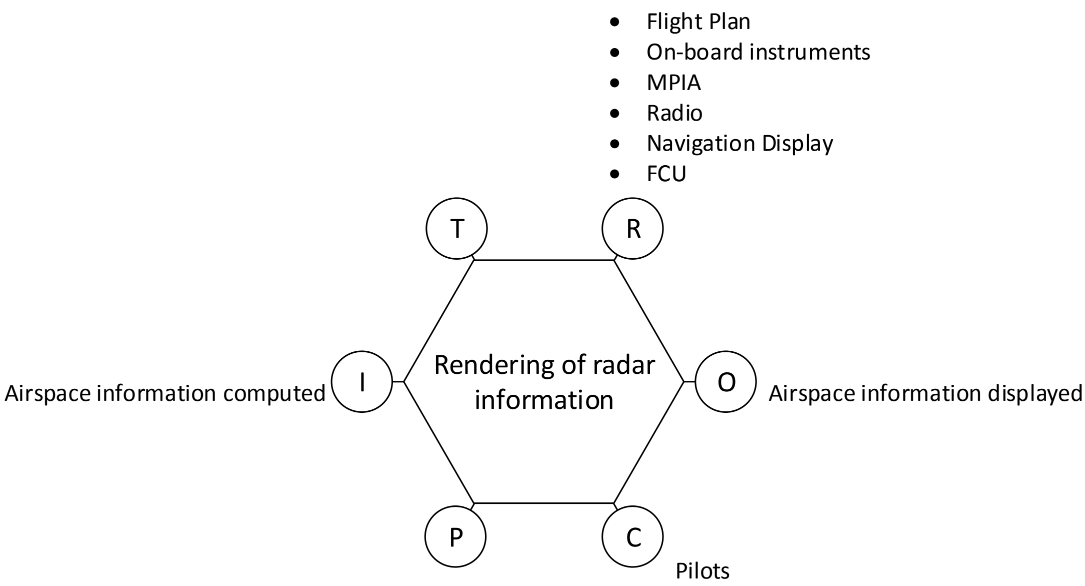


Figure 90: "Rendering of radar information" function as FRAM unit

The following table reports the explanation of Input, Output, Precondition, Resource, Control, and Time aspects.

TABLE 102: "RENDERING OF RADAR INFORMATION" FUNCTION FRAM TABLE

RENDERING OF RADAR INFORMATION	
Input(s)	• Airspace information computed
Output(s)	• Airspace information displayed
Precondition	N/A
Resource(s)	• Flight plan • On-board instruments • MPIA • Radio • Navigation Display • FCU
Control(s)	• Pilots
Time	*Continuous Function – in parallel with the "Scan airspace" function*

Table 103 and Table 104 report respectively the endogenous and the exogenous factors that can affect the variability of the described function. The impact of these factors can be positive, neutral or negative represented by a green, grey or red arrow respectively.

TABLE 103: ENDOGENOUS FACTORS OF VARIABILITY RELATED TO "RENDERING OF RADAR INFORMATION" FUNCTION

System performance	High (level) = ⇧	Medium (level …) = ⇔	Low (level …) = ⇩
Status of the system	High (level …) = ⇧	Medium (level …) = ⇔	Low (level …) = ⇩

TABLE 104: EXOGENOUS FACTORS OF VARIABILITY RELATED TO "RENDERING OF RADAR INFORMATION" FUNCTION

Weather perturbations	High (level …) = ⇩	Medium (level …) = ⇔	Low (level …) = ⇧

These factors can impact on the output variability in terms of time and precision as reported in the following table.

TABLE 105: OUTPUT VARIABILITY OF "RENDERING OF RADAR INFORMATION" FUNCTION

Airspace information displayed		
Time	Too early	N/A
	On time	The airspace information are successfully displayed
	Too late	The airspace information are not displayed in time
Precision	Imprecise	The airspace information are not correctly displayed
	Acceptable	The airspace information are correctly displayed
	Precise	The airspace information are correctly displayed

The following identified function is “Manage weather radar status” which is also a macro function consisting of several lower-level functions that are “Manage modes” and “Manage tilt angle”. The following figure represents “Manage modes” as FRAM unit.

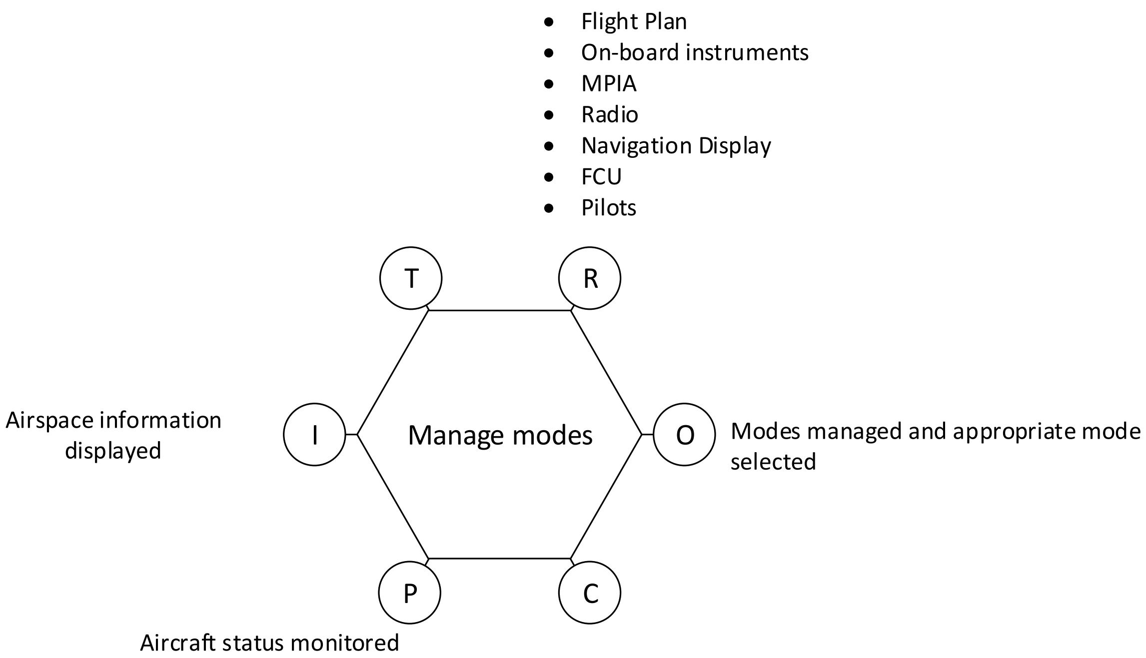


Figure 91: “Manage modes” function as FRAM unit

The following table reports the explanation of Input, Output, Precondition, Resource, Control, and Time aspects.

TABLE 106: "MANAGE MODES" FUNCTION FRAM TABLE

MANAGE MODES	
Input(s)	• Airspace information displayed
Output(s)	• Modes managed and appropriate mode selected
Precondition	• Aircraft status monitored
Resource(s)	• Flight plan • On-board instruments • MPIA • Radio • Navigation Display • FCU • Pilots
Control(s)	N/A
Time	*Repeated Function – in parallel with "Manage tilt angle"*

Table 107 and Table 108 report respectively the endogenous and the exogenous factors that can affect the variability of the described function. The impact of these factors can be positive, neutral or negative represented by a green, grey or red arrow respectively.

TABLE 107: ENDOGENOUS FACTORS OF VARIABILITY RELATED TO "MANAGE MODES" FUNCTION

Experience	High (years …) = ↑	Medium (years …) = ⟺	N/A (people have been trained and qualified)
Focus of attention	High (level …) = ↑	Medium (level …) = ⟺	Low (level …) = ↓
Situation Awareness	High (level …) = ↑	Medium (level …) = ⟺	Low (level …) = ↓
Workload	High (level …) = ↓	Medium (level …) = ⟺	Low (level …) = ↑
Stress	High (level …) = ↓	Medium (level …) = ⟺	Low (level …) = ↑
Complacency	High (level …) = ↓	Medium (level …) = ⟺	Low (level …) = ↑
Fatigue	High (level …) = ↓	Medium (level …) = ⟺	Low (level …) = ↑
System performance	High (level) = ↑	Medium (level …) = ⟺	Low (level …) = ↓
Status of the system	High (level …) = ↑	Medium (level …) = ⟺	Low (level …) = ↓
Knowledge	High (No. of procedures/strategies) = ↑	Medium (No. of procedures) = ⟺	N/A

TABLE 108: EXOGENOUS FACTORS OF VARIABILITY RELATED TO "MANAGE MODES" FUNCTION

Weather perturbations	High (level …) = ↓	Medium (level …) = ⟺	Low (level …) = ↑

These factors can impact on the output variability in terms of time and precision as reported in the following table.

TABLE 109: OUTPUT VARIABILITY OF "MANAGE MODES" FUNCTION

Modes managed and appropriate mode selected		
Time	Too early	The pilot manages the modes too early and the appropriate one is not successfully selected.
	On time	The pilot manages the modes on time. The appropriate one successfully selected
	Too late	The pilot manages the modes too late. The appropriate one is not successfully selected.
Precision	Imprecise	The pilot manages the modes partially or doesn't pay appropriate attention in managing them. The selected mode is not the appropriate one.
	Acceptable	The pilot manages the modes in an acceptable manner. S/he selects the appropriate one
	Precise	The pilot precisely manages the modes selecting the appropriate one.

The following identified function is "Manage tilt angle" which is itself a macro function consisting of several lower-level functions that are "Edit angle", "Display updated value", and "Change tilt angle". The following figure represents "Edit angle" as FRAM unit.

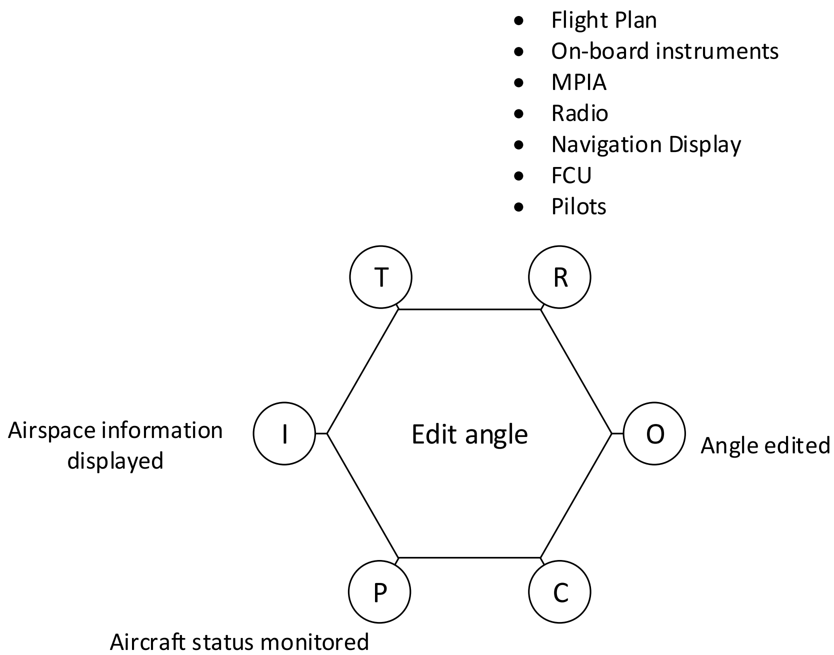


Figure 92: "Edit angle" function as FRAM unit

The following table reports the explanation of Input, Output, Precondition, Resource, Control, and Time aspects.

TABLE 110: "EDIT ANGLE" FUNCTION FRAM TABLE

EDIT ANGLE	
Input(s)	• Airspace information displayed
Output(s)	• Angle edited
Precondition	• Aircraft status monitored
Resource(s)	• Flight plan • On-board instruments • MPIA • Radio • Navigation Display • FCU • Pilots
Control(s)	N/A
Time	*On demand*

Table 111 and Table 112 report respectively the endogenous and the exogenous factors that can affect the variability of the described function. The impact of these factors can be positive, neutral or negative represented by a green, grey or red arrow respectively.

TABLE 111: ENDOGENOUS FACTORS OF VARIABILITY RELATED TO "EDIT ANGLE" FUNCTION

Experience	High (years …) =	Medium (years …) =	N/A (people have been trained and qualified)
Focus of attention	High (level …) =	Medium (level …) =	Low (level …) =
Situation Awareness	High (level …) =	Medium (level …) =	Low (level …) =
Workload	High (level …) =	Medium (level …) =	Low (level …) =
Stress	High (level …) =	Medium (level …) =	Low (level …) =
Complacency	High (level …) =	Medium (level …) =	Low (level …) =
Fatigue	High (level …) =	Medium (level …) =	Low (level …) =
Knowledge	High (No. of procedures/strategies) =	Medium (No. of procedures) =	N/A

TABLE 112: EXOGENOUS FACTORS OF VARIABILITY RELATED TO "EDIT ANGLE" FUNCTION

Weather perturbations	High (level …) = ⬇	Medium (level …) = ⬌	Low (level …) = ⬆

These factors can impact on the output variability in terms of time and precision as reported in the following table.

TABLE 113: OUTPUT VARIABILITY OF "EDIT ANGLE" FUNCTION

Angle edited		
Time	Too early	N/A
	On time	The pilot edits the angle on time.
	Too late	The pilot edits the angle too late. The value cannot be elaborated by the WXR application.
Precision	Imprecise	The pilot edits the angle partially or doesn't pay appropriate attention in editing the angle. The edited angle is not the appropriate one.
	Acceptable	The pilot edits the angle in an acceptable manner. S/he selects the appropriate one
	Precise	The pilot precisely edits the angle selecting the appropriate one.

The following identified function is "Display updated value" which is represented as FRAM unit in the figure below.

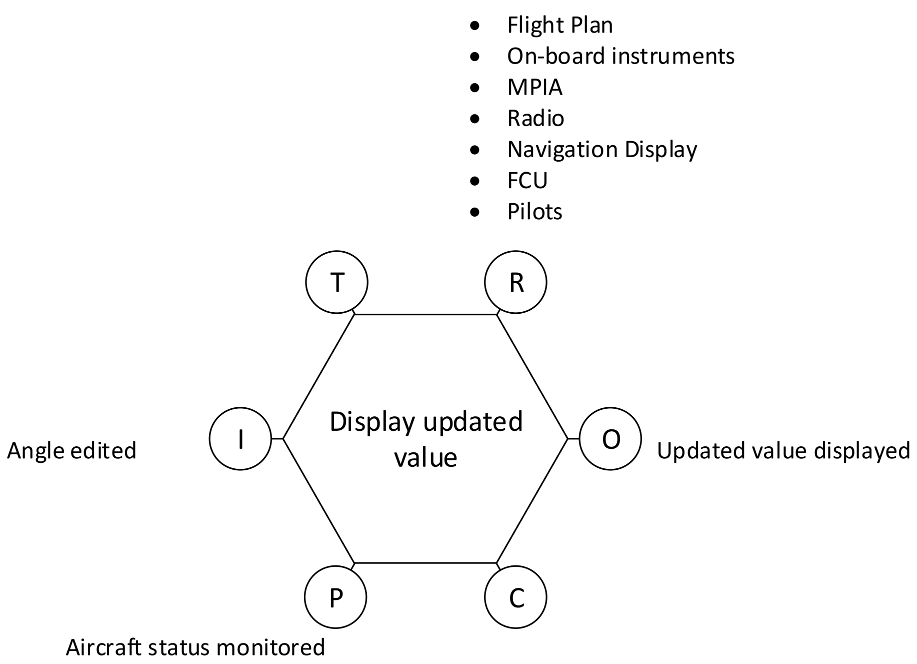


Figure 93: "Display updated value" function as FRAM unit

The following table reports the explanation of Input, Output, Precondition, Resource, Control, and Time aspects.

TABLE 114: "DISPLAY UPDATED VALUE" FUNCTION FRAM TABLE

DISPLAY UPDATED VALUE	
Input(s)	• Angle edited
Output(s)	• Updated value displayed
Precondition	• Aircraft status monitored
Resource(s)	• Flight plan • On-board instruments • MPIA • Radio • Navigation Display • FCU • Pilots
Control(s)	N/A
Time	*On demand*

Table 115 and Table 116 report respectively the endogenous and the exogenous factors that can affect the variability of the described function. The impact of these factors can be positive, neutral or negative represented by a green, grey or red arrow respectively.

TABLE 115: ENDOGENOUS FACTORS OF VARIABILITY RELATED TO "DISPLAY UPDATED VALUE" FUNCTION

System performance	High (level) = ↑	Medium (level …) = ↔	Low (level …) = ↓
Status of the system	High (level …) = ↑	Medium (level …) = ↔	Low (level …) = ↓

TABLE 116: EXOGENOUS FACTORS OF VARIABILITY RELATED TO "DISPLAY UPDATED VALUE" FUNCTION

Weather perturbations	High (level …) = ↓	Medium (level …) = ↔	Low (level …) = ↑

These factors can impact on the output variability in terms of time and precision as reported in the following table.

TABLE 117: OUTPUT VARIABILITY OF "DISPLAY UPDATED VALUE" FUNCTION

Updated value displayed		
Time	Too early	N/A
	On time	The displayed value is successfully updated
	Too late	The displayed value is not updated in time
Precision	Imprecise	The displayed value is not correctly updated
	Acceptable	The displayed value is correctly updated
	Precise	The displayed value is correctly updated

The following identified function is "Change tilt angle" which is represented in the figure below as FRAM unit.

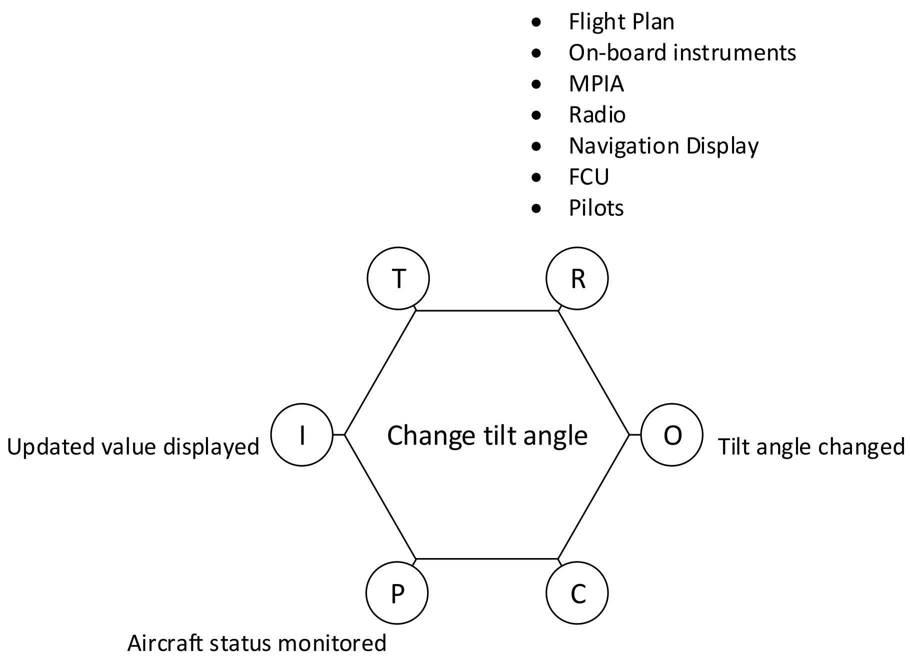


Figure 94: "Change tilt angle" function as FRAM unit

The following table reports the explanation of Input, Output, Precondition, Resource, Control, and Time aspects.

TABLE 118: "CHANGE TILT ANGLE" FUNCTION FRAM TABLE

CHANGE TILT ANGLE	
Input(s)	• Updated value displayed
Output(s)	• Tilt angle changed
Precondition	• Aircraft status monitored
Resource(s)	• Flight plan • On-board instruments • MPIA • Radio • Navigation Display • FCU • Pilots
Control(s)	N/A
Time	*On demand*

Table 119 andTable 120 report respectively the endogenous and the exogenous factors that can affect the variability of the described function. The impact of these factors can be positive, neutral or negative represented by a green, grey or red arrow respectively.

TABLE 119: ENDOGENOUS FACTORS OF VARIABILITY RELATED TO "CHANGE TILT ANGLE" FUNCTION

Experience	High (years …) = ↑	Medium (years …) = ↔	N/A (people have been trained and qualified)
Focus of attention	High (level …) = ↑	Medium (level …) = ↔	Low (level …) = ↓
Situation Awareness	High (level …) = ↑	Medium (level …) = ↔	Low (level …) = ↓
Workload	High (level …) = ↓	Medium (level …) = ↔	Low (level …) = ↑
Stress	High (level …) = ↓	Medium (level …) = ↔	Low (level …) = ↑
Complacency	High (level …) = ↓	Medium (level …) = ↔	Low (level …) = ↑
Fatigue	High (level …) = ↓	Medium (level …) = ↔	Low (level …) = ↑
System performance	High (level) = ↑	Medium (level …) = ↔	Low (level …) = ↓
Status of the system	High (level …) = ↑	Medium (level …) = ↔	Low (level …) = ↓
Knowledge	High (No. of procedures/strategies) = ↑	Medium (No. of procedures) = ↔	N/A

TABLE 120: EXOGENOUS FACTORS OF VARIABILITY RELATED TO "CHANGE TILT ANGLE" FUNCTION

Weather perturbations	High (level …) = ↓	Medium (level …) = ↔	Low (level …) = ↑

These factors can impact on the output variability in terms of time and precision as reported in the following table.

TABLE 121: OUTPUT VARIABILITY OF "CHANGE TILT ANGLE" FUNCTION

Tilt angle changed		
Time	Too early	N/A
	On time	The pilot changes the angle on time.
	Too late	The pilot changes the angle too late.
Precision	Imprecise	The pilot doesn't pay appropriate attention in changing the angle. The changed one is not the right.
	Acceptable	The pilot changes the angle in an acceptable manner. S/he selects the right one
	Precise	The pilot precisely changes the angle selecting the right one.

The following figure represents the "Analyse current weather map" function as FRAM unit.

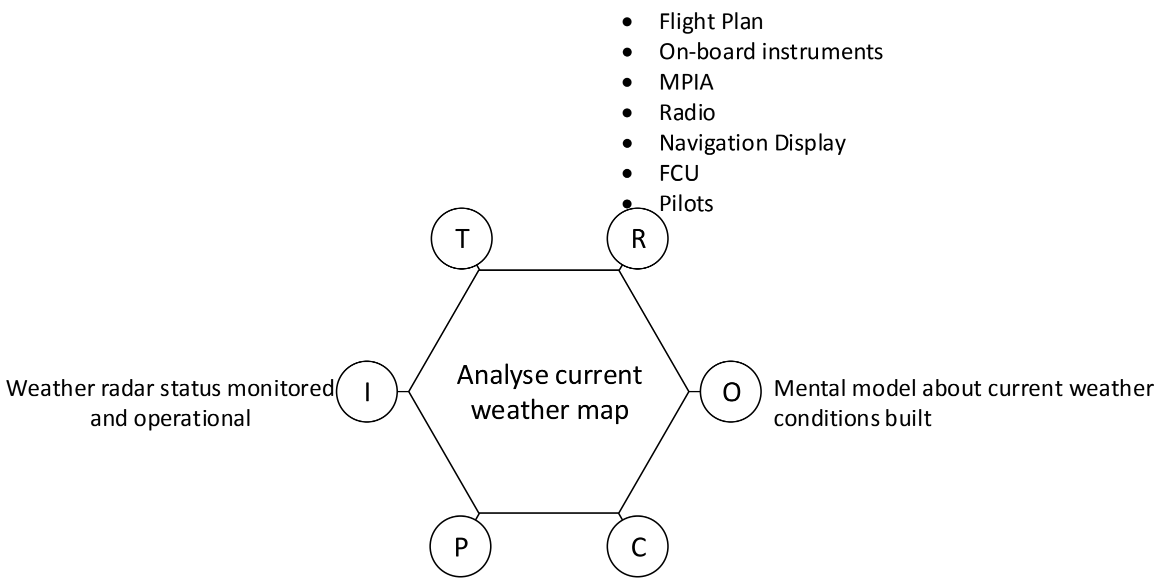


Figure 95: "Analyse current weather map" function as FRAM unit

The following table reports the explanation of Input, Output, Precondition, Resource, Control, and Time aspects.

TABLE 122: "ANALYSE CURRENT WEATHER MAP" FUNCTION FRAM TABLE

ANALYSE CURRENT WEATHER MAP	
Input(s)	• Weather radar status monitored and operational
Output(s)	• Mental model about current weather conditions built
Precondition	N/A
Resource(s)	• Flight plan • On-board instruments • MPIA • Radio • Navigation Display • FCU • Pilots
Control(s)	N/A
Time	*Repeated Function – competing with other functions*

Table 123 and Table 124 report respectively the endogenous and the exogenous factors that can affect the variability of the described function. The impact of these factors can be positive, neutral or negative represented by a green, grey or red arrow respectively.

TABLE 123: ENDOGENOUS FACTORS OF VARIABILITY RELATED TO "ANALYSE CURRENT WEATHER MAP" FUNCTION

Experience	High (years …) =	Medium (years …) =	N/A (people have been trained and qualified)
Focus of attention	High (level …) =	Medium (level …) =	Low (level …) =
Situation Awareness	High (level …) =	Medium (level …) =	Low (level …) =
Workload	High (level …) =	Medium (level …) =	Low (level …) =
Stress	High (level …) =	Medium (level …) =	Low (level …) =
Complacency	High (level …) =	Medium (level …) =	Low (level …) =
Fatigue	High (level …) =	Medium (level …) =	Low (level …) =
Knowledge	High (No. of procedures/strategies) =	Medium (No. of procedures) =	N/A

TABLE 124: EXOGENOUS FACTORS OF VARIABILITY RELATED TO "ANALYSE CURRENT WEATHER MAP" FUNCTION

Weather perturbations	High (level …) =	Medium (level …) =	Low (level …) =

These factors can impact on the output variability in terms of time and precision as reported in the following table TABLE 58.

TABLE 125: OUTPUT VARIABILITY OF "ANALYSE CURRENT WEATHER MAP" FUNCTION

Mental model about current weather conditions built		
Time	Too early	The pilot builds her/his mental model too early without taking into account the current information.
	On time	The pilot builds her/his mental model on time taking into account the current information.
	Too late	The pilot builds her/his mental model too late taking into account dated information.
Precision	Imprecise	The pilot builds her/his mental model taking into account only some information
	Acceptable	The pilot builds her/his mental model in an acceptable manner
	Precise	The pilot precisely builds her/his mental model

The following figure represents the "Select different range map" function as FRAM unit.

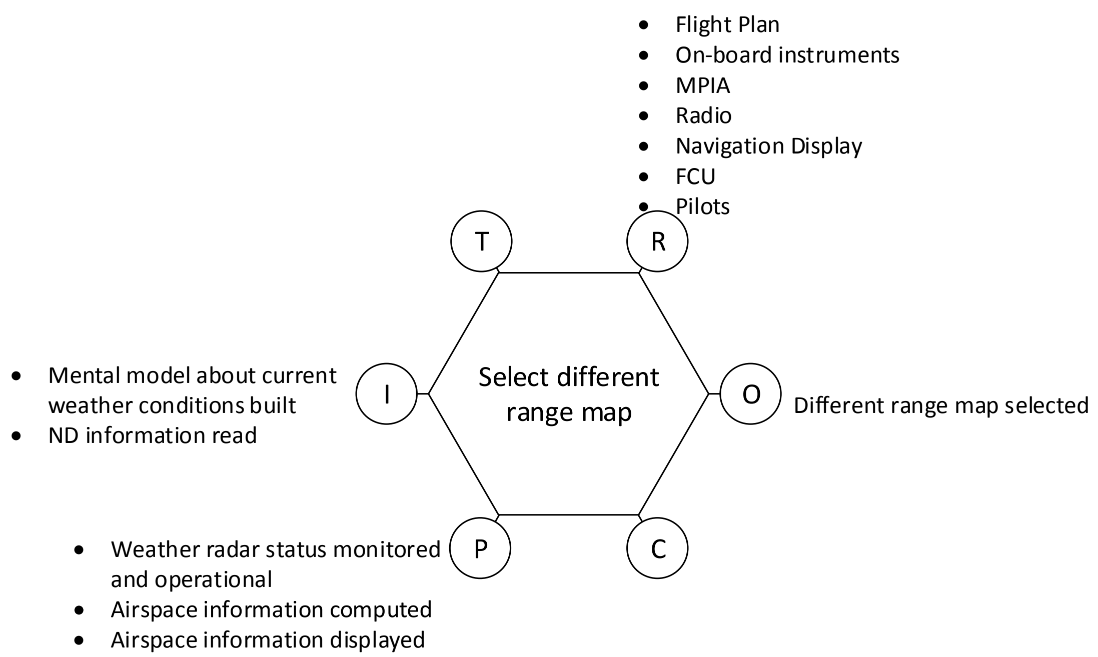


Figure 96: "Select different range map" function as FRAM unit

The following table reports the explanation of Input, Output, Precondition, Resource, Control, and Time aspects.

TABLE 126: "SELECT DIFFERENT RANGE MAP" FUNCTION FRAM TABLE

SELECT DIFFERENT RANGE MAP	
Input(s)	• ND information read • Mental model about current weather conditions built
Output(s)	• Different range map selected
Precondition	• Weather radar status monitored and operational • Airspace information computed • Airspace information displayed
Resource(s)	• Flight plan • On-board instruments • MPIA • Radio • Navigation Display • FCU • Pilots
Control(s)	N/A
Time	*Repeated Function*

Table 127 and Table 128 report respectively the endogenous and the exogenous factors that can affect the variability of the described function. The impact of these factors can be positive, neutral or negative represented by a green, grey or red arrow respectively.

TABLE 127: ENDOGENOUS FACTORS OF VARIABILITY RELATED TO "SELECT DIFFERENT RANGE MAP" FUNCTION

Experience	High (years …) = ↑	Medium (years …) = ↔	N/A (people have been trained and qualified)
Focus of attention	High (level …) = ↑	Medium (level …) = ↔	Low (level …) = ↓
Situation Awareness	High (level …) = ↑	Medium (level …) = ↔	Low (level …) = ↓
Workload	High (level …) = ↓	Medium (level …) = ↔	Low (level …) = ↑
Stress	High (level …) = ↓	Medium (level …) = ↔	Low (level …) = ↑
Complacency	High (level …) = ↓	Medium (level …) = ↔	Low (level …) = ↑
Fatigue	High (level …) = ↓	Medium (level …) = ↔	Low (level …) = ↑
System performance	High (level) = ↑	Medium (level …) = ↔	Low (level …) = ↓
Status of the system	High (level …) = ↑	Medium (level …) = ↔	Low (level …) = ↓
Knowledge	High (No. of procedures/strategies) = ↑	Medium (No. of procedures) = ↔	N/A

TABLE 128: EXOGENOUS FACTORS OF VARIABILITY RELATED TO "SELECT DIFFERENT RANGE MAP" FUNCTION

Weather perturbations	High (level …) = ↓	Medium (level …) = ↔	Low (level …) = ↑

These factors can impact on the output variability in terms of time and precision as reported in the following table.

TABLE 129: OUTPUT VARIABILITY OF "SELECT DIFFERENT RANGE MAP" FUNCTION

Different range map selected		
Time	Too early	The pilot selects the different range map too early without taking into account the previous range.
	On time	The pilot selects the different range map on time taking into account the previous range.
	Too late	The pilot selects the different range map too late and s/he needs to select again the previous one.
Precision	Imprecise	The pilot selects the different range map but doesn't pay appropriate attention to the selection of the range
	Acceptable	The pilot selects the different range s/he was targeting.
	Precise	The pilot takes care of ensuring that s/he selects the targeted range.

The following figure represents the "Detect weather target" function as FRAM unit.

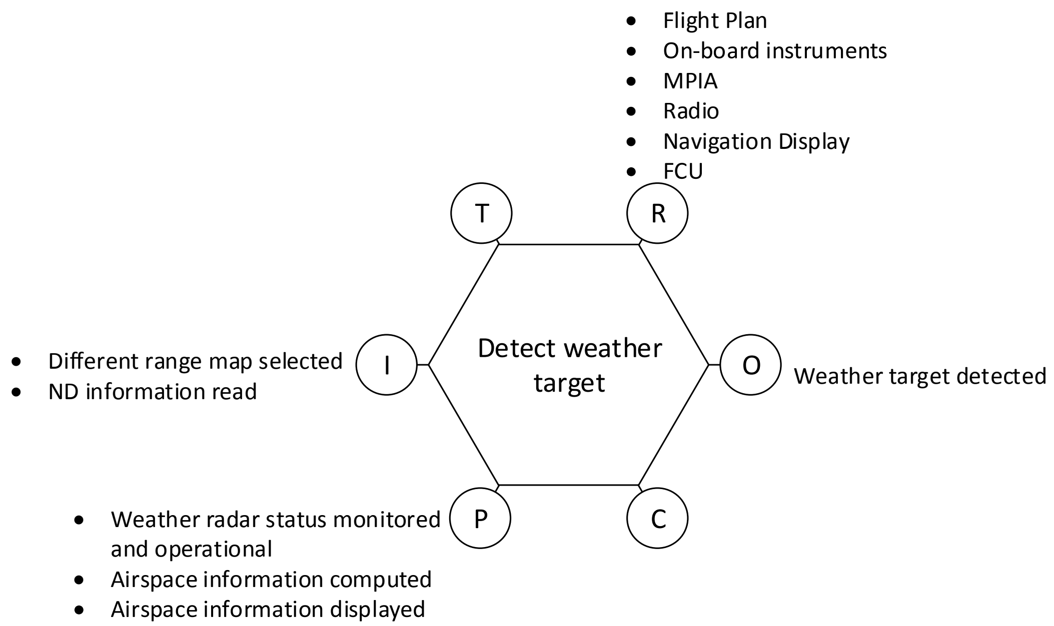


Figure 97: "Detect weather target" function as FRAM unit

The following table reports the explanation of Input, Output, Precondition, Resource, Control, and Time aspects.

TABLE 130: "DETECT WEATHER TARGET" FUNCTION FRAM TABLE

DETECT WEATHER TARGET	
Input(s)	• Different range map selected • ND information read
Output(s)	• Weather target detected
Precondition	• Weather radar status monitored and operational • Airspace information computed • Airspace information displayed
Resource(s)	• Flight plan • On-board instruments • MPIA • Radio • Navigation Display • FCU
Control(s)	N/A
Time	*Continuous Function – competing with other functions*

Table 131 and Table 132 report respectively the endogenous and the exogenous factors that can affect the variability of the described function. The impact of these factors can be positive, neutral or negative represented by a green, grey or red arrow respectively.

TABLE 131: ENDOGENOUS FACTORS OF VARIABILITY RELATED TO "DETECT WEATHER TARGET" FUNCTION

System performance	High (level) = ↑	Medium (level …) = ↔	Low (level …) = ↓
Status of the system	High (level …) = ↑	Medium (level …) = ↔	Low (level …) = ↓

TABLE 132: EXOGENOUS FACTORS OF VARIABILITY RELATED TO "DETECT WEATHER TARGET" FUNCTION

Weather perturbations	High (level …) = ↓	Medium (level …) = ↔	Low (level …) = ↑

These factors can impact on the output variability in terms of time and precision as reported in the following table.

TABLE 133: OUTPUT VARIABILITY OF "DETECT WEATHER TARGET" FUNCTION

Weather target detected		
Time	Too early	N/A
	On time	The weather target are successfully detected
	Too late	The weather target are not detected in time
Precision	Imprecise	The weather target are not correctly detected
	Acceptable	The weather target are correctly detected
	Precise	The weather target are correctly detected

The following figure represents the “Recognise weather target” function as FRAM unit.

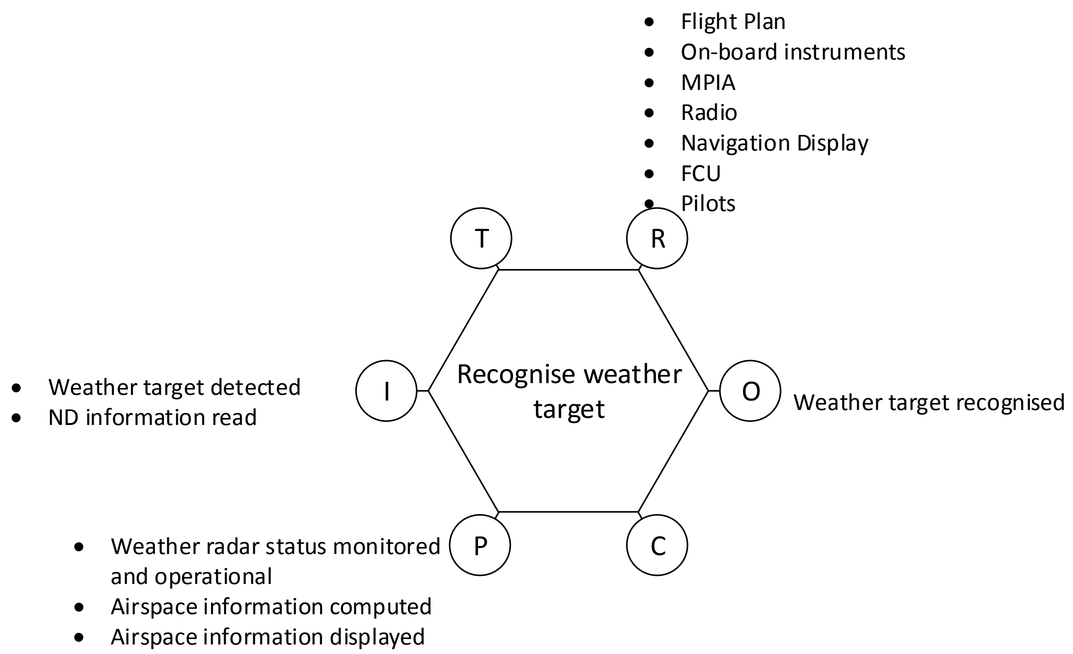


Figure 98: “Recognise weather target” function as FRAM unit

The following table reports the explanation of Input, Output, Precondition, Resource, Control, and Time aspects.

TABLE 134: "RECOGNISE WEATHER TARGET" FUNCTION FRAM TABLE

RECOGNISE WEATHER TARGET	
Input(s)	• Weather target detected • ND information read
Output(s)	• Weather target recognised
Precondition	• Weather radar status monitored and operational • Airspace information computed • Airspace information displayed
Resource(s)	• Flight plan • On-board instruments • MPIA • Radio • Navigation Display • FCU • Pilots
Control(s)	N/A
Time	*Repeated Function – competing with other functions*

Table 135 and Table 136 report respectively the endogenous and the exogenous factors that can affect the variability of the described function. The impact of these factors can be positive, neutral or negative represented by a green, grey or red arrow respectively.

TABLE 135: ENDOGENOUS FACTORS OF VARIABILITY RELATED TO "RECOGNISE WEATHER TARGET" FUNCTION

Experience	High (years …) = ↑	Medium (years …) = ↔	N/A (people have been trained and qualified)
Focus of attention	High (level …) = ↑	Medium (level …) = ↔	Low (level …) = ↓
Situation Awareness	High (level …) = ↑	Medium (level …) = ↔	Low (level …) = ↓
Workload	High (level …) = ↓	Medium (level …) = ↔	Low (level …) = ↑
Stress	High (level …) = ↓	Medium (level …) = ↔	Low (level …) = ↑
Complacency	High (level …) = ↓	Medium (level …) = ↔	Low (level …) = ↑
Fatigue	High (level …) = ↓	Medium (level …) = ↔	Low (level …) = ↑
Knowledge	High (No. of procedures/strategies) = ↑	Medium (No. of procedures) = ↔	N/A

TABLE 136: EXOGENOUS FACTORS OF VARIABILITY RELATED TO "RECOGNISE WEATHER TARGET" FUNCTION

Weather perturbations	High (level …) =	Medium (level …) =	Low (level …) =

These factors can impact on the output variability in terms of time and precision as reported in the following table.

TABLE 137: OUTPUT VARIABILITY OF "RECOGNISE WEATHER TARGET" FUNCTION

Weather target recognised		
Time	Too early	N/A
	On time	The pilot recognises the weather target on time for applying the adequate procedure.
	Too late	The pilot recognises the weather target too late for applying the adequate procedure.
Precision	Imprecise	The pilot partially recognises the weather target or doesn't pay appropriate attention to the recognition of the weather target
	Acceptable	The pilot recognises the weather target in an acceptable manner
	Precise	The pilot precisely recognises the weather target

The following figure represents the "Identify the adequate procedure" function as FRAM unit.

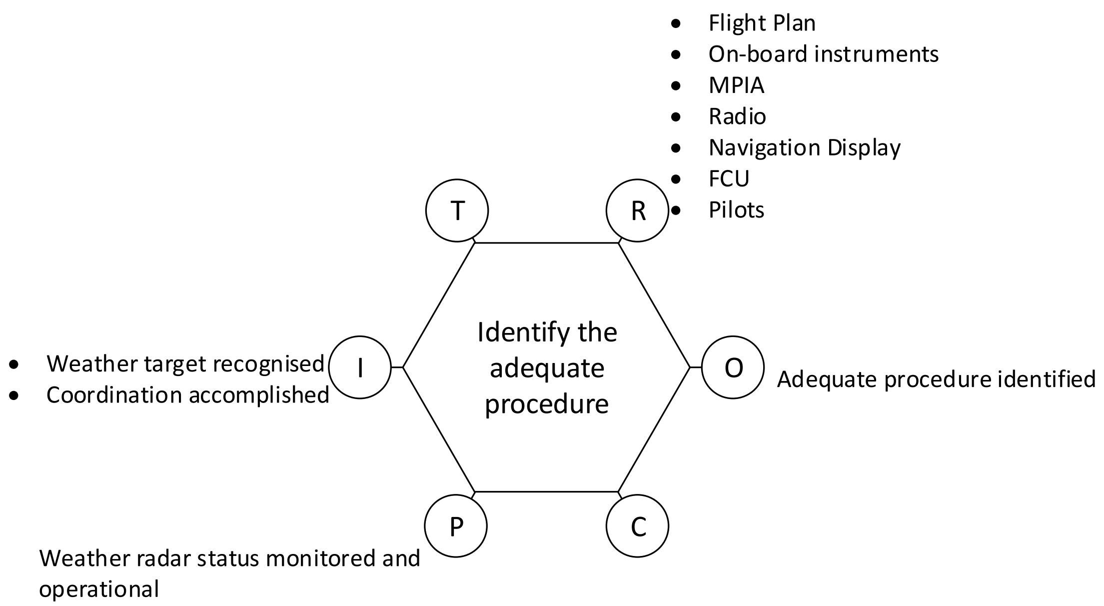


Figure 99: "Identify the adequate procedure" function as FRAM unit

The following table reports the explanation of Input, Output, Precondition, Resource, Control, and Time aspects.

TABLE 138: "IDENTIFY THE ADEQUATE PROCEDURE" FUNCTION FRAM TABLE

IDENTIFY THE ADEQUATE PROCEDURE	
Input(s)	• Weather target recognised • Coordination accomplished
Output(s)	• Adequate procedure identified
Precondition	• Weather radar status monitored and operational
Resource(s)	• Flight plan • On-board instruments • MPIA • Radio • Navigation Display • FCU • Pilots
Control(s)	N/A
Time	*Continuous Function*

Table 139 and Table 140 report respectively the endogenous and the exogenous factors that can affect the variability of the described function. The impact of these factors can be positive, neutral or negative represented by a green, grey or red arrow respectively.

TABLE 139: ENDOGENOUS FACTORS OF VARIABILITY RELATED TO "IDENTIFY THE ADEQUATE PROCEDURE" FUNCTION

Experience	High (years …) =	Medium (years …) =	N/A (people have been trained and qualified)
Focus of attention	High (level …) =	Medium (level …) =	Low (level …) =
Situation Awareness	High (level …) =	Medium (level …) =	Low (level …) =
Workload	High (level …) =	Medium (level …) =	Low (level …) =
Stress	High (level …) =	Medium (level …) =	Low (level …) =
Complacency	High (level …) =	Medium (level …) =	Low (level …) =
Fatigue	High (level …) =	Medium (level …) =	Low (level …) =
Knowledge	High (No. of procedures/strategies) =	Medium (No. of procedures) =	N/A

TABLE 140: EXOGENOUS FACTORS OF VARIABILITY RELATED TO "APPLY THE ADEQUATE PROCEDURE" FUNCTION

Weather perturbations	High (level …) =	Medium (level …) =	Low (level …) =

These factors can impact on the output variability in terms of time and precision as reported in the following table.

TABLE 141: OUTPUT VARIABILITY OF "IDENTIFY THE ADEQUATE PROCEDURE" FUNCTION

Adequate procedure identified		
Time	Too early	N/A
	On time	The pilot identifies the adequate procedure on time.
	Too late	The pilot identifies the adequate procedure too late for avoiding the recognised weather target.
Precision	Imprecise	The pilot partially identifies the adequate procedure or doesn't pay appropriate attention to the identification of the adequate procedure
	Acceptable	The pilot identifies the adequate procedure in an acceptable manner
	Precise	The pilot precisely identifies the adequate procedure

The following figure represents the "Request a clearance" function as FRAM unit.

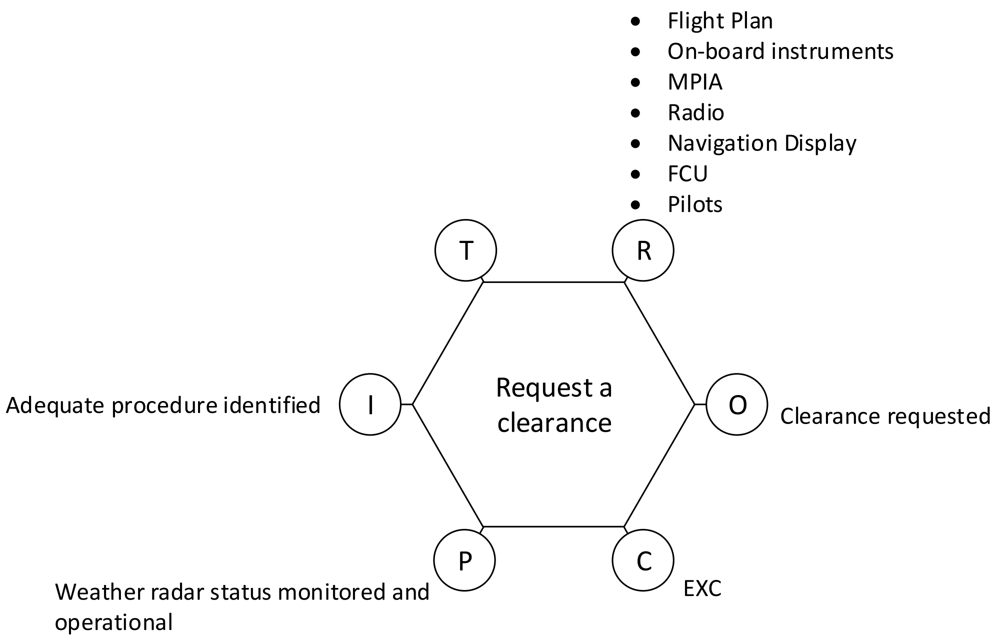


Figure 100: "Request a clearance" function as FRAM unit

The following table reports the explanation of Input, Output, Precondition, Resource, Control, and Time aspects.

TABLE 142: "REQUEST A CLEARANCE" FUNCTION FRAM TABLE

REQUEST A CLEARANCE	
Input(s)	• Adequate procedure identified
Output(s)	• Clearance requested
Precondition	• Weather radar status monitored and operational
Resource(s)	• Flight plan • On-board instruments • MPIA • Radio • Navigation Display • FCU • Pilots
Control(s)	• EXC
Time	*On demand*

Table 143 and Table 144 report respectively the endogenous and the exogenous factors that can affect the variability of the described function. The impact of these factors can be positive, neutral or negative represented by a green, grey or red arrow respectively.

TABLE 143: ENDOGENOUS FACTORS OF VARIABILITY RELATED TO "REQUEST A CLEARANCE" FUNCTION

Experience	High (years …) =	Medium (years …) =	N/A (people have been trained and qualified)
Focus of attention	High (level …) =	Medium (level …) =	Low (level …) =
Situation Awareness	High (level …) =	Medium (level …) =	Low (level …) =
Workload	High (level …) =	Medium (level …) =	Low (level …) =
Stress	High (level …) =	Medium (level …) =	Low (level …) =
Complacency	High (level …) =	Medium (level …) =	Low (level …) =
Fatigue	High (level …) =	Medium (level …) =	Low (level …) =
Knowledge	High (No. of procedures/strategies) =	Medium (No. of procedures) =	N/A

TABLE 144: EXOGENOUS FACTORS OF VARIABILITY RELATED TO "REQUEST A CLEARANCE" FUNCTION

Weather perturbations	High (level …) =	Medium (level …) =	Low (level …) =

These factors can impact on the output variability in terms of time and precision as reported in the following table.

TABLE 145: OUTPUT VARIABILITY OF "REQUEST A CLEARANCE" FUNCTION

Clearance requested		
Time	Too early	N/A
	On time	The pilot requests the clearance on time.
	Too late	The pilot requests the clearance too late for being managed by the EXC.
Precision	Imprecise	The pilot doesn't pay appropriate attention to the clearance request
	Acceptable	The pilot applies requests the clearance in an acceptable manner
	Precise	The pilot precisely requests the clearance

The following figure represents the "Coordinate with EXC" function as FRAM unit.

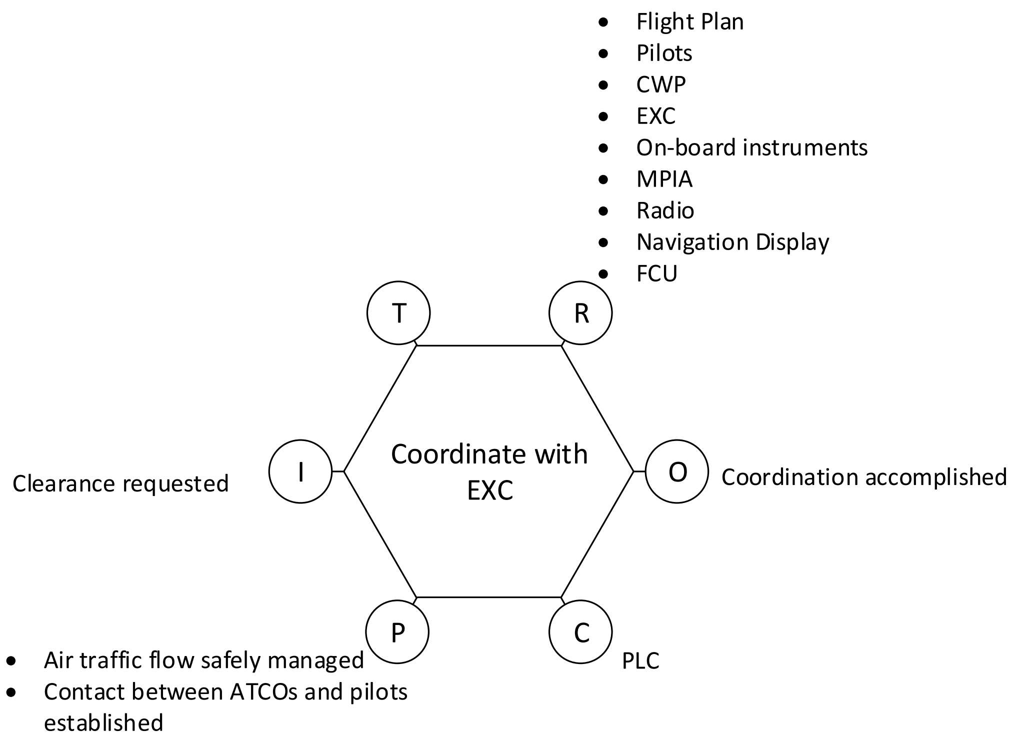


Figure 101: "Coordinate with EXC" function as FRAM unit

The following table reports the explanation of Input, Output, Precondition, Resource, Control, and Time aspects.

TABLE 146: "COORDINATE WITH EXC" FUNCTION FRAM TABLE

COORDINATE WITH EXC	
Input(s)	• Clearance requested
Output(s)	• Coordination accomplished
Precondition	• Air traffic flow safely managed • Contact between ATCOs and pilots established
Resource(s)	• Flight Plan • Pilots • CWP • EXC • On-board instruments • MPIA • Radio • Navigation Display • FCU
Control(s)	• PLC
Time	*Repeated function, every time the procedures require the coordination (aircraft assumptions/transfer – clearance)*

Table 147 and Table 148 report respectively the endogenous and the exogenous factors that can affect the variability of the described function. The impact of these factors can be positive, neutral or negative represented by a green, grey or red arrow respectively.

TABLE 147: ENDOGENOUS FACTORS OF VARIABILITY RELATED TO "COORDINATE WITH EXC" FUNCTION

Experience	High (years …) =	Medium (years …) =	N/A (people have been trained and qualified)
Focus of attention	High (level …) =	Medium (level …) =	Low (level …) =
Situation Awareness	High (level …) =	Medium (level …) =	Low (level …) =
Workload	High (level …) =	Medium (level …) =	Low (level …) =
Stress	High (level …) =	Medium (level …) =	Low (level …) =
Complacency	High (level …) =	Medium (level …) =	Low (level …) =
Fatigue	High (level …) =	Medium (level …) =	Low (level …) =
Knowledge	High (No. of procedures/strategies) =	Medium (No. of procedures) =	N/A

TABLE 148: EXOGENOUS FACTORS OF VARIABILITY RELATED TO "COORDINATE WITH EXC" FUNCTION

N° Aircraft	High (n …) =	Medium (n …) =	Low (n …) =
Complexity of the surrounding air traffic	High (level …) =	Medium (level …) =	Low (level …) =

These factors can impact on the output variability in terms of time and precision as reported in TABLE 82the following table

TABLE 149: OUTPUT VARIABILITY OF "REQUEST A CLEARANCE" FUNCTION

Coordination accomplished		
Time	Too early	N/A
	On time	The coordination between the EXC and the pilot is accomplished on time.
	Too late	The coordination between the EXC and the pilot is accomplished too late for the clearance acknowledging
Precision	Imprecise	The coordination between the EXC and the pilot is partially accomplished and the clearance acknowledged is imprecise
	Acceptable	The coordination between the EXC and the pilot is accomplished in an acceptable manner
	Precise	The coordination between the EXC and the pilot is precisely accomplished through clearance acknowledged

The following figure represents the "Execute the clearance" function as FRAM unit.

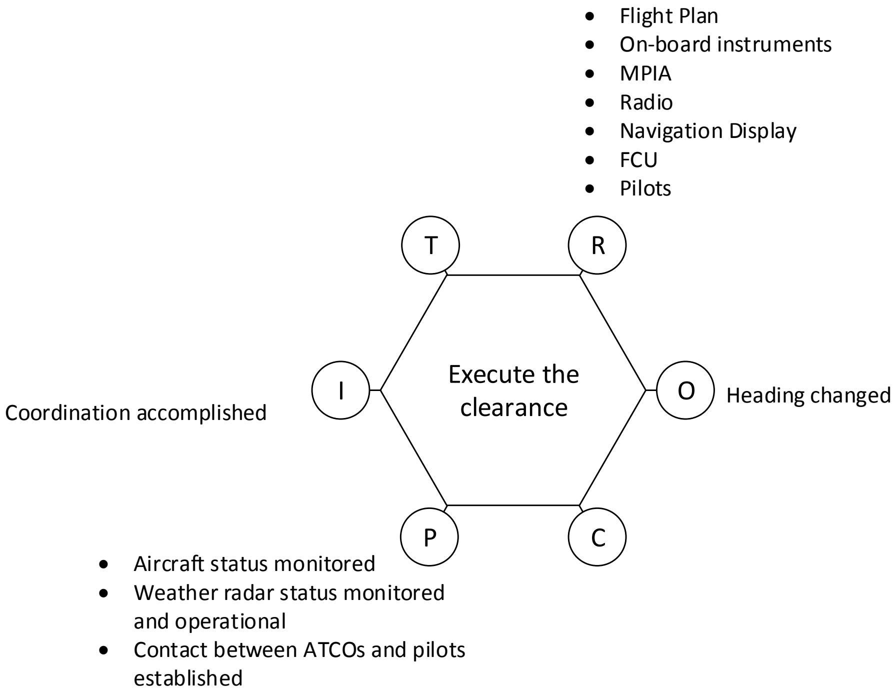


Figure 102: "Execute the clearance" function as FRAM unit

The following table reports the explanation of Input, Output, Precondition, Resource, Control, and Time aspects.

TABLE 150: "EXECUTE THE CLEARANCE" FUNCTION FRAM TABLE

EXECUTE THE CLEARANCE	
Input(s)	• Coordination accomplished
Output(s)	• Heading changed
Precondition	• Aircraft status monitored • Weather radar status monitored and operational • Contact between ATCOs and pilots established
Resource(s)	• Flight plan • On-board instruments • MPIA • Radio • Navigation Display • FCU • Pilots
Control(s)	• EXC
Time	*On demand – competing with other functions*

Table 151 and Table 152 report respectively the endogenous and the exogenous factors that can affect the variability of the described function. The impact of these factors can be positive, neutral or negative represented by a green, grey or red arrow respectively.

TABLE 151: ENDOGENOUS FACTORS OF VARIABILITY RELATED TO "EXECUTE THE CLEARANCE" FUNCTION

Experience	High (years …) = ↑	Medium (years …) = ↔	N/A (people have been trained and qualified)
Focus of attention	High (level …) = ↑	Medium (level …) = ↔	Low (level …) = ↓
Situation Awareness	High (level …) = ↑	Medium (level …) = ↔	Low (level …) = ↓
Workload	High (level …) = ↓	Medium (level …) = ↔	Low (level …) = ↑
Stress	High (level …) = ↓	Medium (level …) = ↔	Low (level …) = ↑
Complacency	High (level …) = ↓	Medium (level …) = ↔	Low (level …) = ↑
Fatigue	High (level …) = ↓	Medium (level …) = ↔	Low (level …) = ↑
System performance	High (level) = ↑	Medium (level …) = ↔	Low (level …) = ↓
Status of the system	High (level …) = ↑	Medium (level …) = ↔	Low (level …) = ↓
Knowledge	High (years …) = ↑	Medium (years …) = ↔	N/A

TABLE 152: EXOGENOUS FACTORS OF VARIABILITY RELATED TO "EXECUTE THE CLEARANCE" FUNCTION

Weather perturbations	High (level …) = ↓	Medium (level …) = ↔	Low (level …) = ↑

These factors can impact on the output variability in terms of time and precision as reported in the following tableTABLE 86.

TABLE 153: OUTPUT VARIABILITY OF "EXECUTE THE CLEARANCE" FUNCTION

Heading changed		
Time	Too early	N/A
	On time	The pilot changes the heading on time for avoiding the weather target
	Too late	The pilot changes the heading too late for avoiding the weather target.
Precision	Imprecise	The pilot doesn't pay appropriate attention to the heading change
	Acceptable	The pilot changes the heading in an acceptable manner.
	Precise	The pilot takes care of ensuring that s/he changes the heading

Once all the identified functions have been represented as FRAM units, the endogenous and exogenous factors have been listed and their potential impact on the function output variability has been expressed in terms of time and precision, it is possible to translate the Timeline as FRAM instantiation. The instantiation supports in preliminary detecting the functional resonance between the functions.

Figure 103 illustrates the FRAM instantiation of the WXR case study. The instantiation shows how the functions are coupled between them. These couplings allow us to preliminary detect the functional resonance and how they can be affected from its propagation. In addition, Figure 104 and Figure 105 respectively illustrate the FRAM instantiations of the "Check weather conditions" which is a macro function consisting of a set of lower-functions and of the "Manage tilt angle" which is a lower-function of "Manage weather radar status" macro function. The instantiation shows how the functions are coupled between them. These couplings allow us to preliminary detect the functional resonance and how they can be affected from its propagation

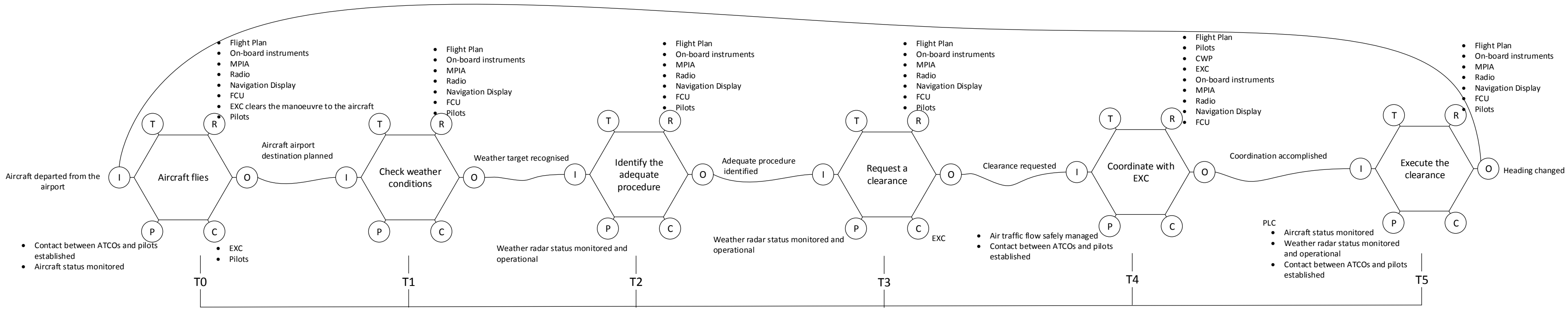


Figure 103: Scenario – FRAM instantiation of the WXR case study

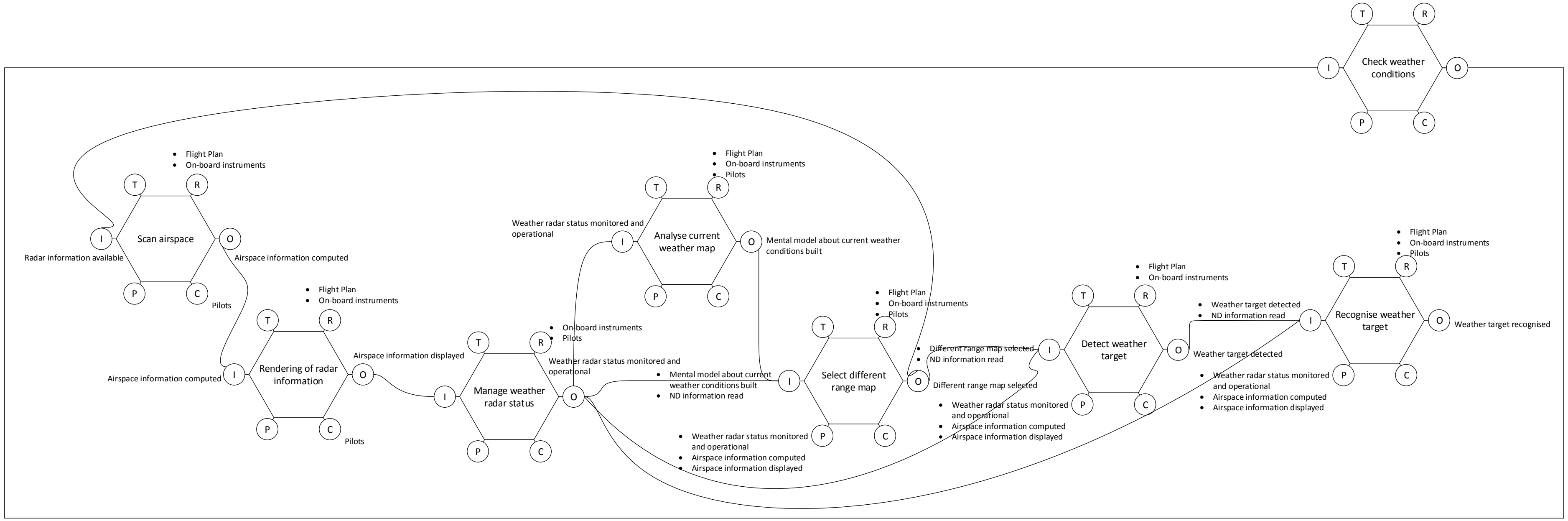


Figure 104: Scenario – FRAM instantiation of the "Check weather conditions" macro function

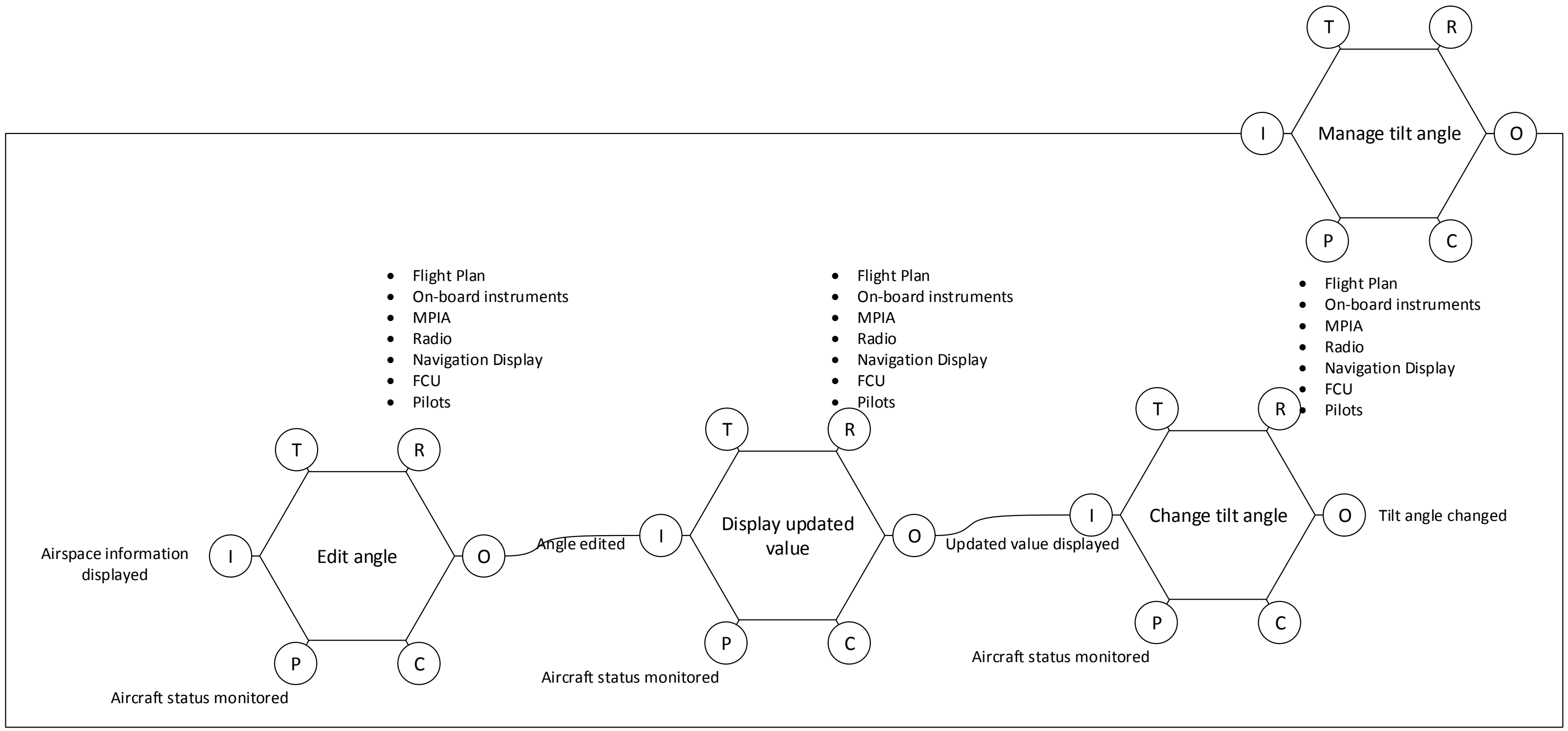


Figure 105: Scenario – FRAM instantiation of the "Manage tilt angle" lower-function of "Manage weather radar status"

As described in Chapter 5 – Section 1.2.1, in parallel, for each identified type of function, the analyst will use dedicated modelling techniques for building the related models. According to Chapter 1 – Section 2.2, we selected HAMSTERS for modelling human functions. According to Chapter 2 – Section 2.2, we selected ICO&PetShop for modelling system functions. According to Chapter 2 – Section 5, we selected both modelling techniques for modelling interactive functions. This flow of the sub-phase outputs are “FRAM instantiations”, “HAMSTERS models”, and “ICO&PetShop models”.

For the WXR case study, Figure 106 illustrates the task model of the “Check weather conditions” main goal. According to the case study, one of the main goals of the pilot is “Check weather conditions” which includes several sub-routines such as “Manage weather status” that has to be performed iteratively (circular arrow symbol).

The task model of this sub-routine is illustrated in Figure 107 which represents the pilot activities performed in order to “Manage weather radar”. At the higher level of the tree, there is an abstract activity “Set up WXR” that is interrupted (operator [>) by a cognitive decision task “Decide WXR is ready”.

Connection between pilot’s activities and cockpit functions is made through interactive tasks (as input “Switch to WXON” and output “Check updated value”). The time required for performing the task heavily depends on the radar type. Such behavioural aspects of systems can be modelled using ICO notation and PetShop tool as detailed in Chapter 2 – Section 2.2. This task model corresponds to the manipulation of the user interface presented in Figure 52a). From these models we can see that the tasks to be performed in order to check weather conditions in a given direction are rather complex. The time required to perform them depends on 3 elements: the operator’s performance in terms of motor movements, perception and cognitive processing. Human performance models such as the one proposed in (Card, Moran, & Newell, 1986) can be used to assess difficulties and delays but the overall performance of the socio-technical system involves interaction and system execution times.

At the same time, tasks performed in a socio-technical system involve human interactions, such as the “Execute the clearance (change heading)” task. Figure 108 illustrates the task model of the “Execute the clearance (change heading)” task. At the higher level of the tree, there is an iterative activity “Manage change heading request” that is interrupted (operator [>) by a cognitive analysis task “Analyse that change heading clearance has been received” which can be allowed to the pilot to “Enter the new heading” abstract task. In the scenario under analysis in which the pilot can request a clearance at time, the iterative activity “Manage change heading request” has to be performed several times in order to find the appropriate clearance matching the pilot and the EXC needs.

However, as represented in the model, the pilot could make a choice (operator [] by “Request a clearance” abstract task): instead of following the communication/coordination protocol with the ATCOs requesting a clearance at time (“Decide about new heading” and “Ask for new heading clearance” tasks linked to the information “requested heading”), s/he can adopt a strategy (the StK “Communication/coordination with the ATCOs”). In this case, the pilot can “Decide about several possible headings” and “Ask for heading clearance with several options” providing a list of possible headings (information represented by a yellow box) to the EXC. This strategy could reduce the number of iterations between the pilot and the EXC. But this strategy is not the standard procedure and will be applied depending on the person who is in charge of the aircraft and on a common agreement with the remote EXC. However, in the current scenario, it is not needed because the pilot request is immediately accepted by the EXC.

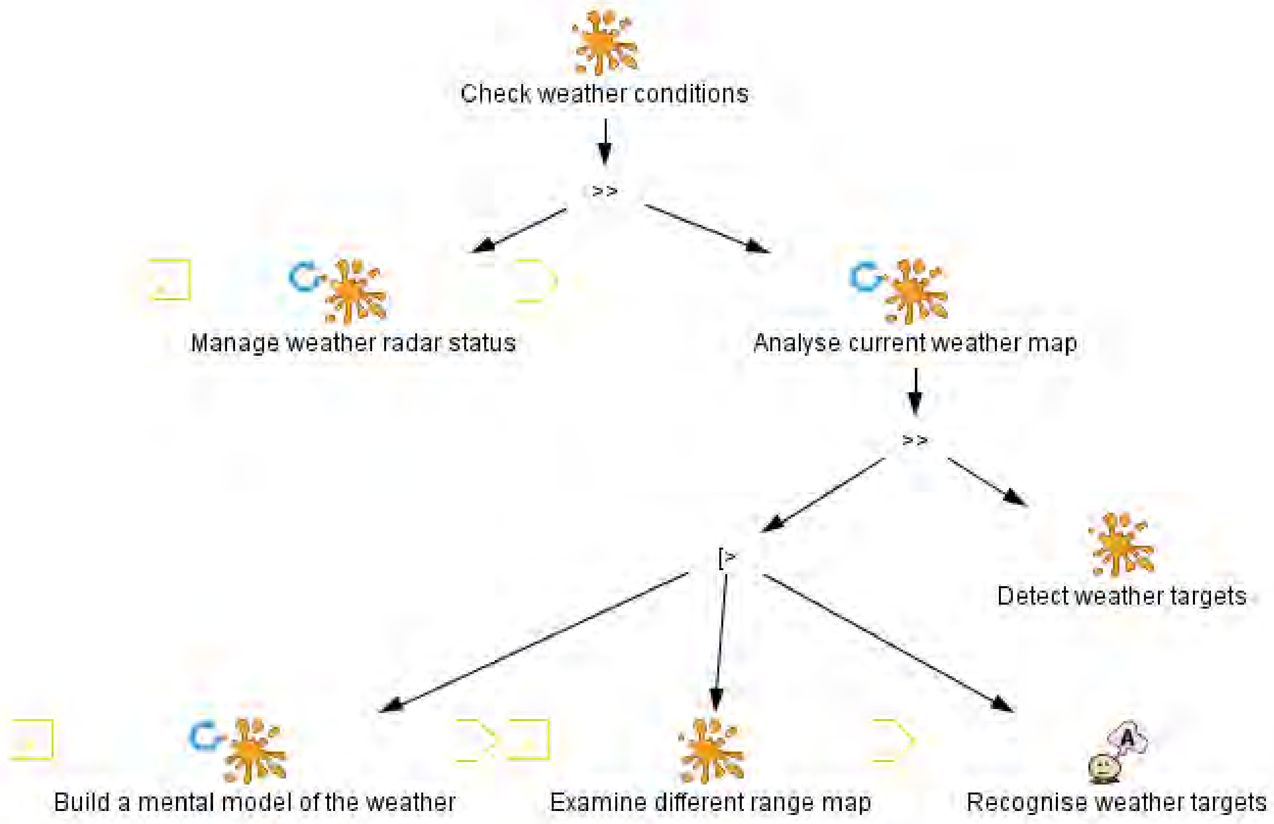


Figure 106: HAMSTERS 2.0 task model of the “Check weather conditions” main goal

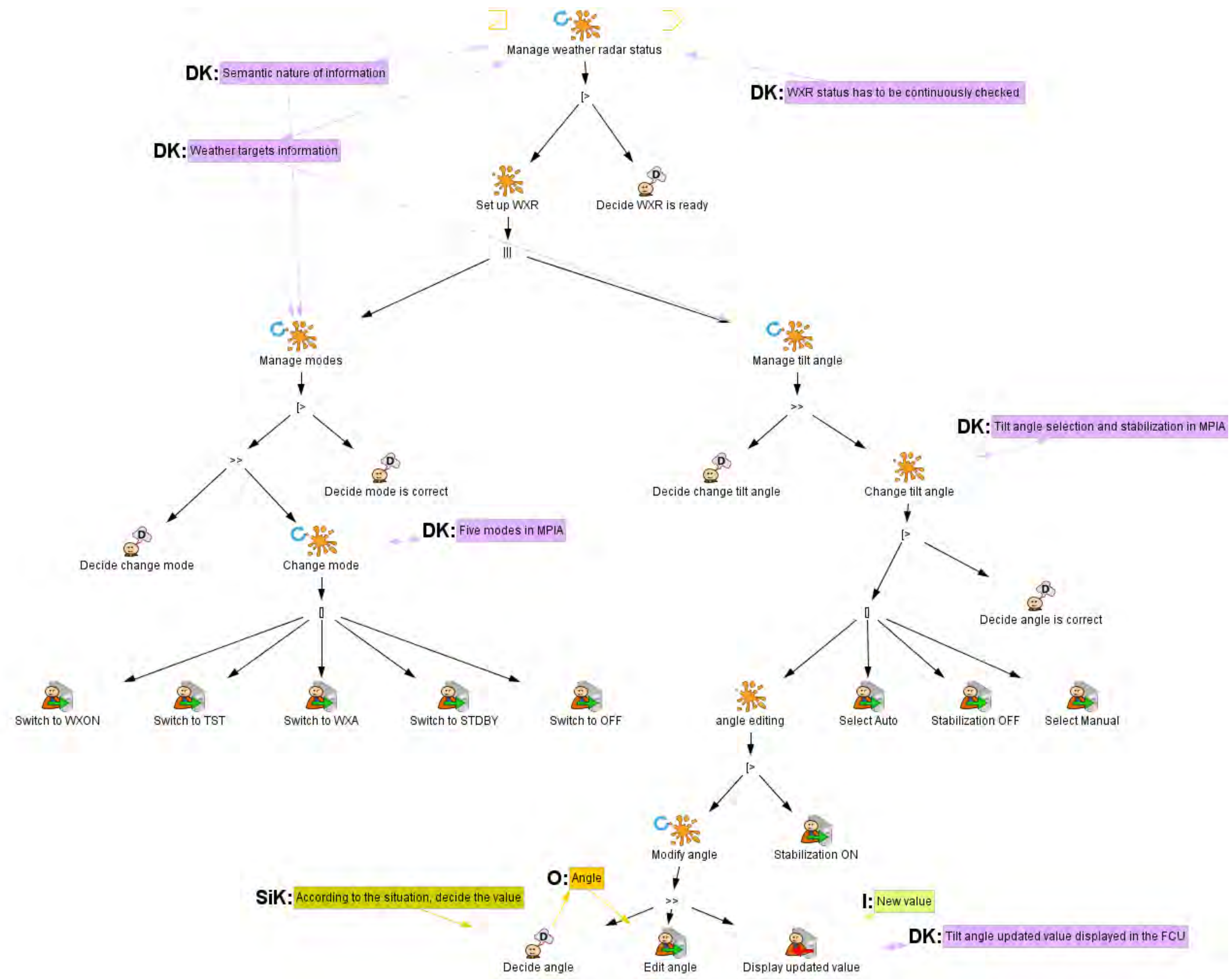


Figure 107: HAMSTERS 2.0 task model of the subroutine “Manage weather radar” task

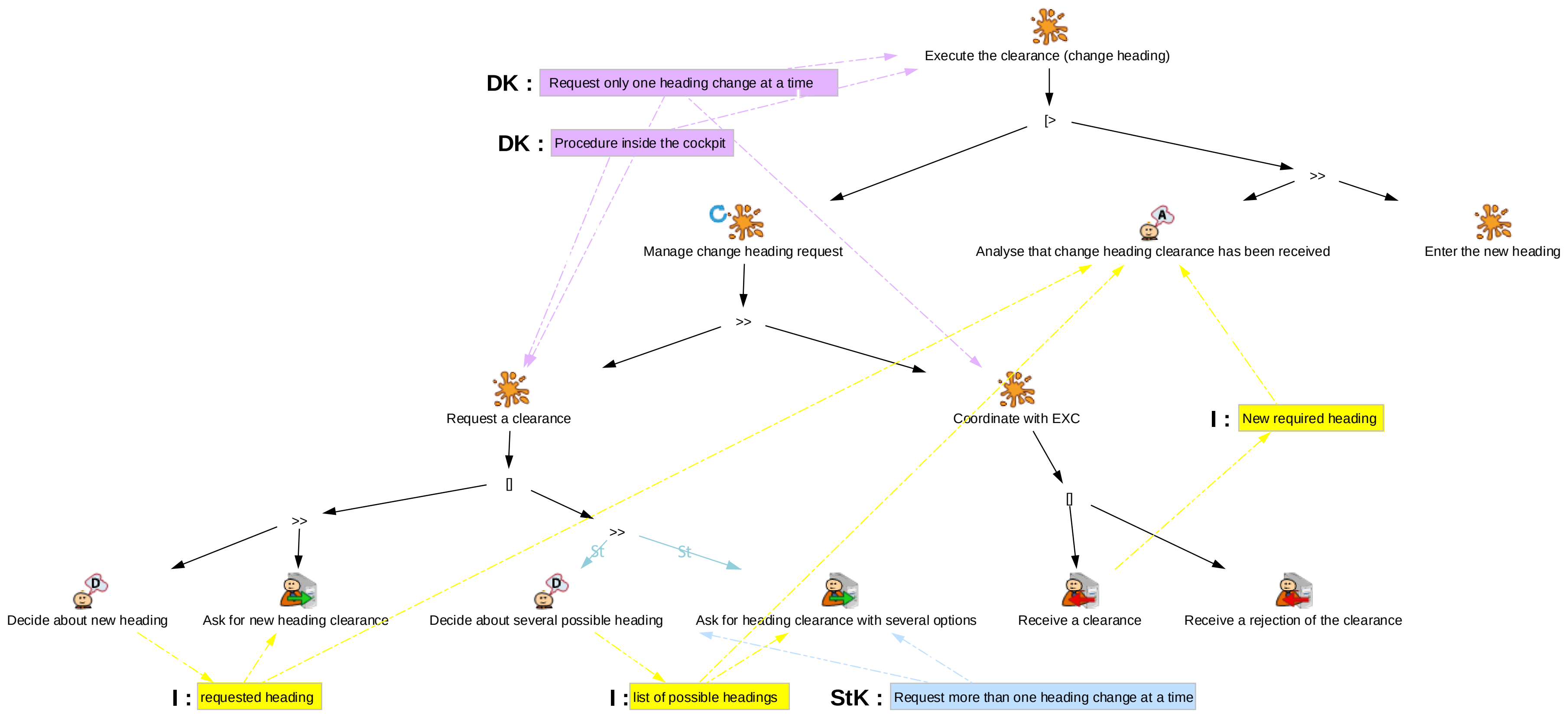


Figure 108: HAMSTERS 2.0 task model of the "Execute the clearance (change heading)" task

Regarding the system models, Figure 109 shows the ICO&PetShop model of mode selection and tilt angle setting WXR functionalities. As also represented through FRAM instantiation, the pilot has to continuously check the WXR status and select different range for detecting weather perturbations.

The ICO&PetShop model describes how it is possible to handle the weather radar configuration of both its mode and its tilt angle. Figure 109 shows the interactive means provided to the user to:

- Switch between the five available modes (upper part of the figure) using radio buttons (the five modes being WXON to activate the weather radar detection, OFF to switch it off, TST to trigger a hardware check-up, STDBY to switch it on for test only and WXA to focus detection on alerts).
- Select the tilt angle control mode (lower part of the figure) amongst three modes (fully automatic, manual with automatic stabilization and manual selection of the tilt angle).

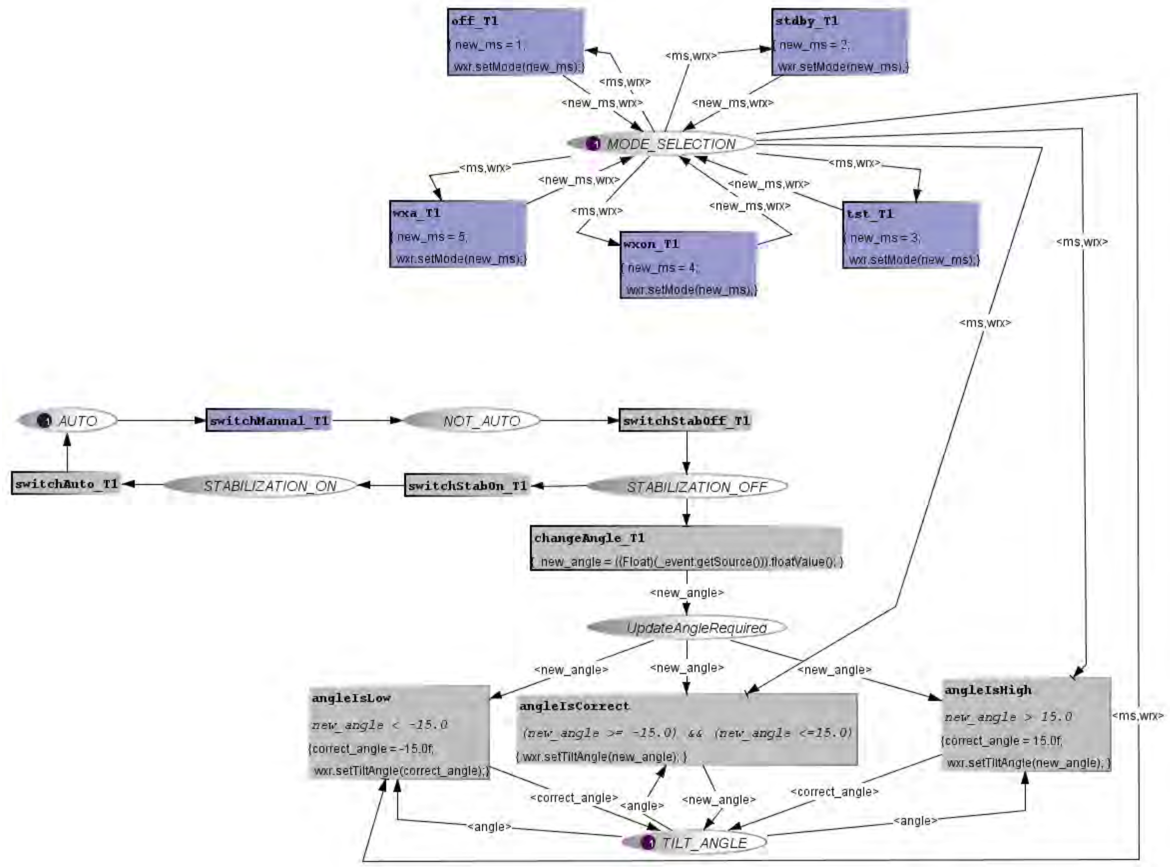


Figure 109: ICO&PetShop – Behaviour of the WRX mode selection and tilt angle setting

Figure 109 presents the description of the behaviour of this part of the interactive cockpit using the ICO formal description technique and may be divided into two parts.

- The Petri net in the upper part handles events received from the 5 radio buttons. The current selection (an integer value from 1 to 5) is carried by the token stored in MODE_SELECTION place and corresponds to one the possible radio buttons (OFF, STDBY, TST, WXON, WXA). The token is modified by the transitions (new_ms = 3 for instance) using variables on the incoming and outgoing arcs as formal parameters of the transitions. Each time the mode value is changed, the equipment part (represented by the variable wxr within the token) is set up accordingly.

- The Petri net in the lower part handles events from the four buttons and the text field (modify tilt angle). Interacting with these buttons changes the state of the application. In the current state, this part of the application is in the state fully automatic (a token is in AUTO place). To reach the state where the text field is available for the angle modification, it is necessary to bring the token to the place STABILIZATION_OFF by successively fire the two transitions switchManual_T1 and switchStabOff_T1 (by using the two buttons MANUAL and OFF represented by Figure 52), making transition change_Angle_T1 available. The selected angle must belong to the correct range (-15 to 15), controlled by the three transitions angleIsLow, angleIsCorrect and angleIsHigh. When checked, the wxr equipment tilt angle is modified, represented by the method called wxr.setTiltangle.

Figure 110 shows the ICO&PetShop model of range selection WXR functionality. As also represented through FRAM instantiation, the pilot has to select different range map for detecting weather perturbations.

The setting of the range detection of the weather radar is done using a FCU physical knob (please refer to Figure 52b) by switching between 6 values (from 1 to 6). Each time the value is set an event is raised (holding this value) by the knob and received by a dedicated part of the cockpit application. This part of the application is represented by the model of Figure 52b) Figure 110 that maps the value (form 1 to 6) into a range value that is sent to the WRX equipment.

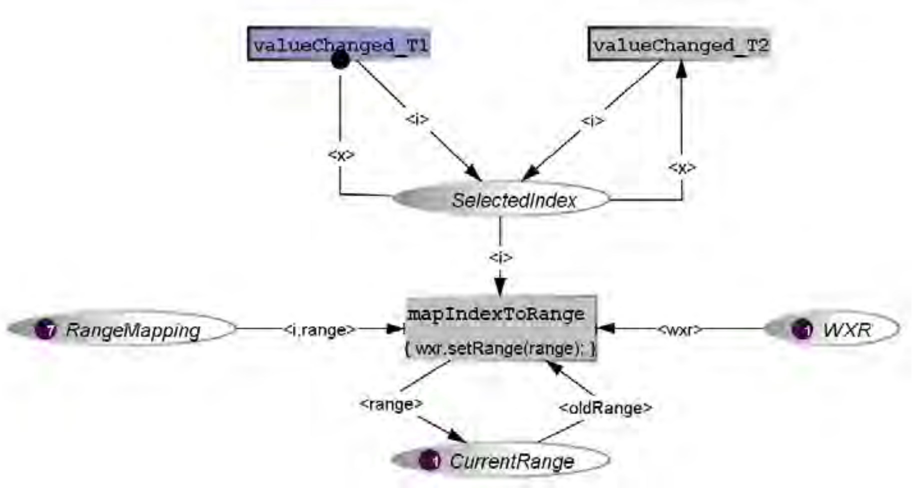


Figure 110: ICO&PetShop – Behaviour of the range selection

The event is received and the selected value is extracted by one of the two transitions called valueChanged_T1 and valueChanged_T2. The place RangeMapping contains the mapping between a value and the corresponding range (for instance 1 corresponds to range 10, 2 to 20…). Finally, the wxr equipment range is set with the selected range by the firing of transition mapIndexToRange.

In this IIa sub-phase, the other parallel flow is "Identify concepts", which will lead to the production of concept maps (for all sub-processes description please refer to Chapter 5 – Section 1.2.1). As demonstrated in PART II – Contributions - Chapter 4 , theoretically in Section 2 and practically in Section 3, concept maps provides a unified description of all the concepts (information, objects, knowledge and their relationships) and they can be used either to represent knowledge and/or to support automated systems for reasoning about the represented knowledge.

For the WXR case study, Figure 111 illustrates the concept map associated to HAMSTERS 2.0 task models for representing knowledge, information and objects (as explained theoretically in the first contribution of this thesis in Chapter 4 – Section 2) related to this case study.

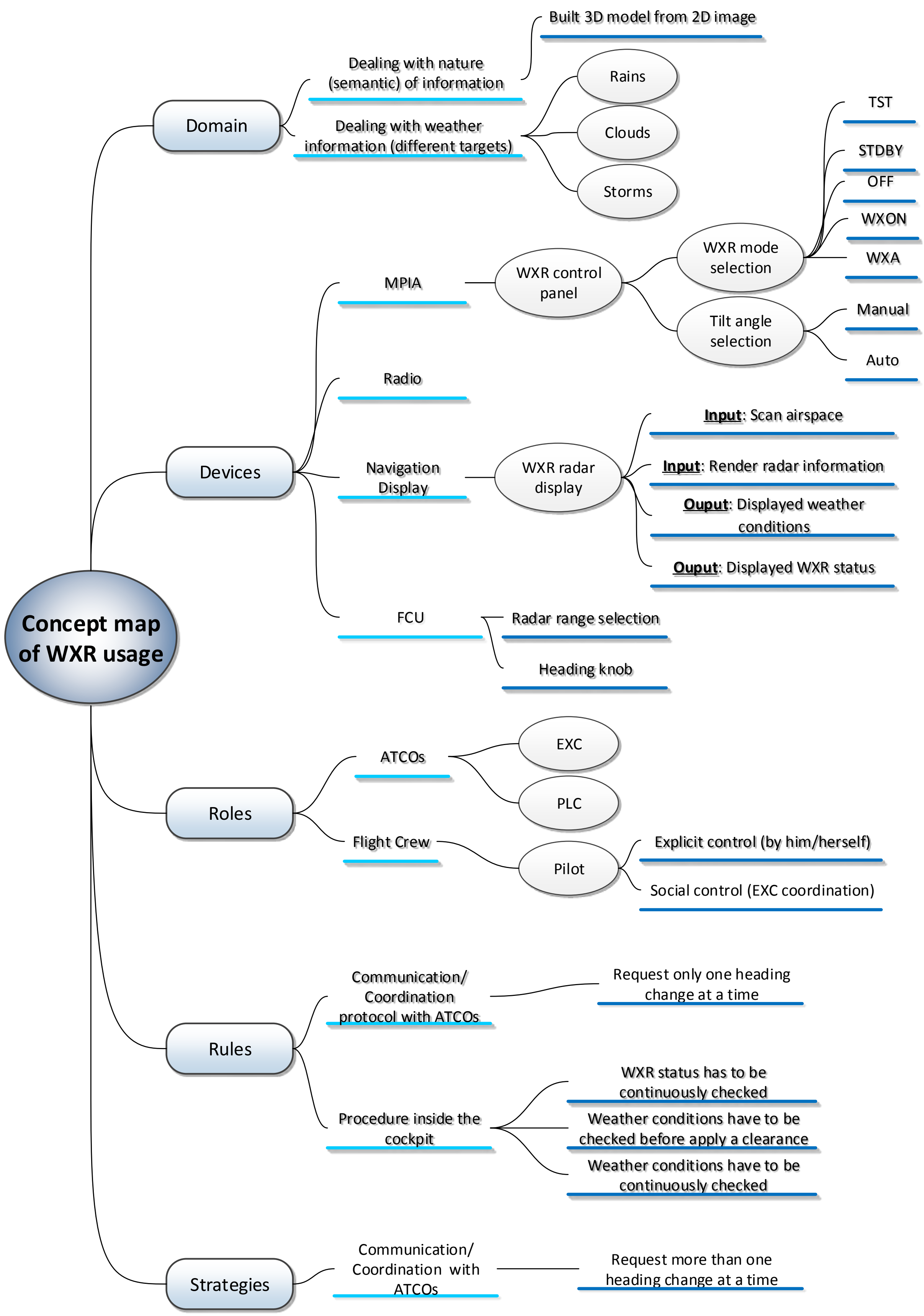


Figure 111: Concept map of WXR usage

"Figure 111: Concept map of WXR usage" represents required declarative knowledge necessary to be able to use the WXR. The concepts which are necessary or useful to perform actions during the flight phase are grouped and linked through a semantic network. It is made up of 5 key concepts or nodes using the recommended terminology explained in Section Chapter 42 of Chapter 4. These key concepts are Domain, Roles, Rules, Strategies and Devices. In this case study, these concepts are further detailed and applied to the specific context of the WXR. Structure from left to right shows the refinement of concepts, from abstract to concrete, as well as the instantiation. For example, the concept of "ATCOs" is instantiated twice: the "EXC" instance and the "PLC" instance. Lastly, relationships between refined concepts can also be represented in an explicit way, using links (as reported in the concept map of the Game of 15 in Figure 20). Unfortunately, in order to illustrate the concept map in a seamless way and facilitate its reading, these have been removed.

All the represented concepts should have a corresponding representation in one or several models produced with the task, system or organisation modelling techniques. To point out this correspondence between the WXR case study concept map and HAMSTERS 2.0, we can observe that the declarative knowledge violet box "Weather targets information" in "Figure 107: HAMSTERS 2.0 task model of the subroutine "Manage weather radar" task" corresponds to the bubble "Dealing with weather information (different targets)" in "Figure 111: Concept map of WXR usage". These correspondences enable to ensure consistency between the different views needed to properly model and analyse a partly autonomous interactive system. The concept map can be integrated into the models, as presented in next section.

In order to proceed with the next sub-phase, the analyst has to check whether or not all the functions and all the concepts are represented in the models. If all of them are not represented ("No" back loop in Figure 24), the analyst has to re-perform some previous steps. Hence, s/he has to rework on identifying and describing functions and concepts (flow of steps starting by "Identify functions" and/or flow of steps starting by "Identify concepts" in Figure 24). Once all the functions and the concepts are represented in the models, s/he can carry on to the next sub-phase.

In this case, all the functions and the concepts are presented in the models. So, we can carry on the next sub-phase.

The Integration of the Models sub-phase applied to the WXR scenario

As explained in Chapter 2 – Section 5, HAMSTERS and ICO&PetShop models can be integrated both at model and at tool level. This means that a variation of the system model that impacts a human task will trigger a change in the corresponding task model and vice versa. This helps in iteratively refining models of both types until all the corresponding models are consistent and coherent between each other. In addition, the integration of the system and task models with the FRAM instantiations, at the model level, provides a complementary view for assessing consistency and coherence between system and task models, as it enables to verify the sequence of function calls between models.

The correspondences assessment between models at tool level applied to the WXR case study

This section details the correspondences at tool level between HAMSTERS 2.0 models and ICO models regarding the case study of the WXR.

As reported in Table 154 and according to the corresponding models, the interactive input task "Change mode" (from the task model built through HAMSTERS 2.0 and shown in Figure 107) corresponds to an Event handler "ModeSelection" in the system model (which is built through ICO&PetShop and shown in Figure 109). Moreover, the principle of editing the correspondences between the two models is to put together system outputs Place Event "UpdateAngleRequired" in

Place “AngleIsCorrect” (from the ICO&PetShop model) with Interactive output tasks “Display update value” (from the HAMSTERS 2.0 model).

TABLE 154: WXR CASE STUDY – HAMSTERS 2.0AND ICO&PETSHOP MODELS CORRESPONDENCES

Pilot		
Input correspondences table		
HAMSTERS	**ICO**	
Interactive input task	**Event handler**	
Switch to WXON	wxon_T1	
Switch to TST	tst_T1	
Switch to WXA	stdby_T1	
Switch to STDBY	off_T1	
Switch to OFF	wxa_T1	
Select Auto	switchAuto_T1	
Stabilization OFF	switchStabOff_T1	
Select Manual	switchManual_T1	
Edit angle	changeAngle_T1	
Stabilization ON	switchStabOn_T1	
Select range	mapIndexToRange	
Output correspondences table		
HAMSTERS	**ICO**	
Interactive output task	**Place**	**Place Event**
Display updated value	angleIsCorrect	UpdateAngleRequired

The synergistic use allows us in representing all the elements of a partly autonomous interactive system, and in this particular case, the user interface for the WXR (illustrated in Figure 52 of Chapter 6 – Section 1). Setting up this correspondence may show inconsistencies between the task and system model such as interactive tasks not supported by the system or rendering information not useful for the tasks performance.

This integration allows a real co-evolution of two models, as the execution of one model impacts the execution of the other related model. This integration can provide designers with shorter iterations in the task and system modelling process. It also represents an improvement for the end user as the execution of the system should support training and provide contextual help.

The correspondences assessment between models at model level applied to the WXR case study

This section describes the connections between FRAM and HAMSTERS 2.0 and ICO models applied to the WXR case study. As reported in Chapter 3 – Section 5, a support tool for FRAM is not available yet. However, this method can be extended and integrated with the others two modelling techniques at model level and it plays a fundamental role in assessing the models connections.

Indeed, FRAM (and its function instantiations) can be considered as a metamodel for connecting and representing the correspondences between HAMSTERS 2.0 and ICO models. It can be defined as a correspondence editor (please refer to Chapter 5 – Section 1.2.2.2).

The task models built through HAMSTERS 2.0 have their respective “Human functions” in the FRAM instantiations. This is also true for the system models built through PetShop which have the correspondent “System functions” in FRAM. In case of the interactive functions, at each variation on the task model, which impacts on system component, corresponds a change on the system model and vice versa. This means that in FRAM the output of the “Human function” (which corresponds to the HAMSTERS model) causes an input in the “Interactive function” generating a change in the correspondent system model (modelled through PetShop).

Figure 112 shows the role of FRAM as correspondence editor in the WXR case study. On the left side, an excerpt of the "HAMSTERS 2.0 task model of the subroutine "Manage weather radar" task" (please refer to Figure 74 for the whole representation). The task model has its correspondent function, named "Edit angle", in the FRAM instantiation.

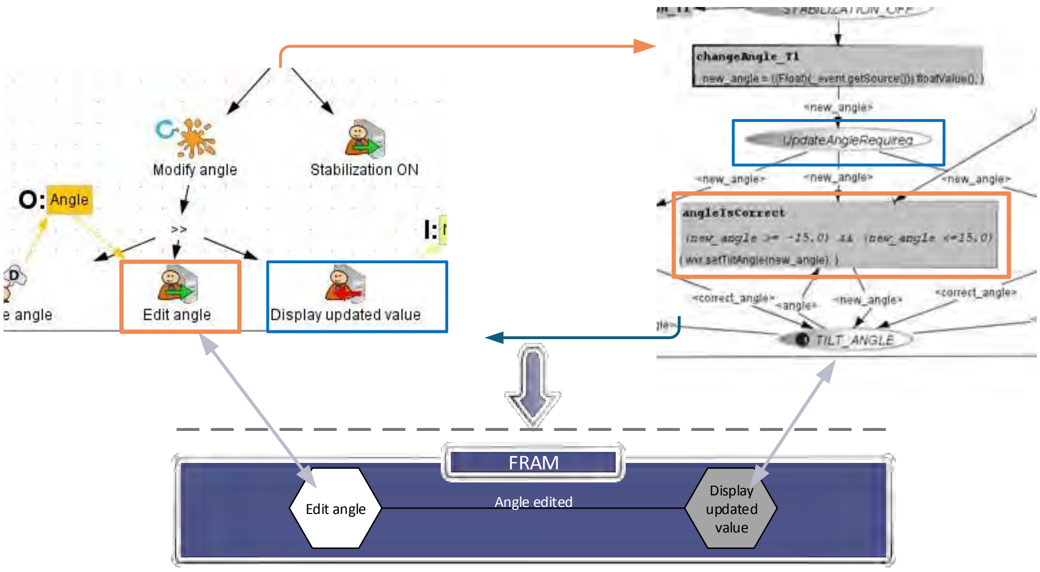


Figure 112: FRAM as correspondence editor in the WXR case study

For accomplishing this task, the pilot has to perform some consecutive tasks. One of these is to "Edit angle" (the orange box in the task model). This HAMSTERS 2.0 Interactive Input task has its correspondent in the PetShop model (the orange box in the system model). In FRAM, this correspondence is assessed through the output "Angle edited" of the function "Edit angle" which is the input for activating the "Display updated value" function. Consequently, this system function has its correspondent in the ICO model (the blue box in the system model) which has its correspondent in the HAMSTERS 2.0 model (the blue box in the task model).

In the WXR case study, all the correspondences have been assessed both at tool and also at model level. These assessments ensure that the models are consistent, coherent, and complete. Then, they have been integrated for properly representing the scenario and the related performance objectives. So, we can carry on to the Evaluation phase.

The Evaluation phase applied to the WXR case study

The Variability assessment sub-phase applied to the WXR case study

This phase starts with the step "Assess the variability" which consists of two complementary and parallel flows:

- "Perform the quantitative variability analysis" and
- "Perform the qualitative variability analysis".

The quantitative analysis flow provides an output which is the result of a refinement of the type of function variability. This can belong to human, to system and/or to interactive aspects. It results in a "Performance data of the scenario". To allow performance assessment we have to address timing issues at three levels: the operator side using the task models presented in Chapter 4 – Section 1, the system side exploiting the ICO behaviour models in Chapter 5 – Section 2.2, the interaction side related to the graphical interface described also in ICOs in Chapter 5 – Section 2.2.

With regard to the users' performance from task models, we first restricted the study to the interaction with the WXR interface (please refer to "Figure 52: Image of a) the weather radar control panel b) of the radar display manipulation" in Chapter 6 – Section 1). We have estimated the time it will take a user to point and click on a control of graphical interface. One of the evaluation approaches used in human factors domain is based on Fitts's law (Fitts, 1954) which is suitable for assessing motor movements. Fitts's law is presented in Formula (1) representing an index of difficulty for reaching a target (of a given size) from a given distance. Movement time (MT) for a user to access a target depends on width of the target (W) and the distance between the start point of the pointer and the centre of target (A).

$$MT = a + b\, log_2 \left(1 + \frac{2A}{W}\right) \quad (1)$$

For predicting movement time on the systems under consideration constants are set as follows: a=0 and b=100ms (mean value for users).

The following table presents the set of interactive widgets used within the weather radar control panel. For each widget, it provides a short name used for the following tables and the size used as the width for the Fitts's law (we use the minimum value between the width and the height to provide the assessment of the maximum difficulty to reach the considered widget).

TABLE 155: MIN WIDTH OF THE WIDGETS OF THE WXR USER INTERFACE

Interactive widgets	radio button off	radio button stdby	radio button tst	radio button wxon	radio button wxa	button Auto	button Manual	button ON	button OFF	text field angle
Short name	r1	r2	r3	r4	r5	b1	b2	b3	b4	t1
Min width (mm)	18	18	18	18	18	31	31	31	31	26

The following table provides the distances from the centre to each widget and between each widget. These distances are used to apply the Fitts's law when reaching a widget with a start point that can be the centre of the control panel or any widget.

TABLE 156: A) DISTANCE BETWEEN A WIDGET AND THE CONTROL PANEL CENTER IN THE WXR USER INTERFACE– B) TEMPORAL VALUES (IN MS) FOR USER INTERACTION USING FITTS'S

a)

Interactive widgets	r1	r2	r3	r4	r5	b1	b2	b3	b4	t1
Distance	104	130	115	100	87	77	17	128	104	132

b)

Interactive widgets	r1	r2	r3	r4	r5	b1	b2	b3	b4	t1
Temporal values (ms)	110	119	114	108	103	77	32	97	89	105

In addition to these motor values cognitive and perceptive values have to be used in order to cover all the elements of the task models. From (Card, Moran, & Newell, 1986) we know that the mean time for performing a comparison at the cognitive level is 100ms (ranging from 25ms to 170ms) while eye perception mean is 100mn too (ranging from 50ms to 200ms).

With regard to system models, in the ICO Petri net dialect, time is directly related to transition, which invokes services from the weather radar system (this is the case for transition off_T1 on Figure 109 which switches off the equipment). The duration of each invocation is presented in the following table (each value is coarse grain and depends on the type of weather radar). The 2000-4000ms value corresponds to the time required by the weather radar to scan the airspace in front of the aircraft (two or three scans are needed to get a reliable image).

TABLE 157: INTERACTION DELAYS INTRODUCED BY THE WXR APPLICATION

Model	Transition	Duration (ms)
WXR control panel model	Off_T1	500
	Stdby_T1	200
	Wxa_T1	500
	Wxon_T1	1000
	Tst_T1	1000
	angleIsLow	2000-4000
	angleIsCorrect	2000-4000
	angleIsHigh	2000-4000
Range selection model	mapIndexToRange	200

Using the HAMSTERS 2.0 task model in Figure 107 and the time estimations of the table above, we calculate that:

- the maximum time to configure the WXR interactive application for displaying weather conditions on the current route is 6688ms (change titl angle and change modes for ensuring that the WXR is properly work) while if s/he has to change 2 times the tilt angle the maximum time is 7376ms
- the minimum time to configure the WXR interactive application for displaying weather conditions on the current route is 4000ms (only test WXR appliacation),

In parallel to the quantitative analysis, we can perform the qualitative variability analysis. In particular, for the variability assessment of the usability performance objective (please refer to Chapter 6 – Section 1), an example of functional resonance can be between the two factors, time and precision, of the macro function "Check weather conditions". The results of the analysis are presented in a variability analysis table. This table synthesizes the downstream coupling between functions in order to identify functional resonance or dampening effects. Table 158 presents the variability analysis of the coupling factors of the function "Check weather conditions" and describes how they can have an impact on the output of the function, and consequently, on the input of the "Identify the adequate procedure" downstream function. Indeed, the output "Weather target recognised" of the "Check weather conditions" upstream function can have an impact on the input "Weather target recognised" of the "Identify the adequate procedure" downstream function with the three following different severity degrees:

- No impact (green box in the following table) which means that there is no impact on the output and on the downstream function,
- Medium impact (orange box in the following table) which means that the output presents a certain level of variability affecting the downstream function. However, this can be triggered by the output,
- High impact (red box in the following table) which means that the output presents a severe level of variability affecting the downstream function and this cannot be triggered.

TABLE 158: WXR CASE STUDY – TIME AND PRECISION COUPLING FACTORS OF "CHECK WEATHER CONDITIONS" MACRO FUNCTION

		Precision		
		Precise	Acceptable	Imprecise
Time	Too early	The pilot precisely recognises the weather target but too early. No impact on the output "Weather target recognised" and on the downstream function.	The pilot recognises the weather target in an acceptable manner but too early. No impact on the output "Weather target recognised" and on the downstream function.	The pilot partially recognises the weather target or doesn't pay appropriate attention to the recognition of the weather target. MEDIUM IMPACT (The adequate procedure could be applied because there is still time)
	On time	The pilot precisely recognises the weather target on time. No impact on the output "Weather target recognised" and on the downstream function.	The pilot recognises the weather target in an acceptable manner but on time. No impact on the output "Weather target recognised" and on the downstream function.	The pilot partially recognises the weather target or doesn't pay appropriate attention to the recognition of the weather target. MEDIUM IMPACT (The adequate procedure could be applied because s/he is on time).
	Too late	The pilot precisely recognises the weather target but too late. MEDIUM IMPACT (The adequate procedure could be still applied)	The pilot recognises the weather target in an acceptable manner but too late. HIGH IMPACT (The adequate procedure cannot be applied)	The pilot partially recognises the weather target or doesn't pay appropriate attention to the recognition of the weather target. HIGH IMPACT (The adequate procedure cannot be applied)

Moreover, these three different severity degrees can be represented in the FRAM instantiation as:

- No impact: the FRAM instantiation remains the same (as illustrated in "Figure 103: Scenario – FRAM instantiation of the WXR case study")
- Medium impact: the affected function is marked with an orange wave representing this level of severity but the downstream function can be active (as illustrated in Figure 113),
- High impact: a red cross and a dotted line indicate that the output and the downstream function do not exist anymore (as illustrated in Figure 114).

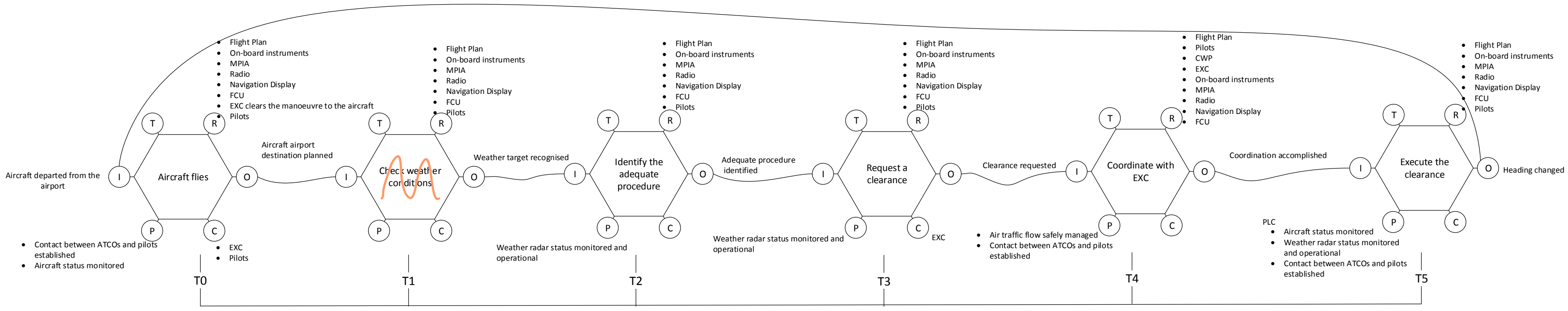


Figure 113: Scenario - Medium impact of “Check weather conditions” macro function coupling factors – FRAM instantiation of the WXR case study

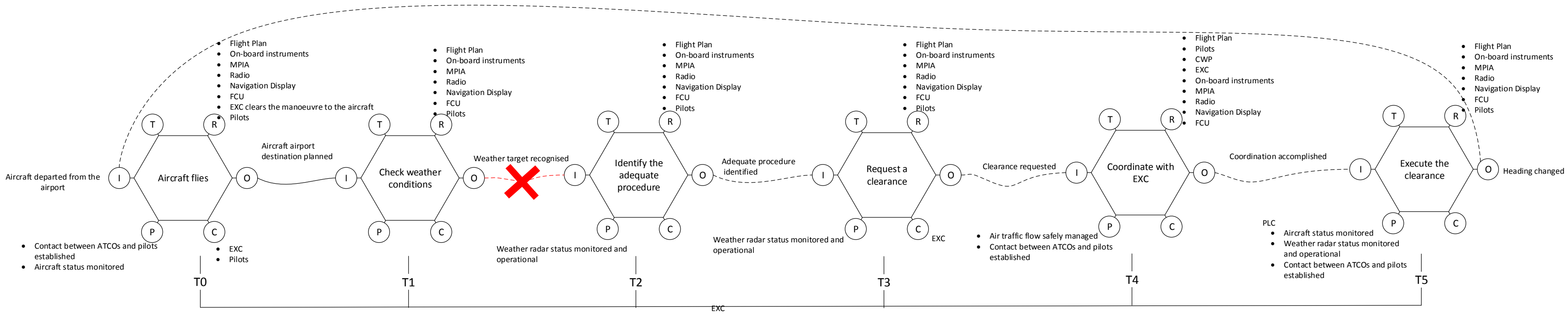


Figure 114: Scenario - High impact of “Check weather conditions” macro function coupling factors – FRAM instantiation of the WXR case study

With regard to the variability assessment of the resilience performance objective (please refer to Chapter 6 – Section 1), an example of functional resonance can be between the two factors, time and precision, of the function “Coordinate with EXC”. However, according to the scenario description, the pilot request is immediately accepted by the EXC and the functional resonance of the “Coordinate with EXC” function cannot be properly assessed because there is any sourroundig traffic or weather issue to be taken into account. So, this scenario is not coherent with the resilience performace objective.