Fine Grain Modeling of Task Deviations for Assessing Qualitatively the Impact of Both System Failures and Human Error on Operator Performance

Célia Martinie, Philippe Palanque
ICS-IRIT, Université Paul Sabatier Toulouse 3, 118, route de Narbonne, 31062 Toulouse Cedex 9, France
{martinie, palanque}@irit.fr

Abstract

Operators of critical interactive systems are trained and qualified before being allowed to operate critical systems in "real" contexts. However, during operation, things might happen differently from during training sessions as system failures may occur and operators may make errors when interacting with the system. Both events may also be cross-related as a misunderstanding of a system failure can lead to an erroneous subsequent operation.

The proposed approach focuses on assessing the impact that potential failures and/or human errors may have on human performance. This analysis targets the design and development phases of the system, when user tasks are analyzed in order to build the right system (i.e. corresponding to the users' needs and activities they have to perform on the system). We use a task modeling notation for describing precisely operators' activities as well as information, knowledge and objects required for performing these activities. These task models are then augmented into several variants through integration of potential system failure patterns (with associated recovery tasks) and human error patterns. The produced deviated task models are used to assess the impact of the task deviation on the operators' performance.

Introduction

Design and development of interactive critical systems require methods to account for their dependability. Several aspects of dependability have to be addressed: dependability of the system, dependability of the user and dependability of the interaction between the system and the user. In this paper we focus on management of system failures and human error at runtime. Even if systems have been designed and developed with dependability in mind, failures may occur. Even if operators are trained before being allowed to operate critical systems, they may make errors when interacting with the system. We propose a notation and associated CASE tool to provide support for analysis of the impact of system failures and human error on human performance. First section presents an overview of classifications of system failures and human errors. Second section presents the proposed task modeling notation used to describe human activities and potential human errors that may be made during the planned activities. Last section discusses about how these task models can be used to assess the impact of failures and human errors on human activities and on the system. The proposed approach is exemplified all along the sections with a case study from the space satellite ground segments domain. This case study belongs to the category of complex command and control systems from the space domain. Such interactive systems are less time constrained than other ones (such as aircraft cockpits). These systems are less safety critical (the only possible safety issue would correspond to a spacecraft falling on earth and injuring people). However, the potential cost of a failure is far beyond the development cost of these systems making them belong to the category of critical systems.

This paper first reviews how system failures and human errors can be taken into account. Second part is dedicated to task modeling and integration of human errors in task models. Last part describes how the deviated task models can be used to assess the impact of failure or human errors on human activities and system in the context of a satellite ground segment.

Accounting for System Failures and Human Errors

In the area of dependable systems such issues have been looked at and current state of the art in the field identifies five different ways to increase a system's reliability (Avizienis et al., 2004) and (Bowen and Stavridou, 1993):

- *Fault avoidance*: preventing the occurrence of faults by construction (usually by using formal description techniques and proving safety and liveness properties (Pnueli, 1986)).
- *Fault removal*: reducing the number of faults that can occur (by verification of properties).
- *Fault forecasting*: estimating the number, future incidence and likely consequences of faults (usually by

statistical evaluation of the occurrence and consequences of faults).

- *Fault tolerance*: avoiding service failure in the presence of faults (usually by adding redundancy, multiple versions and voting mechanisms).
- *Fault mitigation*: reducing the severity of faults (by adding barriers or healing behaviors (Neema et al., 2004)).

Fault avoidance can be attained by the formal specification of the interactive system behavior provided all the aspects of interactive systems are accounted for including device drivers' behaviors, graphical rendering and events handling and a Petri net based approach dealing with these aspects can be found here (Navarre et al., 2009).

However, due to this software/hardware integration faults might occur at runtime regardless the effort deployed during design phases. To increase the system reliability concerning runtime faults, we have previously proposed (Tankeu-Choitat et al., 2011) ways to address both fault tolerance and fault mitigation for safety critical interactive systems, while fault recovery was addressed through interaction reconfiguration as described in (Navarre et al., 2008). While fault tolerance and fault mitigation can be seen as rather different they require the deployment of the same underlying mechanisms:

- *Fault detection*: identifying the presence of faults, the type of the fault and possibly its source,
- *Fault recovery*: transforming the system state that contains one or more faults into a state without fault.

Of course, the training program must deal with these adverse events and prepare the user to be able to deal with them in a dependable and timely manner. However, as aforementioned some autonomous mechanisms can be defined and deployed leaving most of the faults un-notified to the operator. However, the operator may also make errors.

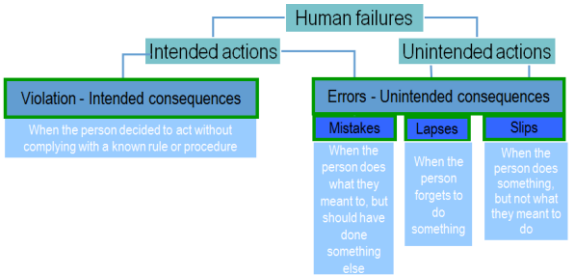


Figure 1. Overview of Human Errors

Several taxonomies of human errors have been proposed (Reason, 1990; Hollnagel, 1998) and Figure 1 depicts a summary of these errors. In order to mitigate human errors, several actions can be taken:

- Error detection or notice: identifying the presence of errors, the type of the error and possibly its root cause. One of the key elements here is to detect errors even though they have no impact on operations as this might be due to contingencies that might lead to incidents or accidents in another context.
- Error prevention: reducing the potential number of occurrences by designing adequate training, designing affording products and designing usable system
- Error protection: reducing the impact of an error by including barriers in the design and duplicating operators.

While the same applies to human error and system failures current methods, techniques and tools address them independently and promote different treatment.

Next section presents how these failures and errors can be described when analyzing user activities during system design proposing a unified process for both.

Modeling Operator Tasks with Having System Failures and Human Errors in Mind

Task models are a mean to gather and structure data from the analysis of users' activities. They aim at recording, refining and analyzing information about users' activities. Several notations are available and provide various formats to describe tasks and having various expressiveness levels depending on targeted analysis, one of the most famous being CTT (Mori et al. 2002). This section briefly introduces HAMSTERS, the task modeling notation and its associated CASE tool used for assessing impact of system failures and human errors on human performance.

HAMSTERS Notation and Tool

HAMSTERS is a tool-supported graphical task modeling 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. Output of this decomposition is a graphical tree of nodes. Nodes can be tasks or temporal operators.

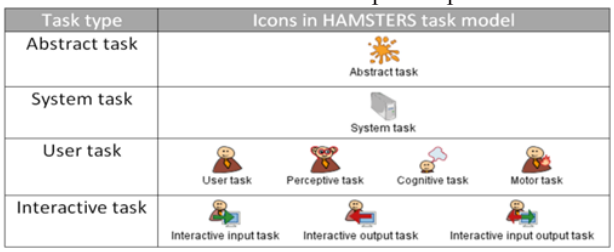


Figure 2. High-level Task Types in HAMSTERS

Tasks can be of several types (as illustrated in Figure 2) and contain information such as a name, information details, critical level… Only the high-level task type are presented here (due to space constraints) but they are further refined (for instance the cognitive tasks can be refined in Analysis and Decision tasks (Martinie, Palanque, Ragosta, Barboni, 2011).

Temporal operators are used to represent temporal relationships between sub-goals and between activities (as detailed in Table 1). Tasks can also be tagged by temporal properties to indicate whether or not they are iterative,

optional or both. Composition and structuration mechanisms provide support for description of large amounts of activities (Martinie, Palanque, Winckler, 2011).

Table 1. Temporal Ordering Operators in HAMSTERS

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

Explicit and systematic integration of object, information and knowledge provides support for description of required objects, information, declarative knowledge and procedural knowledge required to accomplish the tasks (Martinie et al., 2013).

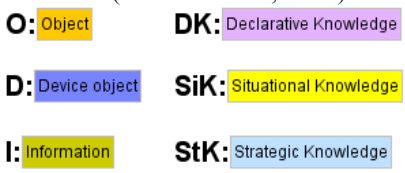


Figure 3. Representation of Objects, Information and Knowledge with HAMSTERS Notation

Figure 3 presents the notation elements for objects, input and output device objects, information used by users to perform the tasks, and knowledge required to perform the tasks. Figure 4 shows an extract (high level tasks) of the HAMSTERS task model of PICARD satellite platform management. Refined models that include low-level routine activities (such as the ones depicted in Figure 6) are not included here due to space constraints.

Modeling Failure Detection and Recovery

Once planned activities have been described, we propose to build task models for failure and detection recovery activities that may take place after a system failure. In this way, it is possible to ensure that the interactive system provide support for this type of activities but also to en sure that the operators will be trained (Martinie et al., 2011) for this type of adverse events. A new version of the high level task model has been produced (presented in Figure 4). A "Failure detection and recovery" branch has been added ("Detect and recover from failure" sub-goal concurrent to the "Monitor satellite parameters" and "Handle routine activities" sub-goals). This "Failure detection and recovery" sub-goal has then been refined in several task models to describe the activities that the user may have to lead depending of the various types of failures that may occur. For the rest of the paper, we will take the example of a failing Sun Array Driver Assembly (SADA). If this appliance fails, operators have to detect it and try to switch ON the redundant SADA.

An excerpt of this activity is depicted in Figure 5.a). The operator has to select and launch a procedure through the ground segment application (interactive input tasks "Select procedure Switch ON SADA2" and "Start procedure "Switch ON SADA2").Operator has then to wait for the system to trigger the rotation of the redundant SADA and wait for a message from the system asking if the rotation has to be stopped (interactive output task "Display rotation stop message").

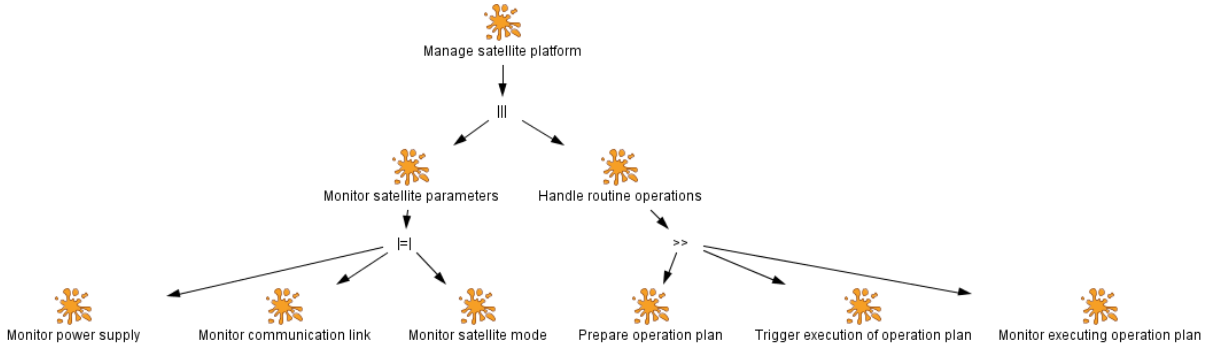


Figure 4. HAMSTERS Task Model of PICARD Satellite Platform Management (High Level Tasks)

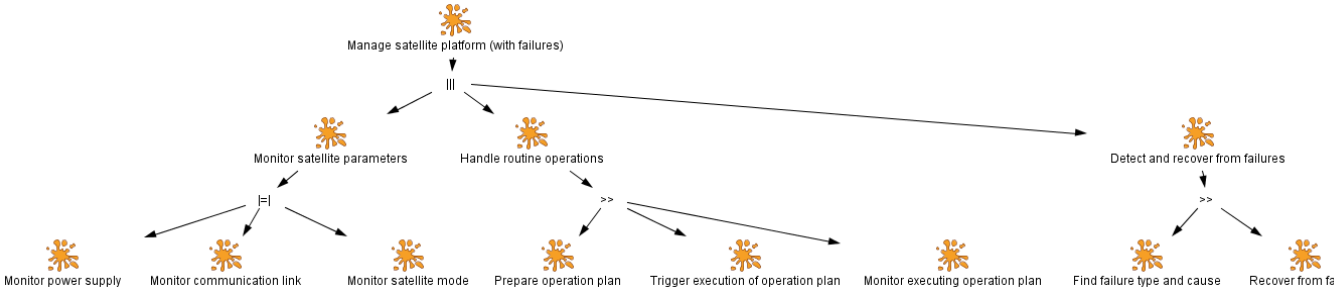
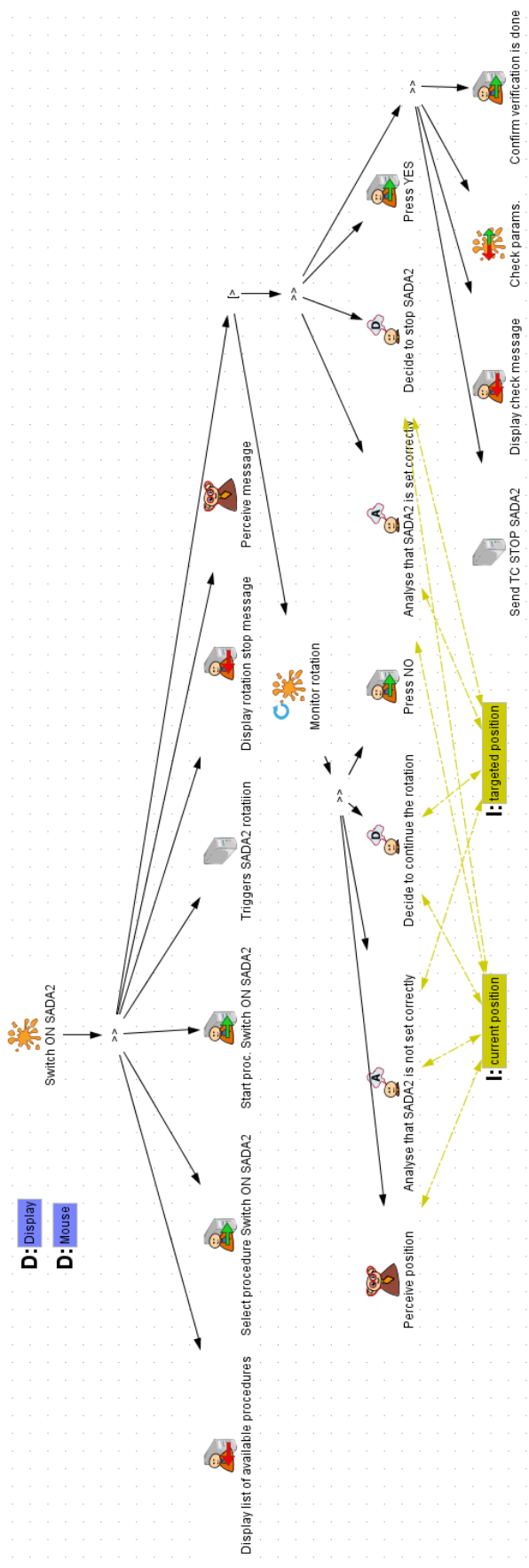


Figure 5. HAMSTERS Task Model of PICARD Satellite Platform Management (Including Failures)



a) b)

Figure 6. Task Model of "Switch ON SADA2" Procedure a) without Human Error b) with Human Error

Modeling Human Error

Operators have then to check parameters in order to identify the current position (human perceptive task "Perceive position" which output the information "current position") of the SADA2 and whether or not it is compliant to the targeted position (human cognitive analysis tasks "Analyze that SADA2 position is set correctly" and "Analyze that SADA2 position is not set correctly" which need "current position" and "targeted position" as inputs). Once operators have made a decision (human cognitive decision tasks "Decide to…"), they press the corresponding button on the user interface associated with the interactive input tasks "Press YES" and "Press NO".

Human errors can then be integrated into the tasks models for ensuring system robustness. In that perspective, erroneous behavior described in task models can be used to evaluate the impact of a human error on the system as proposed by (Bass and Bolton, 2013). Also in that perspective, mutant task specifications can be used to analyze the ability of the system to remain safe if a user performs deviated tasks on the system (Yasmeen and Gunter, 2011). Human errors can also be integrated into task models in order evaluate the usability of the system (Paterno and Santoro, 2002) and to inform design. In current paper, we focus on human performance and on task recovery whether the deviated task is performed upon system failure and/or upon human error. We propose to extend the work done by (Palanque & Basnyat, 2004) who proposed a Task Analysis for Error Identification technique. This technique can be used to identify potential human errors during routine activities as well as during failure detection and recovery activities. In this example, we focus on one type of error but an example of a complete case study of task analysis for error identification can be found in (Palanque & Basnyat, 2004). However, in that earlier work, information, devices and objects required to perform a task were not represented in the task models. Thus, it did not provide support to assess performance at the information level. This made impossible to reason about workload aspect of operator performance. In our presented case study, human errors can occur while accomplishing the procedure to setup the redundant Sun Array Driver Assembly. For example, Figure 5 b) presents the task model of erroneous actions performed by the operator in that case. From Reason's classification, an associative-activation error (Reason, 1990) can occur if an operator clicks on "YES" while s/he had decided not to confirm stop or if s/he clicks on "NO" while s/he had decided to confirm stop. This error implies that the operators will have additional tasks to perform in order to reach the goal of switching to the redundant SADA. These additional tasks are presented in detail in Figure 6. Operators will have to understand that the rotation did not stop in spite of the fact he/she wanted it to stop (cognitive analysis task "Analyze the rotation did not stop" in Figure 6). They will then have to wait for the SADA2 position to become correct and for the next confirmation message in order to be able to terminate the procedure.

Assessment of the Impact of Failures or Human Errors on Human Performance

Task models of failure detection and recovery as well as task models integrating potential human errors and their impact on the operators' activities provide support for establishing requirements on the future system but also for evaluating the impact of failures and errors on the global human performance and on the mission execution. In our example, we can see that in the case of a human error while switching to the redundant SADA:

- At least 9 more activities will have to be performed. As "Monitor position task" is iterative (round shaped arrow on the left side of the task widget), operators might have to examine several times the position before it becomes correct.
- Operators will have to cognitively handle more information and during a longer period of time ("current position" and "targeted position" information objects in Figure 5.b)) than without the interference error.

In this example, given the decision to switch on the redundant SADA, the satellite is maybe currently in a low

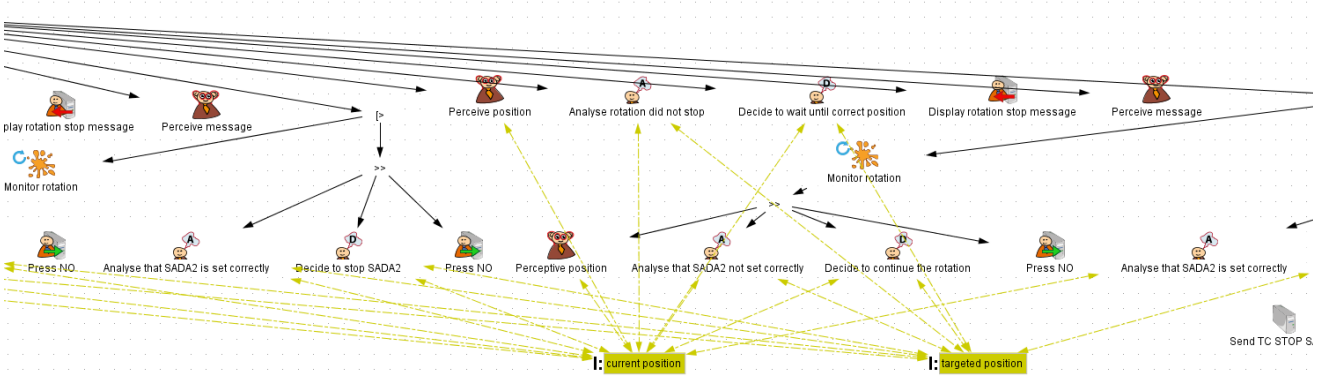


Figure 4. Zoom on Tasks Inserted for Error and Recovery from Error (from Figure 3)b))

power state (survival mode). This means that energy has to be spared and redundant SADA set as soon as possible. This analysis highlights that a human error during this recovery task could be fatal for the mission.

Conclusion

The proposed approach combining operators' tasks modeling with operators' errors information provides support for assessment of the articulatory activities the operators will have to perform in order to recover from system failure. This can be applied to routine activities, failure detection and recovery activities or human error detection and recovery activities. By making explicit the tasks, the information and the objects that have to be handled by the operators, this approach enables assessing the recovery cost from a system failure (i.e. to set the system in an acceptable state) but also from a human error i.e. performing a set of corrective actions in order to, as for a system failure, set the system to an acceptable state.

This short presentation of the approach has not made it possible to exemplify a set of other benefits that become reachable using such task models enhanced with human error descriptions. For instance:

- Some of the information explicitly represented in the task model might correspond to information that has to be stored in the operator's working memory (e.g. a flight level clearance received by a pilot from an air traffic controller). The modeling approach would make explicit how much time (quantitative) but also how many actions have to be performed while keeping in mind such information.
- The tasks and the related information might be located on specific devices. This is not the case for a space ground where monitoring is co-located with telecommands triggering, but the possibility to represent that information in HAMSTERS enables to assess low-level complexity of tasks such as device localization, moving attention and activity from one device to another one…

The presented analysis is performed informally and manually but HAMSTERS models edition and simulation are supported by the eponym tool. Performance analysis functionalities are currently being integrated exploiting contributions previously made for synergistic system-task execution (Barboni et al., 2011) and training program assessment (Martinie et al., 2011).

References

Avizienis, A., Laprie, J.-C., Randell, B., Landwehr, C. 2004. Basic concepts and taxonomy of dependable and secure computing. In IEEE Trans. on Dependable and Secure Computing, vol.1, no.1, pp. 11- 33.

Barboni, E., Ladry J.F., Navarre, D., Palanque, P., Winckler, M. 2010. Beyond modelling: an integrated environment supporting co-execution of tasks and systems models. ACM SIGCHI EICS, 165-174.

Bolton, M. L., Bass, E. J. 2013. Generating Erroneous Human Behavior From Strategic Knowledge in Task Models and Evaluating Its Impact on System Safety With Model Checking. IEEE Trans. on Systems, Man, and Cybernetics, vol.43, n. 6.

Bowen J. and Stavridou V. 1993. Formal Methods, Safety-Critical Systems and Standards. Software Engineering Journal, 8(4):189–209.

European Cooperation for Space Standardization, Space Engineering, Ground Systems and Operations. 2008. ECSS-E-70C & ECSS-E-70B Draft 4.2.

Hollnagel, E. 1998. Cognitive Reliability and Error Analysis Method. Elsevier Science, Oxford.

Martinie C., Palanque P., Ragosta M, Barboni E. 2011. Task-Model Based Assessment of Automation Levels: Application to Space Ground Segments. IEEE int.conf . on Systems, Man and Cybernetics, IEEE. pp. 3267-3273.

Martinie C., Palanque, P., Ragosta, M., Fahssi, R. 2013. Extending Procedural Task Models by Explicit and Systematic Integration of Objects, Knowledge and Information. Europ. Conf. on Cognitive Ergonomics, 23-33.

Martinie C. et al. 2011. Model-Based Training: An Approach Supporting Operability of Critical Interactive Systems: Application to Satellite Ground Segments. ACM SIGCHI EICS 2011, 53-62.

Martinie C., Palanque P., Winckler M. 2011. Structuring and Composition Mechanism to Address Scalability Issues in Task Models. IFIP TC13 Conference on Human-Computer Interaction (INTERACT 2011), Springer LNCS 6948.

Mori G., Paternò F. & Santoro C: CTTE: Support for Developing and Analyzing Task Models for Interactive System Design. IEEE Trans. Software Eng. 28(8): 797-813 (2002)

Navarre D., Palanque P., Ladry J-F., Barboni E: ICOs: A model-based user interface description technique dedicated to interactive systems addressing usability, reliability and scalability. ACM Trans. Computer-Human Interactions 16(4) (2009)

Navarre D., Palanque P. & Sandra Basnyat: A Formal Approach for User Interaction Reconfiguration of Safety Critical Interactive Systems. SAFECOMP 2008: 373-386

Neema S., Bapty T., Shetty S. & Nordstrom S. 2004. Autonomic fault mitigation in embedded systems. Eng. Appl. Artif. Intell. 17, 7, 711-725.

Palanque, P., & Basnyat, S. Task Patterns for Taking Into Account in an Efficient and Systematic Way Both Standard and Erroneous User Behaviours. HESSD 2004, pp. 109-130.

Paterno, F., Santoro, C. 2002. Preventing user errors by systematic analysis of deviations from the system task model. Int. J. on Human-Computer Studies, Elsevier, vol. 56, n. 2, 225-245.

Pnueli A. 1986. Applications of Temporal Logic to the Specification and Verification of Reactive Systems: A Survey of Current Trends. LNCS n° 224, p.510-584. Springer Verlag.

Reason J. 1990. Human Error, Cambridge University Press.

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, US, IEEE.

Yasmeen, A., Gunter, E. L. 2011. Robustness For Protection Envelopes with Respect to Human Task Variation. Int. Conference on Systems, Man and Cybernetics, p. 1809-1816.