HAL
open science

A User-Centered View on Formal Methods: Interactive Support for Validation and Verification

Eric Barboni, Arnaud Hamon, Célia Martinie, Philippe Palanque

To cite this version:

Eric Barboni, Arnaud Hamon, Célia Martinie, Philippe Palanque. A User-Centered View on Formal Methods: Interactive Support for Validation and Verification. Workshop on Formal Methods in Human Computer Interaction (FoMHCI 2015), Jun 2015, Duisburg, Germany. pp. 24-29. hal-01334684

HAL Id: hal-01334684

https://hal.science/hal-01334684v1

Submitted on 21 Jun 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Open Archive TOULOUSE Archive Ouverte (OATAO)
OATAO is an open access repository that collects the work of Toulouse researchers and makes it freely available over the web where possible.

This is an author-deposited version published in : http://oatao.univ-toulouse.fr/
Eprints ID : 15405

The contribution was presented at FoMHCI 2015 :
https://sites.google.com/site/wsfomchi

To cite this version : Barboni, Eric and Hamon, Arnaud and Martinie De Almeida, Celia and Palanque, Philippe *A User-Centered View on Formal Methods: Interactive Support for Validation and Verification.* (2015) In: Workshop on Formal Methods in Human Computer Interaction (FoMHCI 2015), 23 June 2015 - 23 June 2015

Any correspondence concerning this service should be sent to the repository administrator: staff-oatao@listes-diff.inp-toulouse.fr

A User-Centered View on Formal Methods: Interactive Support for Validation and Verification

Eric Barboni, Arnaud Hamon, Célia Martinie & Philippe Palanque

ICS-IRIT, University of Toulouse,
118 Route de Narbonne,
F-31062, Toulouse, France
{Name}@ irit.fr

ABSTRACT

During early phases of the development of an interactive system, future system properties are identified (through interaction with end users e.g. in the brainstorming and prototyping phases of the development process, or by requirements provided by other stakeholders) imposing requirements on the final system. Some of these properties rely on informal aspects of the system (e.g. satisfaction of users) and can be checked by questionnaires, while other ones require the use of formal methods. Whether these properties are specific to the application under development or generic to a class of applications, the verification of the presence of these properties in the system under construction usually involve verification tools to process the formal description of the system. The usability [26] of these tools has a significant impact on the V&V phases which usually remains perceived as very resource consuming. This position paper proposes the application of action theory to identify complex aspects of verification and exploits it for identifying areas of improvement.

Author Keywords

Formal methods; interactive systems; object oriented Petri nets; Analysis.

INTRODUCTION

Nowadays interactive applications are deployed in more and more complex command and control systems including safety critical ones. Dedicated formalisms, processes and tools are thus required to bring together various properties such as reliability, dependability and operability. In addition to standard properties of computer systems (such as safety or liveness), interaction properties have been identified. Properties related to the usage of an interactive system are called external properties [6] and characterize the capacity of the system to provide support for its users to accomplish their tasks and goals, and prevent or help to recover from errors. Although all types of properties are not always completely independent one from each other (any might be conflicting), external properties are related to the user's point of view and usability factor, whereas internal properties are related to the design and development process of the system itself (modifiability, run time efficiency). Interactive systems have to support both types of properties and dedicated techniques and approaches have been studied for this purpose, amongst them are formal methods. Formal languages have proven their value in several domains and are a necessary condition to understand, design, develop systems and check their properties.

Formal methods have been studied within the field of HCI as a means to analyze in a complete and unambiguous way interactions between a user and a system. Several types of approaches have been developed [9], which encompass contributions about formal description of an interactive system and/or formal verification of its properties. Other approaches such as [6] exploit formal methods for understanding interactive systems and provide a better description of their specificities.

The use of formal methods depends on the phase of the development process describing the activities necessary to develop the interactive system under consideration. In the case of critical interactive systems, [16] proposes a development process relying on formal methods and taking into account both interactive and critical aspects of such systems which necessitates several formal descriptions used by different user types (system designers, human factor specialists…). Consequently, the usability of tools for validation and verification of these interactive systems target will depend on their users' activity in the development process.

Whether being used as a means for describing in a complete and unambiguous way the interactive system or as means for verifying properties, formal description techniques and their associated tools need to be designed to be usable and not error prone.

The paper proposes a user–centered view on the use of formal methods. We take as a running example the ICO notation [18] and its associated tool Petshop [2] and use Norman's action theory [19] as a framework for identifying usability gaps and error sources. We believe however that this framework would be applicable to other notations and tools and hope to discuss this during the workshop.

VALIDATION AND VERIFICATION TOOLS FOR THE DEVELOPMENT OF INTERACTIVE CRITICAL? SYSTEMS

Norman's action theory is quickly presented in Figure 1. Its application to formal modelling is presented in Figure 2.

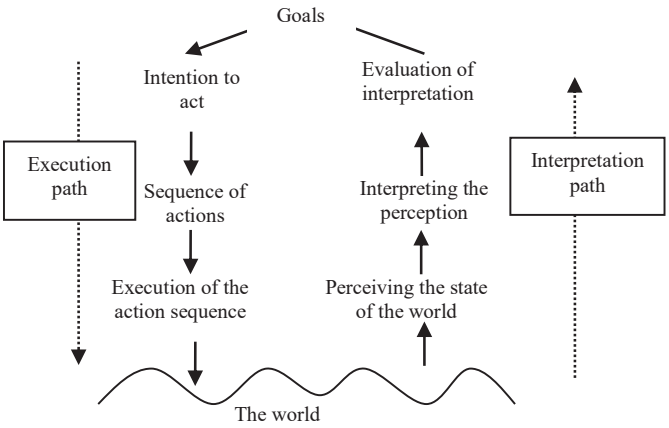

Figure 1. The seven stages of the action theory

That figure shows the refinement of specific activities to take into account modelling activities. The two main stages we are focusing on are:

- On the execution side, the activity of going from an intention to its actual transcription into some model,
- On the perception side, the activity of perceiving model behavior and interpreting this perception.

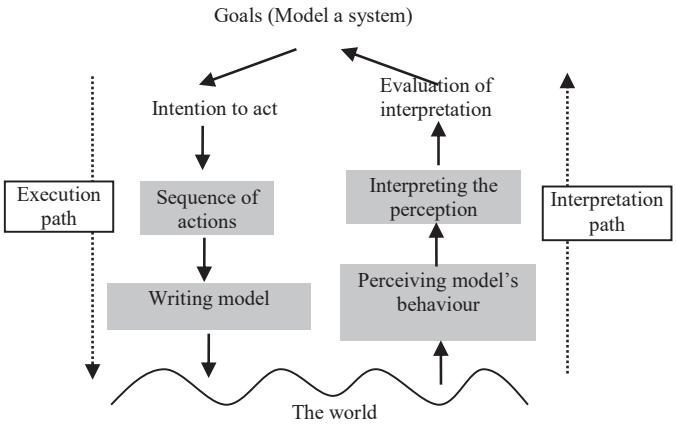

Figure 2. Formal modelling within the action theory

User tasks when validating and verifying interactive critical systems

We have identified three main tasks when verifying formal model, and Petri nets (which is the underlying formalism of ICOs) in particular: analysis, simulation in conjunction of the edition task the formers reflecting the verifiability [11] and the latter the executability [8] aspects.

Simulation is the task where users (people building the formal model) have to check that the model built exhibits the expected behavior. The interpretation task can be eased if the state changes in the model are shown in an animated way thus reducing the users' activity of comparing the current state with the previous one.

For analysis related tasks, users check the validity of some properties hold on the model. One of the issues is then to understand the analysis result (for instance one place in not in any "P invariant") in Petri net tools and then to map this analysis result with the goal (was this place meant to be unbounded?).

Existing tools for validation and verification of interactive critical systems

Verification and validation of formal models can be divided in two categories, whether they rely on static analysis or on simulation (step by step, interactive…).

System models

A representative classification of User Interface Description Languages (UIDLs) have shown only few frameworks describing the interactive system behavior and providing support for analysis [10].

Among these, Marigold [24] addresses limited validation and verification analysis based on reachability graph analysis and allows exporting to the Integrated Net Analyzer tool [22] offering other analysis capabilities. Proton++ [14] only provides static analysis as the tool only handles gesture conflict detection between models at compilation time. ICO [18] provides support for validation and verification but only through invariant analysis (Place/Transition invariants) provided by the Petshop tool as detailed in the tool description section.

Finally, the colored Petri nets (CPN) approach is more complete in terms of analysis enabling validation, verification and performance analysis accomplished by all different types of analysis techniques (e.g. reachability analysis) except invariants. The analysis of CPN models is supported by the CPN Tools [13].

In addition, both CPN and PetShop enable simulation of the models they produce. However, CPN does not connect the model to the user interface of the application requiring an additional step when interactive application are concerned, forcing the system designers .

Tasks models

CTT [21] proposed an approach based on formal model-checking (with CADP[1] toolset) of LOTOS [12] specifications of tasks between the user and the system. The HAMSTERS tool [15] (supporting the HAMSTERS notation) also provides static analysis support via basic check on the model's structure.

However, here again, tasks have to be checked on the user interface and simulation should integrate execution on the user interface. If this is not the case then a gulf exists between the task model and its simulation and the actual behavior of the interactive application.

Tools Usability

Building or modifying system and task models belongs to the type of human activities that is highly demanding on the user's side. Figure 2 provides a generic framework for investigating where the main difficulties can occur, and thus to provide design rules for environments to support user's activities and reduce difficulties.

In order to increase usability during validation and verification phases, tools should provide "continuous and permanent feedback", "modeless support to users' tasks", "reversibility of actions (undoing actions)" and taking into account the different formalisms' characteristics. This dimension becomes particularly important when considering large and complex interactive systems for which both user and system models are used jointly.

CIRCUS: A CASE TOOL FOR FORMAL ANALYSIS AND DESCRIPTION OF INTERACTIVE SYSTEMS

In the following section, we illustrate the validation and verification of interactive system using formal methods with the Circus suite which combines task models in HAMSTERS and high level Petri nets in

Notations supported by the tool suite

The ICO notation

The ICO notation [18] (Interactive Cooperative Objects) is a formal description technique devoted to specify interactive systems. Using high-level Petri nets for dynamic behavior description, the notation also relies on object-oriented approach (dynamic instantiation, classification, encapsulation, inheritance and client/server relationships) to describe the structural or static aspects of systems.

This notation have been applied to formally specify interactive systems in the fields of Air Traffic Management [18], satellite ground systems [20] and cockpits of military [4] and civil [1] aircrafts.

[1] http://cadp.inria.fr/

The HAMSTERS' tool tasks notation

HAMSTERS features a task model notation that enables structuring users 'goals and sub-goals into a hierarchical tasks tree in which qualitative temporal relationship amongst tasks are described by operators [17]. Goals or sub-goals are modeled using the type of task called "abstract". An abstract task can be refined in 3 types of tasks: "user task", "system tasks" and "interactive tasks". A "user task" can be refined in the following sub-types: "perceptive task", "cognitive task" and "motor task". An interactive task can be refined in the following sub-types: "input task" and "output task".

In addition, [25] extended the notation to integrate objects, knowledge and information; thus describing the exchanges between users and the systems they interact with.

Tool description

The following sub-sections describe both tools (system and task models' tool) which are merged into a single framework called Circus as introduced in [2].

Though one recommendation is to provide modeless tools, PetShop combines these three modes and allow the users direct visualization of the structural analysis results while editing and simulating the Petri nets' models. These analysis mode can be activated without stopping either the edition or the simulation. In addition, a dedicated panel presents the incidence matrix and the P/T invariants, which allow the identification of potential deadlock in the system models.

The HAMSTERS tool [15] is the part of the framework which enable task model editing and analysis.
The Circus tool provides a framework for both system and task models. It enables the co-execution of the system with the corresponding user's task model [2] which corresponds to executability related tasks. Regarding the verifiability, the tool provides a checking mechanism to ensure tasks and system models are compatible.

Tool usability

The Circus tool targets engineers, system designers and human factors specialist and helps them achieve their specific tasks while developing interactive critical systems. Their objective is to design and develop usable, reliable and dependable applications for these critical systems. It encompasses formal verification of the system's behavior as well as its compatibility with the system's targeted users.
The Circus tool development has been, so far, oriented towards enhancing one particular dimension of usability: effectiveness. We developed new capabilities so the tools' users can achieved their goals by being able to complete their tasks during most phases of the development process. Therefore, the tool enables the analyst to formally validate both system and tasks models and ensure they match so that the end-users will be able to perform their tasks on the system described. In addition, the tool allows the assessment of

the impact of dependability on usability on the considered interactive critical systems as well as exhibiting design choices and trade-offs in (potentially) conflicting user interface guidelines. Finally Circus provides support for usability testing by providing execution logs at the model level so system designers are able to match the potential issues with the model's nodes and opens the way to user performance measurement.

Although the various Circus user types are identified, we did not measure their satisfaction and analyze improvements to be made with respect to their user profiles and the related tool functionalities. The evaluation of users' efficiency during verification and validation is also identified for future work with performance measures among others.

ILLUSTRATIVE EXAMPLE: WXR APPLICATION

This section illustrates the use of the Circus tool which enables the formal description of interactive critical system as a socio-technical system; and which supports is validation and verification.

Case study description

Weather radar (WXR) is an application currently deployed in many cockpits of commercial aircrafts. 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 3 - Image of a) the weather radar control panel b) of the radar display manipulation

Figure 3 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.

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 3 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 4 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 and size of the clouds.

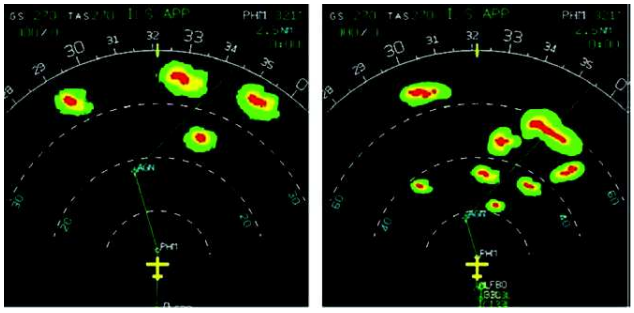

Figure 4 - Screenshot of weather radar displays

Formal modelling of the application's behavior using ICO

The first model presented here describes how it is possible to handle the weather radar configuration of both its mode and its tilt angle.

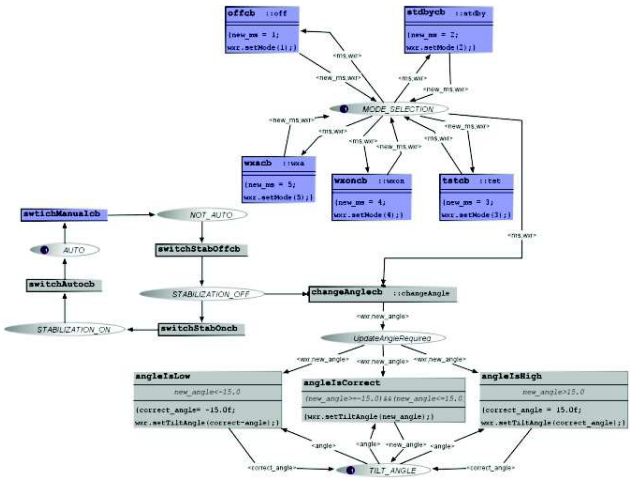

Figure 5 - Behavior of the WXR mode selection and tilt angle setting

Figure 3 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 checkup, STDBY to switch it on for test only and WXA to focus detection on alerts).

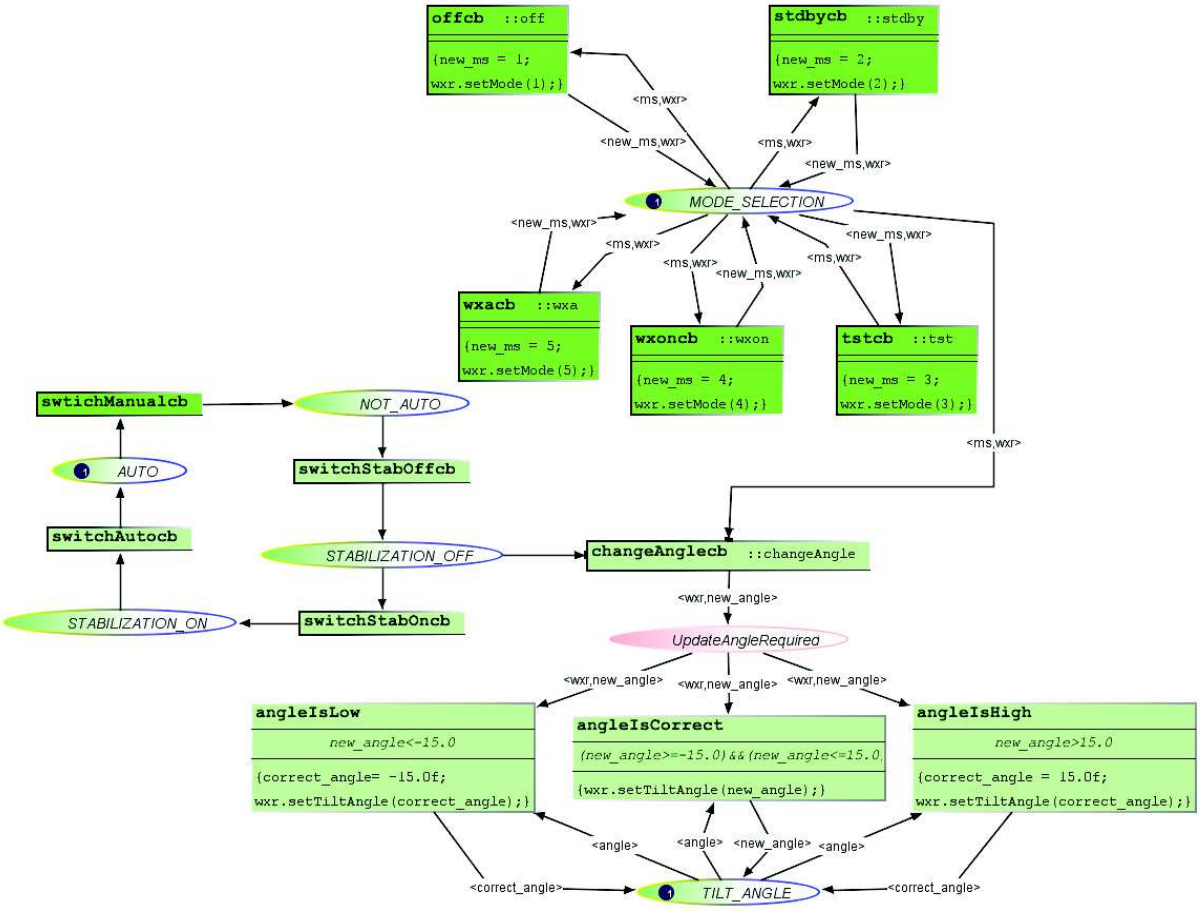

Figure 6 - Representation of invariants with the Analysis feature of Petshop CASE tool

- 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.

The corresponding task model is not presented in this paper but is described in [15].

Illustration of verification and validation tasks to be tool supported

In this section we emphasis on the validation aspect of the system model presented in Figure 5. When activated, the static analysis mode of PetShop displays a dedicated panel we do not present in this paper but which results are also displayed in the main edition view as shown Figure 6. The green overlay on the places and transitions identifies the node part of the model's invariants otherwise the red overlay is used as for the place UpdateAngleRequired. Places with yellow borders are syphons whereas taps use a blue stroke.

In order to determine which nodes belong to the same invariant as another node, a model pop-up menu is provided, taking over the standard pop-up menu for edition. This modal pop-up menu can be switch from analysis mode to normal edition mode using a dedicated icon on the toolbar.

The main current limitation of the tool regarding the static analysis lies in the fact that the PetShop algorithms do not close opened models (models that provided services to other models).

CONCLUSION AND PERSPECTIVES

This position paper has presented the use of a generic model in the field of HCI and its application for identify issues related to the use of formal methods for interactive systems.

The framework has been applied to the tool suite called CIRCUS which embeds the HAMSTERS and Petshop tools.

We would like to discuss and possibly extend and refine this approach to understand better where usability problems arise while using formal methods for the design and analysis of interactive systems.

We hope also that participants will provide information about the notations and tools they are using to assess if the proposed framework is applicable more widely.

REFERENCES

1. Barboni E., Conversy S., Navarre D. & Palanque P. Model-Based Engineering of Widgets, User Applications and Servers Compliant with ARINC 661 Specification. 13th conf. on Design Specification and Verification of Interactive Systems (DSVIS 2006), LNCS Springer Verlag. p25-38
2. Barboni E., Ladry J.-F., Navarre D., Palanque P., and Winckler M.. 2010. Beyond modelling: an integrated environment supporting co-execution of tasks and systems models. In Proceedings of the 2nd ACM SIGCHI symposium on Engineering interactive computing systems (EICS '10). ACM, New York, NY, USA, 165-174.
3. Barboni, E., Bastide, R., Lacaze, X. Navarre, D., Palanque, P. Petri Net Centered versus User Centered Petri Nets Tools. AWPN 2003 - 10th Workshop on Algorithms and Tools for Petri Nets, Eichstätt, Germany, 26/09/2003-27/09/2003.
4. Bastide R., Navarre D., Palanque P., Schyn A. & Dragicevic P. A Model-Based Approach for Real-Time Embedded Multimodal Systems in Military Aircrafts. Sixth International Conference on Multimodal Interfaces (ICMI'04) October 14-15, 2004, USA, ACM Press
5. Beaudouin-Lafon, M. et al. CPN/Tools: A Post-WIMP Interface for Editing and Simulating Coloured Petri Nets. In Proc. of ICATPN'2001: 22nd International Conference on Application and Theory of Petri Nets (June 2001, Newcastle upon Tyne, England), Lecture Notes in Computer Science, Springer-Verlag, 2001, pp. 71-80.
6. Dix, A. Upside down ∀s and algorithms – computational formalisms and theory. J. Carroll (Ed.), HCI Models Theories and Frameworks: Toward a Multidisciplinary Science, Morgan Kaufmann, San Francisco (2003), pp. 381–429 (Chapter 14).
7. Dix, A. Formal methods for interactive systems. Academic Press, 1991.
8. Fuchs, N.E. Specifications are (preferably) executable. Journal on Software Engineering, vol. 7, issue 5, September 1992, pp. 323-334.
9. Gram, C., Cockton, G. Design principles for Interactive Software. London. 1996. Chapman & Hall.
10. Hamon A., Palanque P., Silva J-L., Deleris Y., and Barboni E. 2013. Formal description of multi-touch interactions. In Proceedings of the 5th ACM SIGCHI symposium on Engineering interactive computing systems (EICS '13). ACM, New York, NY, USA, 207-216.
11. Hayes, I., Jones, C.B. Specifications are not (necessarily) executable. Journal on Software Engineering, vol. 4, issue 6, November 1989, pp. 330-338.
12. International Standard Organisation, ISO 8807:1989, Information processing systems -- Open Systems Interconnection -- LOTOS -- A formal description technique based on the temporal ordering of observational behaviour, 1989.
13. Jensen, K., Kristensen, L. M., & Wells, L. (2007). Coloured Petri Nets and CPN Tools for modelling and validation of concurrent systems. Intern. Journ. on Software Tools for Technology Transfer, 9(3-4), 213-254
14. Kin K., Hartmann B., DeRose T., and Agrawala M.. 2012. Proton++: a customizable declarative multitouch framework. In Proc. of the 25th annual ACM Symp. on User Interface Software and Technology (UIST '12). ACM, 477-486
15. Martinie C., Palanque P., Barboni E., Winckler M., Ragosta M., Pasquini A., and Lanzi P.. 2011. Formal tasks and systems models as a tool for specifying and assessing automation designs. In Proceedings of the 1st International Conference on Application and Theory of Automation in Command and Control Systems (ATACCS '11). IRIT Press, France, 50-59.
16. Martinie C., Palanque P., Navarre D., and Barboni E.. 2012. A development process for usable large scale interactive critical systems: application to satellite ground segments. In Proceedings of the 4th international conference on Human-Centered Software Engineering (HCSE'12), Springer-Verlag, Berlin, Heidelberg, 72-93.
17. Martinie C., Palanque P., Winckler M.. Structuring and Composition Mechanisms to Address Scalability Issues in Task Models. IFIP TC13 Human Computer Interaction 2011 (INTERACT). p:134-152. Springer-Verlag.
18. 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. Comput.-Hum. Interact., 16(4), 18:1–18:56. 2009
19. Norman, D. «The Design of Everyday Things (Originally published: The psychology of everyday things).» The Design of Everyday Things (Originally published: The psychology of everyday things). New York: Basic Books, 1988.
20. Palanque P., Bernhaupt R., Navarre D., Ould M. & Winckler M. Supporting Usability Evaluation of Multimodal Man-Machine Interfaces for Space Ground Segment Applications Using Petri net Based Formal Specification. Ninth Int. Conference on Space Operations, Italy, June 18-22, 2006
21. Paternó, F., Santoro, C. Integrating model checking and HCI tools to help designers verify user interface properties. Palanque and Paternó (eds), DSV-IS 2000 Interactive Systems: Design, Specification and Verification. LNCS 1946, Springer 2001, pp. 135–150.
22. Roch, S. and P. H. Starke (1999, April). INA Integrated Net Analyser (V. 2.2). Humboldt-Universitat zu Berlin.
23. Silva, J-L., Fayollas, C., Hamon, A., Palanque, P., Martinie, C., Barboni, E. Analysis of WIMP and Post WIMP Interactive Systems based on Formal Specification. International Workshop on Formal Methods for Interactive Systems (FMIS 2013), London, 24/06/2013, Electronic Communications of the EASST.
24. Willans, J. S. and Harrison, M. D. 2001. Prototyping pre-implementation designs of virtual environment behavior. In Proc. of the 8th IFIP Int. Conf. on Engineering for Hum.-Comput. Interact., Lecture Notes In Comput. Sc., vol. 2254. Springer, 91–10.
25. Martinie C., Palanque P., Ragosta M., and Fahssi R.. 2013. Extending procedural task models by systematic explicit integration of objects, knowledge and information. In Proceedings of the 31st European Conference on Cognitive Ergonomics (ECCE '13). ACM, New York, NY, USA, Article 23, 10 pages.
26. International Standard Organization. (1996). DIS 9241-11: Ergonomic requirements for office work with visual display terminals (VDT) - Part 11 Guidance on Usability.