



HAL
open science

Task Models for Supporting Function Allocation between Operators and Autonomous Systems: Application to Collision Avoidance Operations for Spacecraft (2016)

Célia Martinie, Eric Barboni, David Navarre, Philippe Palanque, Racim Fahssi, Erwann Poupart, Eliane Cubero-Castan

► To cite this version:

Célia Martinie, Eric Barboni, David Navarre, Philippe Palanque, Racim Fahssi, et al.. Task Models for Supporting Function Allocation between Operators and Autonomous Systems: Application to Collision Avoidance Operations for Spacecraft (2016). 2016 Spring Symposium on Intelligent systems for supporting distributed human teamwork (AAAI-SS 2016), AAAI: Association for the Advancement of Artificial Intelligence, Mar 2016, Stanford, CA, United States. pp.85-94. hal-04109418

HAL Id: hal-04109418

<https://hal.science/hal-04109418v1>

Submitted on 30 May 2023

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 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's version published in:
<http://oatao.univ-toulouse.fr/24684>

Official URL

<https://www.aaai.org/ocs/index.php/SSS/SSS16>

To cite this version: Fayollas, Camille and Martinie De Almeida, Celia and Palanque, Philippe and Fahssi, Racim Mehdi *Task Models for Supporting Function Allocation between Operators and Autonomous Systems: Application to Collision Avoidance Operations for Spacecraft*. (2016) In: AAI 2016 Spring Symposium on Intelligent systems for supporting distributed human teamwork (AAAI-SS 2016), 21 March 2016 - 23 March 2016 (Stanford, CA, United States).

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

Task Models for Supporting Function Allocation between Operators and Autonomous Systems: Application to Collision Avoidance Operations for Spacecraft

Camille Fayollas, Célia Martinie, Philippe Palanque & Racim Fahssi

ICS-IRIT, University of Toulouse, 118, route de Narbonne, F-31062, Toulouse, France

{lastname}@irit.fr

Abstract

This position paper advocates that enhanced tasks description techniques can support the design of collaborative systems by supporting function allocation. This support can be useful for allocating functions between operators but also for migrating functions from operators to automation.

Designing systems in such a way that as much functions as possible are automated has been the driving direction of research and engineering in aviation, space and more generally in computer science for many years. In the 90's many studies (e.g. (Palmer 1995) related to the notion of mode confusion) have demonstrated that fully automated systems are out of the grasp of current technologies and that additionally migrating functions (Boy 1998) from the operator to the system might have disastrous impact on operations both in terms of safety and usability. In order to be able to design automation with a hedonic view of the involved factors (safety, usability, reliability, ...) a complete understanding of operator's tasks is required prior to considering migrating them to the system side or distributing them between a group of operators. This paper proposes a contribution for reasoning about multi-user (including automation) designs using a model-based approach exploiting refined collaborative task models. These models describe operations with enough details in order to reason about automation, function allocation and to rationalize the related designs. In this paper we present how such representations can support the assessment of alternative design options including automation. These contributions are exemplified in a multi-national context of Spacecraft Collision Avoidance Systems.

Introduction

The widespread of internet and the rise of social computing has demonstrated that dealing with single user applications is nowadays part of history. Designing interactive systems thus requires, most of the time, to address the needs of group of users involved in common tasks for which communication, cooperation and production activities are mediated by computers. Despite this undeniable situation, most of the research contributions in the area of interactive systems engineering still focus on single user applications. This is easily

understandable as multi-users application are far more difficult to build than single user ones. This difficulty comes from different sources:

- The difficulty to gather and understand the requirements as well as the need of groups of users;
- The difficulties to address the required communication infrastructures in order to allow both synchronous and asynchronous communication between collaborating users (as for instance argued in (Olson et al. 1995));
- The difficulty to define work allocation between operators and with automation in order to ensure adequate distribution of work load, authority and tasks.
- The difficulty to ensure usability of these applications that are used jointly by different users (with different characteristics and needs) and under different environmental conditions (time zones, seasons, light, sound, ...);
- The difficulty to ensure the reliability of these computing systems relying deeply on underlying communication mechanisms, networks, concurrent behaviours ...

This paper aims at proposing a model-based approach for the design of usable and reliable collaborative applications. To address the usability issue we propose a notation for describing collaborative task i.e. tasks having group of users trying to achieve common goals. This notation extends current models such as GTA (Van der Veer et al. 1996) or CTT (Mori et al. 2002). As for CTT, which the most mature notation in that domain, extensions refine further the task types (see section 3), adds explicit representation of information and knowledge required for performing the tasks and does not require the construction of an "artificial" task model describing the collaboration.

To address reliability, we propose the use of the ICO formalism and its related tool PetShop extended in order to edit and execute models of interactive distributed applications. This work takes advantage of previous work done with ICO notation to formally specify distributed applications over

Corba middleware (Bastide et al. 2000). Following the philosophy presented in (Barboni et al. 2010) we propose also a synergistic approach integrating models of operators' tasks (described using the extended HAMSTERS notation) with models of the interactive system (described using the ICO notation).

These various elements are successively presented in the paper. This presentation is followed by the description of the application of the approach on a real life case study from the space domain. This case study consisted in designing and modeling a collaborative collision avoidance management application for the CNES (French Space Government Agency) Orbit Computation Center. Current existing and in use applications are not supported by dedicated tools for collaboration. They are distributed over many time zones, involve multi-national teams and aim at forecasting and avoiding collisions between spacecraft and space debris.

Enhanced Collaborative Tasks Descriptions to Address Collaboration and Automation

The HAMSTERS notation and CASE tool has been introduced in 2010 in order to provide support for task-system integration at the tool level (Barboni et al. 2010). Since then, this tool and notation has been refined several times in order to provide support for:

- Automation design. The notation has been extended to

- Structuring a large number and complex set of tasks introducing the mechanism of subroutines (Martinie et al. 2011b).
- Precise description of knowledge, information and objects required and manipulated (Martinie et al. 2013) in order to accomplish tasks.

These elements are necessary to describe collaborative activities but they are not sufficient. Hereafter are the extensions we propose in order to deal with collaborative activities of multiple operators involved in a common goal.

Adding notation elements to describe collaborative activities

Collaborative work is performed by several persons, each one having a role in the achievement of common goals. The concept of role we are using is the same as the one used in (Mori et al 2002) and (Van der Veer et al. 1996). In the same way, we also integrate the concept of actor (Van der Veer et al. 1996) in the HAMSTERS notation and tool.

Collaborative work can be described at different abstraction levels: at the group level and at the individual level. A group task is a set of task that a group has to carry out in order to achieve a common goal (McGrath 1984), whereas a cooperative task is an individual task performed by a person in order to contribute to the achievement of the common goal

		Abstract	Input	Output	I/O	Processing	Group
Abstract			Not applicable	Not applicable	Not applicable	Not applicable	
User	Indiv.						
	Coop.						
Interactive	Indiv.					Not applicable	
	Coop.					Not applicable	
System							

Figure 1. Task types in HAMSTERS

help with the analysis of function allocation between human and system thanks to the refinement of cognitive tasks into analysis and decision subtypes of cognitive tasks according to the Parasuraman model of human information processing (Martinie et al. 2011a).

(Roschelle et al. 1995).

In order to be able to describe group tasks, we introduce several new task types illustrated in Figure 1 (in the last right column). These group tasks provide support for describing high level activities that a group of person have to accomplish:

- An abstract group task is a task that can be decomposed into user, system, interactive and collaborative tasks.
- A group (of users) task is task that can be decomposed in user and collaborative user tasks.
- An interactive group task can be decomposed in interactive and collaborative interactive tasks.
- A system group task can be decomposed in system tasks.

The refinement of group tasks into low-level activities needs fine-grain task types to describe individual and cooperative tasks that have to be performed in order to contribute to the group activities. As individual task types were already available within HAMSTERS, we then introduce cooperative tasks, illustrated in Figure 1. A cooperative task is a task related to a role and accomplished in correlation with another cooperative task that relates to a different role. A cooperative task may be of various types within the user and interactive main family types.

Cooperative tasks may be performed within various space-time constraints (local/distant, synchronous/asynchronous) (Ellis 1991). These constraints can be described with notation elements illustrated in Figure 2.





	Local	Distant
Synchronous	 Cooperative input task	 Cooperative input task
Asynchronous	 Cooperative input task	 Cooperative input task

Figure 2. Elements of notation related to space-time constraints

Cooperative task may be dedicated to one or more of the following type of collaborative activities: production, coordination, communication. It is then possible to associate one or more properties amongst this set. For example, Figure 3 a) shows that one task is dedicated to coordination whereas Figure 3 b) shows that the task is dedicated to both coordination and communication.

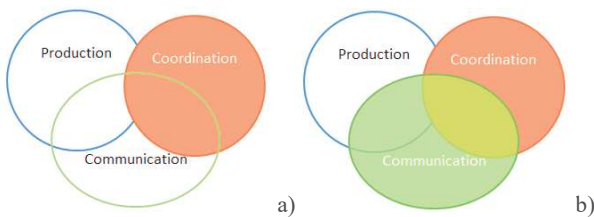


Figure 3. Example of cooperative task properties from a “functional clover” (Laurillau & Nigay 2002)

Automation Levels and Human Models for Automation

Parasuraman and Riley in (Parasuraman & Riley 1997) have expanded the *automation's* definition emphasizing human-

machine comparison and they describe automation as a device or system that accomplishes (partially or fully) a function that was previously, or conceivably could be, carried out (partially or fully) by a human operator. This implies that automation can vary across a continuum of levels, from the lowest level of fully manual performance to the highest level of full automation and the several levels between these two extremes are illustrated in the following table.

TABLE I. LEVELS OF AUTOMATION OF DECISION AND ACTION SELECTION (EXCERPT FROM (PARASURAMAN ET AL. 2000))

HIGH	10. The computer decides everything, acts autonomously, ignoring the human
	9, 8, 7. ... 3, 2
LOW	1. The computer offers no assistance: human must take all decisions and actions

Even though those levels can support the understanding of automation they cannot be used as a mean for assessing the automation of a system which has to be done at a much finer grain i.e. “function” by “function”. However, if a detailed description of the “functions” is provided they make it possible to support both the decision and the design process of migrating a function from the operator’s activity to the system or vice versa.

As stated in (Parasuraman et al. 2000), automated systems can operate at specific levels within this continuum and automation can be applied not only to the output functions but also to *input* functions. Figure 4 presents the four-stages model of human information processing as introduced in (Parasuraman et al. 2000).



Figure 4: Simple four-stages model of human information processing

The first stage refers to the acquisition and recording of multiple forms of information. The second one involves conscious perception, and manipulation of processed and retrieved information in working memory. The third stage is where decisions are accomplished by cognitive processes and the last one contains the implementation of a response or action consistent with decision made in the previous stage.



Figure 5: Four classes of system functions (that can be automated)

This model of human information processing has a similar counterpart in system’s functions as shown in Figure 5. Each of these functions can be automated to different degrees. For instance, the *sensory processing* stage (in Figure 4) could be migrated to the *information acquisition* stage (in Figure 5)

by developing hardware sensors. The second stage in the human model could be automated by developing inferential algorithms (as for instance in recommender systems). The third stage involves selection from several alternatives which can be easily implemented in algorithms. The final stage called *action implementation* refers to the execution of the choice. Automation of this stage may involve different levels of machine execution and could even replace physical effectors (e. g. hand or voice) of the operator (Parasuraman and Wickens 2008).

These aspects have been integrated in HAMSTERS as represented by dedicated tasks types as shown in Figure 2 (column Processing where decision, analysis ... are made explicit).

Example from a Large Case Study

HAMSTERS, as presented above, has been used to design and develop a prototype of groupware application belonging to the space ground segment category of applications. This study has been led in the context of a Research and Technology project funded by the French Space Government Agency (CNES). MARACCASS stands for Models and Architectures for the Resilience and Adaptability of Collaborative Collision Avoidance System for Spacecraft and aims at studying methods, techniques and tools to design and develop collaborative applications. This project is particularly targeting groupware for the management of collision avoidance between satellites and space objects. In this section, we present illustrative extracts from the case study which are relevant to highlight the key points of the contribution.

Management of collision risks between space objects and satellites

CNES and various other international agencies have to cope with the increasing number of space fragments, which are a threat to on-going satellite missions. Collision avoidance management is a collaborative, cross-team, and international activity. Amongst the national and international organizations, two main types of teams can be distinguished: the space observation teams and the satellite mission teams. The observation teams, thanks to various equipment's and tools are gathering information about space objects and their trajectories (past, present, future). The mission teams focus on one particular space object (usually a satellite) and are monitoring and controlling the space object they are in charge of and its operations. If the observation team detects a collision risk between a satellite and a space object, it contacts and alerts the mission team in charge of the satellite.

Roles and main goals to manage collision risks

In this case study, we take the example of the collaboration between the CNES team in charge of monitoring space objects (called the Orbit Computation Center or OCC) and the SMOS satellite mission team. In order to collaboratively

manage a collision risk, the teams are assisted with several non-integrated software tools: individual software tools to analyze probability of collision and traditional communication tools (email and telephone) to coordinate and communicate about the risk.

Preliminary work before high-fidelity prototyping phase

The first phase of the project has consisted in analyzing current activity with the production of corresponding task models. Then, we proposed several low-fidelity prototypes for a new groupware application to support collaborative activities of collision risk management. These low-fidelity prototypes take into consideration groupware principles (Ellis et al. 1991) but also contributions about design considerations for collaborative visual analytics (Heer & Agrawala 2007). We then produced task and system models from low-fidelity prototypes that had been validated with operational teams. In the next paragraphs we present extracts from models and from the high-fidelity groupware prototypes that highlight how the proposed framework has been applied to develop a high-fidelity prototype of the collaborative application for collision risk management. In these extracts, we will focus on the collaborative asynchronous activities related to posting annotations (OCC engineer role) and consulting these annotations (SMOS controller role) in the corresponding remote applications. Figure 8 and Figure 9 presents screenshots of the two remote applications dedicated to collaborative management of collision risks. Figure 8 presents the application dedicated to OCC engineers (with a larger set of functionalities such as deep probabilistic calculus and Conjunction Summary Messages creation and edition). In the presented screenshot, a popup window is opened in order to let the OCC engineer edit an annotation. Figure 9 presents the application dedicated to the mission controllers with a reduced set of functionalities. Its main purpose is to provide situation awareness about the collision risks related to the mission and communication and coordination support. In the presented screenshot, an annotation is displayed (pinned to the table) to the attention of the SMOS mission controller.

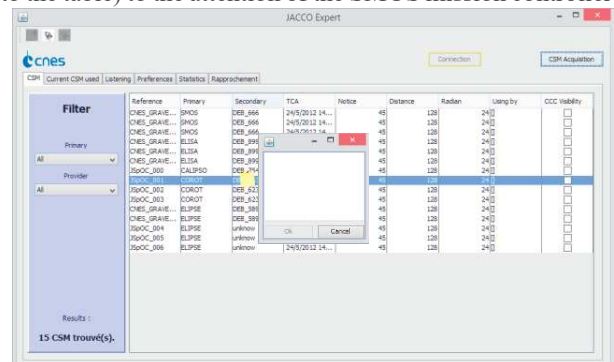


Figure 6. Screenshot of the Hi-Fi prototype for collision risks management dedicated to OCC engineers

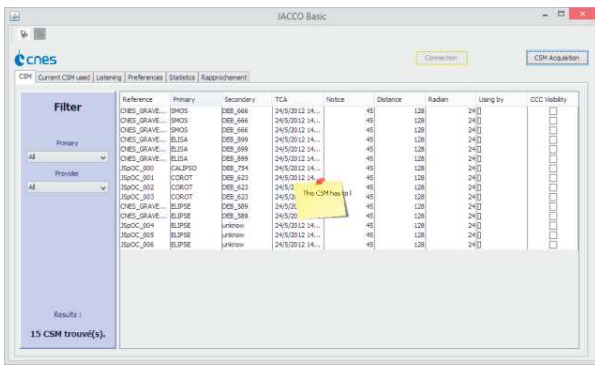


Figure 7. Screenshot of the Hi-Fi prototype for collision risks awareness dedicated to SMOS mission engineers

Task models

In this section, extracts of the task models illustrate the HAMSTERS extensions and especially how the new cooperative task types have been applied to support the development of the groupware Hi-Fi prototype.

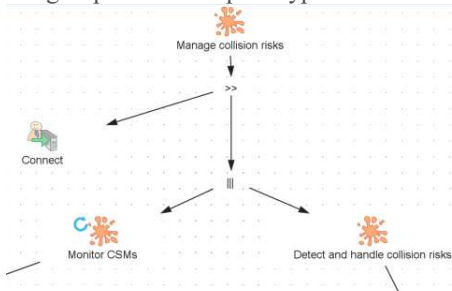


Figure 8. Extract of the highest level task model for the OCC activities (OCC engineer role)

Orbit Computation Center (OCC) engineers

Figure 8 presents an extract from the set of activities that have to be performed by the expert engineer on duty from the Orbit Computation Center to monitor and manage collision risks.

Figure 9 presents an extract from the set of activities performed once a collision risk has been detected for the SMOS satellite mission. In particular, it shows the sequence of activities led when the SMOS mission controller was not available for a live communication. The OCC engineer first creates an annotation (“Create annotation” input tasks), then positions the annotation (iterative task “Move annotation”) until the position is adequate (“Fix annotation position” input task). The OCC engineer then edits the annotation (input task “Edit annotation”), decides to send the annotation (cognitive decision task “Decide to send annotation”) and then send the annotation (cooperative asynchronous task “Send annotation”).

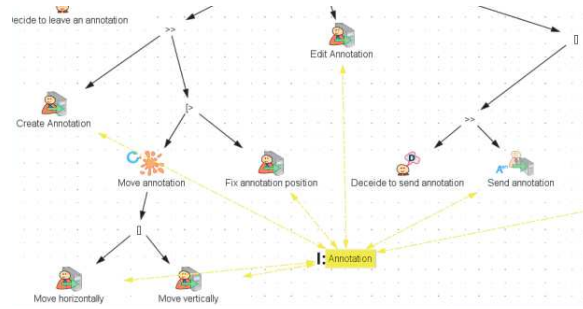


Figure 9. Extract of the task model “Handle collision risk between satellite and fragment” for the OCC engineer role

SMOS command and control room controllers and engineers

Figure 10 presents an extract from the set of activities that have to be performed by the SMOS controller when warned by the OCC engineer.

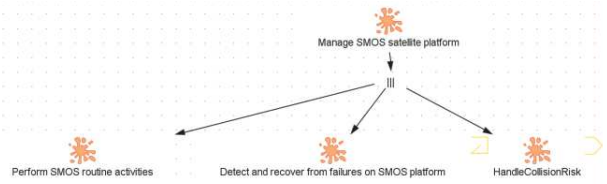


Figure 10. Highest level model for the SMOS mission activities

Figure 11 presents an extract from the set of activities performed once a collision risk has been detected. In particular, this set of activities is cooperative and bound to the above presented set of activities for the OCC engineer role. Once the OCC engineer has sent an annotation, it is displayed in the SMOS remote application (cooperative output task “Display new annotation”). When the SMOS mission controller will be available for consulting the application, s/he detects and acknowledges reception of the annotation (cooperative input asynchronous task “Acknowledge lecture of annotation”). S/he then analyzes the reported risk and may delete the annotation (cooperative input task “Delete annotation”).

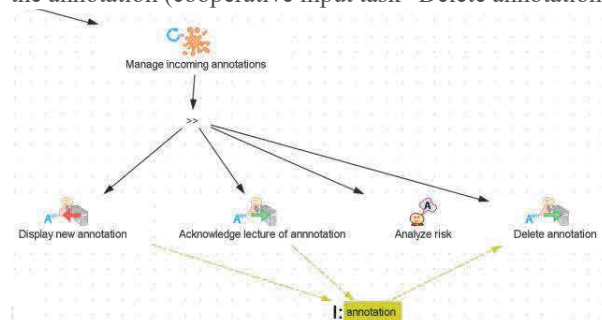


Figure 11. Extract of the task model “Handle collision risk” for the SMOS mission controller role

Discussion

This position paper has presented how extended tasks descriptions can support the description and the analysis of team work. That point has not been detailed due to space

constraints but the detailed descriptions make it possible to assess respective work load, needs for information sharing and work pressure (such as performance of tasks under temporal constraints).

The fact that information acquisition, analysis of such information and decision making are explicit, detailed task description allow identification of good candidate towards automation either being based on the task types (following MABA-MABA concepts in (Fitts 1951) and refined in (Carver & Turoff 2007).

Work that has been done on the satellite collision avoidance system could be used for discussion in the workshop. Some important aspects that are complex to handle such as multi-language, time difference, asynchronous/synchronous communications as well as the critical aspects.

References

- Boy, G. (1998). Cognitive Function Analysis for Human-Centered Automation of Safety-Critical Systems. Proc. of ACM CHI 1998: 265-272
- Palmer, E. (1995) Oops, it didn't arm - A Case Study of Two Automation Surprises. 8th International Symposium on Aviation Psychology, Ohio State University.
- Olson, J.S., Olson, G.M., & Meader, D.K. (1995) "What mix of video and audio is useful for remote real-time work?" Proceedings of CHI '95, ACM SIGCHI, 362-368.
- Van der Veer, G. C., Lenting, V. F., Bergevoet, B. A. (1996) GTA: Groupware Task Analysis - modeling complexity. Acta Psychologica, 91, 297-322.
- Mori, G., Paternò, F., Santoro, C. (2002) CTTE: support for developing and analyzing task models for interactive system design. IEEE Trans. Softw. Eng. 28, 8, 797-813
- Bastide, R., Palanque, P., Sy, O., Navarre, D. (2000) Formal specification of CORBA services: experience and lessons learned. In Proc. of ACM OOPSLA 2000, 105-117.
- Barboni E., Ladry J-F., Navarre D., Palanque P. and Winckler M. (2010). Beyond modeling: an integrated environment supporting co-execution of tasks and systems models. ACM EICS'10, 165-174
- Martinie C., Palanque P., Barboni E., Ragosta M. (2011a) Task-Model Based Assessment of Automation Levels: Application to Space Ground Segments. Proc. of the IEEE System Man and Cybernetic conference, Anchorage, 2011
- Martinie, C.; Palanque, P. A. and Winckler, M. (2011b): Structuring and Composition Mechanisms to Address Scalability Issues in Task Models. Proc. IFIP TC 13 conference INTERACT, Springer Verlag, p. 589-609.
- Martinie, C., Palanque, P., Ragosta, M and Fahssi, R. (2013) Extending Procedural Task Models by Explicit and Systematic Integration of Objects, Knowledge and Information, ECCE 2013, 23-33.
- McGrath J. E. (1984) Groups: Interaction and Performance. Prentice Hall, Inc., Englewood Cliffs, 1984.
- Roschelle, J., & Teasley, S. D. (1995). The construction of shared knowledge in collaborative problem solving. In C. E. O'Malley (Ed.), Computer-supported collaborative learning (pp. 69-197).
- Ellis C. A., Gibbs S. J., Rein G. (1991) Groupware: some issues and experiences, Comm. of the ACM, v.34 n.1, p.39-58, Jan. 1991.
- Yann Laurillau and Laurence Nigay. (2002). Clover architecture for groupware. ACM conference on Computer supported cooperative work (CSCW '02). ACM, 236-245
- Parasuraman R. & Riley V.A. (1997). Humans and automation: Use, misuse, disuse, abuse. Human Factors, vol. 39, pp. 230-253.
- Pasuraman, R.; Sheridan, T.B. & Wickens, C.D. (2000). A model for types and levels of human interaction with automation. Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Trans. on, vol.30, no.3, pp.286-297, May 2000.
- Parasuraman, R. & Wickens, C.D. (2008) Humans: still vital after all these years of automation. Human Factors, vol. 50, no. 3, June 2008, pp. 511-520.
- Heer; J., Agrawala, M. 2007. Design Considerations for Collaborative Visual Analytics. Proc. of IEEE Symp. on Visual Analytics Science and Technology (VAST '07). IEEE Computer Society, 171-178.
- Fitts PM (ed) (1951) Human engineering for an effective air navigation and traffic control system. National Research Council, Washington, DC
- Carver L., Turoff M. (2007) Human-computer interaction: the human and computer as a team in emergency management information systems. Commun. ACM 50(3): 33-38 (2007)
- Palanque P., Basnyat S. (2004) Task Patterns for Taking Into Account in an Efficient and Systematic Way Both Standard and Erroneous User Behaviours. In: Johnson C.W., Palanque P. (eds) Human Error, Safety and Systems Development. IFIP International Federation for Information Processing, vol 152. Springer, Boston, MA
- Marco A. Winckler, Philippe Palanque, and Carla M. D. S. Freitas. 2004. Tasks and scenario-based evaluation of information visualization techniques. In Proceedings of the 3rd annual conference on Task models and diagrams (TAMODIA '04). ACM, New York, NY, USA, 165-172.
- Célia Martinie, Philippe Palanque, David Navarre, Marco Winckler, and Erwann Poupart. 2011. Model-based training: an approach supporting operability of critical interactive systems. In Proceedings of the 3rd ACM SIGCHI symposium on Engineering interactive computing systems (EICS '11). ACM, New York, NY, USA, 53-62. DOI: <https://doi.org/10.1145/1996461.1996495>
- Philippe Palanque, Eric Barboni, Célia Martinie, David Navarre, and Marco Winckler. 2011. A model-based approach for supporting engineering usability evaluation of interaction techniques. In Proceedings of the 3rd ACM SIGCHI symposium on Engineering interactive computing systems (EICS '11). ACM, New York, NY, USA, 21-30. DOI: <https://doi.org/10.1145/1996461.1996490>
- Célia Martinie, Philippe Palanque, Martina Ragosta, and Racim Fahssi. 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. DOI=<http://dx.doi.org/10.1145/2501907.2501954>
- Célia Martinie, Eric Barboni, David Navarre, Philippe Palanque, Racim Fahssi, Erwann Poupart, and Eliane Cubero-Castan. 2014. Multi-models-based engineering of collaborative systems: application to collision avoidance operations for spacecraft. In Proceedings of the 2014 ACM SIGCHI symposium on Engineering interactive computing systems (EICS '14). ACM, New York, NY, USA, 85-94. DOI: <https://doi.org/10.1145/2607023.2607031>