What Can Be Learnt from Engineering Safety Critical Partly-Autonomous Systems when Engineering Recommender Systems

Camille Fayollas
Célia Martinie
Philippe Palanque
Eric Barboni
ICS-IRIT, University of Toulouse,
118, route de Narbonne, 31042
Toulouse, France
{lastname}@irit.fr

Yannick Deleris
AIRBUS Operations,
316 Route de Bayonne,3 1060
Toulouse, France
yannick.deleris@airbus.com

EICS'16, June 21-24, 2016, Bruxelles, Belgium.

Abstract
Human-Automation Design main target is to design systems in such a way that the couple system operator performs as efficiently as possible. Means for such designs include identifying functions (on the system side) and tasks (on the operator's side) and balancing the allocation of tasks and functions between operators and the systems being operated. Allocating functions to the most suitable actor has been the early driver of function allocation [18]. The philosophy of recommender systems is that the system will provide a set of options for the users to select from. Such behavior can be easily connected to previous work on levels of automation as defined by Sheridan [34] and lessons can be drawn from putting together these two views. When these automations (including the one of recommender systems) are not adequately designed (or correctly understood by the operator), they may result in so called automation surprises [25, 32] that degrade, instead of enhance, the overall performance of the operations. This position paper identifies issues related to bringing recommender systems in the domain of safety critical interactive systems. While their advantages are clearly pointed out by their advocates, limitations are usually hidden or overlooked. We present this argumentation in the case of the ECAM (Electronic Centralised Aircraft Monitor) of which some behavior could be considered as similar to the one of a recommender sys-

tem. We also highlight some engineering aspects of deploying recommender systems in the safety critical domain.

Author Keywords
Automation, recommender systems, transparent automation, operator tasks, performance

ACM Classification Keywords
D.2.2 [Design Tools and Techniques]: Computer-aided software engineering (CASE); H.5.3 [Group and Organization Interfaces]

Introduction
Human-Automation Design main target is to design systems in such a way that the couple system operator performs as efficiently as possible. Allocating functions to the most suitable actor has been the early driver of function allocation as advocated by Fitts [18]. Such work is known as MABA-MABA (Men Are Better At âĂŞ Machine Are Better At) where the underlying philosophy is that automate as many functions as possible was perceived as adequate (see for instance [11]). This technology-centered view has led to unsafe and unusable systems [32] and it became clear that the design of usable partly-autonomous systems is a difficult task.

Recommender systems belong to this trend of work on automation, even though that characteristic is not put forward or even ignored by their designers and promoters. When the word automation is connected to recommender systems it is usually for explaining that the recommender system is evolving autonomously as in “automated collaborative filtering” used for instance in GroupLens [21].

When automation is not adequately designed, or correctly perceived and then understood by the operator, they may result in so called automation surprises [33] that degrade, instead of enhance as expected, the overall performance of the operations and might lead to incidents or even accidents [25]. The issue of usability of recommenders systems has also been identified in the early days [35] even though only perceived as an interactive application being used by users and not an interaction taking place with a partly-autonomous system. Work in that domain focuses on usefulness of the recommender systems and on their usability or user experience [20].

This paper argues that having an automation-centered view on recommender systems helps to identify design issues related to their user interfaces and could inform design decisions and evaluation of these systems. Such a perspective could also help understanding issues that have to be addressed prior to the deployment of such systems in the context of safety critical interactive systems.

The next section proposes a short overview of the main concepts related to automations and focuses on the human aspects of automation. Then the paper positions recommender systems within that context and highlights similarities with other systems as well as their specificities. The paper then presents the case study of the ECAM (Electronic Centralised Aircraft Monitor) and how this system relates to recommender systems. Last section lists design and engineering issues related to the deployment of recommender systems in the area of safety critical interactive systems.

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.

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

Figure 1: Levels of automation of decision and action selection from [34] and [27]

As stated in [27], 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 2 presents the four-stage model of human information processing as introduced in [27].

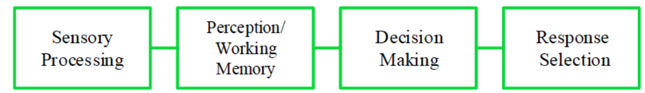

Figure 2: Simple four-stages model of human information processing.

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. The last one contains the implementation of a response or action consistent with decision made in the previous stage. The first three stages in that model represent how the operator processes the information that is rendered by the interactive system. The last stage identifies the response from the user that may correspond to providing input to the controlled system by means of the interactive system (flow of events from the user towards the controlled system.

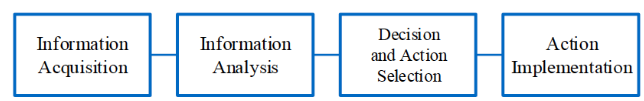

Figure 3: Four classes of system functions (that can be automated)

The model in Figure 2 (about human information processing) has a similar counterpart in system's functions as shown in Figure 3. Each of these functions can be automated to different degrees. For instance, the sensory processing stage (in Figure 2) could be migrated to the information acquisition stage (in Figure 3) 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 with 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 [28]. The stages of human information processing as well as their corresponding classes of system functions are used to analyze and design which tasks are performed by the human operator and which functions are performed by the system (also called function allocation as defined in [15]).

Based on this theoretical framework, several techniques and methods have been proposed to analyze, design and evaluate human automation interaction. Proud et al. proposed the LOA (Level Of Autonomy) Assessment Tool [30] (based on a LOA Assessment Scale) which produces analytical summaries of the appropriate LOA for particular functions and has been applied to an Autonomous Flight Management System. Cummings et al. [13] identified a refinement mechanism for the decision making step, to help in deciding which one of the human or of the system should perform a given decision task. More generally, techniques based on cognitive task analysis, such as the one proposed in [3], help in understanding precisely the different tasks that are actually performed by the human operator. Model based approaches take advantage of task analysis and propose to systematically ensure consistency and coherence between task models and system behavioral description [22]. Johansson et al. [19] developed a simulation tool to analyze the effect of the level of automation and emphasize the importance of a simulation framework to have a feedback on design choices before deploying the system. Finally, several techniques have been coined to provide support for formal verification of human automation interaction [9], which aim at providing tools for checking conformance between what the system has to perform, and what the user is responsible for. For each of these techniques and methods, human automation interaction is dealt with as a whole and thus focusing on goal-related tasks.

Recommender systems as partly-autonomous system

Recommender systems may be based on different approaches. They may implement content-based filtering, knowledge-based filtering, collaborative filtering or hybrid filtering [8]. We don’t describe here the various types of recommender systems in details and encourage the interested reader to see [8] for a very detailed and pedagogic survey. We here focus on content-based approaches for recommender systems as it is a relevant filtering approach for interactive cockpits. Figure 4 presents a typical architecture of a content-based recommender system. The system stores a set of items and each item is described according to a set of attributes. In such systems the user is described according to these attributes too, this description being named user profile. According to the user profile and the attributes of the set of items, the systems proposes to the user a set of recommendations.

Recommender Systems and Levels of Automation: According to the levels of automation presented in Figure 1 recommender systems typically fall within levels 2 to 4 depending on the number of alternatives presented to the user. Rules for designing autonomous systems would thus apply to recommender systems to avoid know issues

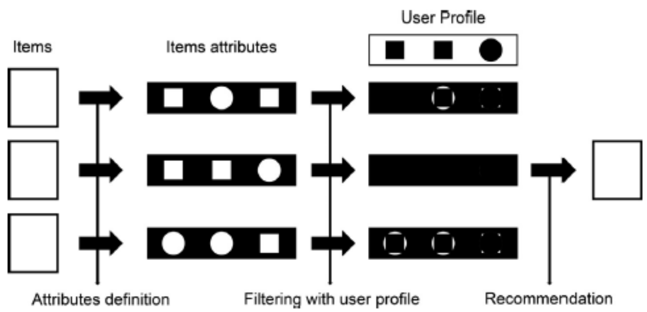

Figure 4: Content-based recommendation principle (from [8])

such as automation surprises [32] and [25]. Issue of transparency of automation has been also identified for recommender systems [14] while a process and a notation to systematically engineer transparency for partly-autonomous systems was proposed in [7].

Recommender Systems and systems functions: The recommender system behavior in Figure 4 and the classes of system functions in Figure 3 can be related as follows:

- **Information acquisition:** this stage corresponds to the process in the recommender systems browsing the internal source of items being candidates for recommendation. Information can be entirely stored at initialization or gathered during execution.
- **Information analysis**: this stage corresponds to the recommender system process of correlating user profile with items description.
- **Decision and action selection**: this stage corresponds to the filtering process of the recommender system selecting amongst the list of candidate items and ranking them before presentation to the operator.
- **Action implementation**: this stage corresponds to the presentation on the user interface of the selected items to the operator. That presentation of information can be sometimes enriched with argumentation about the rationale for selecting items [12]. It is also important to note that, following that presentation of information, user interaction is usually available allowing users to browse the list of recommended items, to access more information about them and to select the desired one. Such operator behavior is taken explicitly into account by the operator behavior model of Figure 2 and detailed below.

Recommender Systems and operators behavior: As far as user activity is concerned the operator behavior described in Figure 2 can be refined for describing interaction with a recommender system.

According to the four stages of human information processing proposed in Figure 2 it is easy to relate to the recommender system behavior:

- **Sensory processing:** while interacting with the recommender system this activity would consist in all operators' information sensing both from the recommender and from the system under use. Localization of the information from the recommender system might deeply impact that sensing.
- **Perception/working memory:** at this stage information from the recommender system will be integrated with the information presented by the system. It is important to note that human errors such as interference, overshooting a stop rule... [31] and thus should be avoided (and is not possible detected, recovered or mitigated).

- **Decision making:** this corresponds to the decision of the operator for selecting one of the recommendations presented by the recommender system (if more than one is offered).

- **Response selection:** this is the actual selection of one of the recommender system recommendation. As stated above, that stage might involve additional cycles within this 4 stages model (while operators interact with the recommender systems e.g. browsing the recommendation or accessing more information about a given recommendation).

The mapping between recommender system processes and Parasuraman's system functions as well as the mapping between activities done with recommender systems and Parasuraman's stages of human information processing, provides support to analyze the impact of the level of automation of the recommender system functions on the operators' task.

Illustrative example

To exemplify the concept presented above in the context of a safety critical system, we present a case study from the avionics domain: the ECAM (Electronic Centralized Aircraft Monitor) in the Airbus family. The ECAM is a system that monitors aircraft systems (e.g., the engines) and relays to the pilots data about their state (e.g., if their use is limited due to a failure) and the procedures that have to be achieved by the pilots to recover from the failure.

The ECAM system is composed of several systems. More particularly, the Flight Warning System (FWS) is in charge of the processing of data from the monitoring of the aircraft systems. This processing enables: i) the display of information about the status of the aircraft systems parameters (using the System Display (SD)); ii) the display of warnings about system failures and procedures that have to be completed by the pilot to process the detected warning (using the Warning Display (WD)) and iii) the production of aural and visual alerts (using several lights and loudspeakers in the cockpit).

The SD and WD are displayed, in the cockpit of the A380, on two separated Display Units (DU). These two DUs are highlighted in Figure 5 and are part of the eight of DUs composing the Control and Display System (CDS). The CDS is the interactive system in aircraft cockpits (flight decks) that offers various operational services of major importance for flight crew. It displays aircraft parameters, and enables the flying crew to graphically interact with these parameters using a keyboard and a mouse (KCCU for Keyboard and Cursor Control Unit) to control aircraft systems.

Figure 5: WD and SD in the cockpit of the A380

As presented in Figure 6, if the ECAM has created, simultaneously, several warning messages, it sorts them, in order to obtain a display order, according to three inhibition mechanisms:

- *Their priority:* a priority is associated to each warning message;
- *FWS internal behavior:* some warning messages may be inhibited in case of presence of others warning messages (for instance, the "APU fault" warning message is not displayed if the "APU fire" is already detected);
- *The current flight phase:* some warning messages are only displayed when the aircraft is in a given flight phase (for instance, flight management systems failures are not displayed after landing).

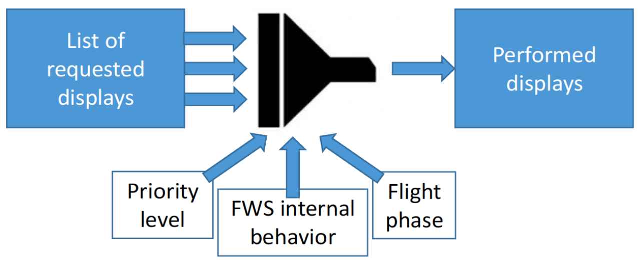

Figure 6: Principle for the display of the warnings

Therefore, the warning messages are displayed (in the processed display order) to the pilots, within the WD, with three different colors, representing their priority level:

- *Red warnings* that require immediate actions from the pilots (e.g., the loss of an engine);
- *Amber warnings* that require non-immediate actions from the pilots (e.g., a fault within the APU);
- *Green warnings* that only require monitoring from the pilots but do not present any danger.

Figure 7 presents an example of the display of warning messages (one red warning and three amber warnings) and their associated recovery procedures on the ECAM. In this example, the red warning (L1 in Figure 7) informs the pilot that the autopilot is not working anymore. Therefore, the first amber warning (L2 in Figure 7) informs the pilot that the auto-thrust is not working anymore. The corresponding recovery procedure (L3 in Figure 7) indicates to the pilot that s/he has to take responsibility for the thrust by moving the thrust levers. The second amber warning (L4 and L5 in Figure 7) informs the pilot that the flight control laws are not working anymore. The corresponding recovery procedures (L6 in Figure 7) indicates to the pilot that s/he has to take responsibility for the aircraft speed that must be under 0.82 MACH.

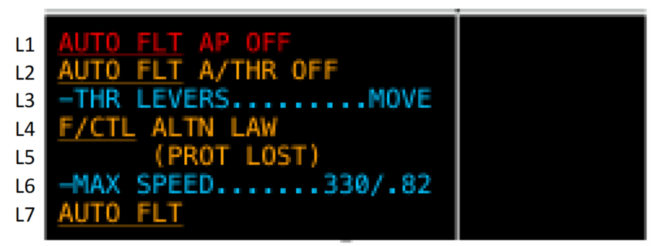

Figure 7: Example of the display of warning messages on the ECAM (from [10])

These warnings messages notification are similar to recommendations in recommender systems (see, for instance, the one presented in Figure 4) in the sense that the system sorts the warning messages and their associated recovery procedures and proposes, to the pilots, an order for their treatment. In this example, the system indicates to the pilot that the auto-thrust management function is off and indicates to the pilot that s/he shall manually move the thrust levers (lines 2 and 3 in Figure 4). Using Parasuraman's models, we can analyze that the system fall within level 4 of automation (Figure 1). In other cases, the system may propose a list of prioritized alarms and recovery procedures, which make the system fall within level 3 of automation (Figure 1). As inferred in the Parasuraman's level of automation, the more alternatives the system proposes, the less automated.

Design and engineering issues of recommender systems in safety critical domain

This section tries to identify the potential of recommender systems as well as the design issues related to their engineering.

The classification framework of recommender systems proposed in [29] identifies multiple application domains where recommender systems have been deployed (as an excerpt is presented in Figure 8).

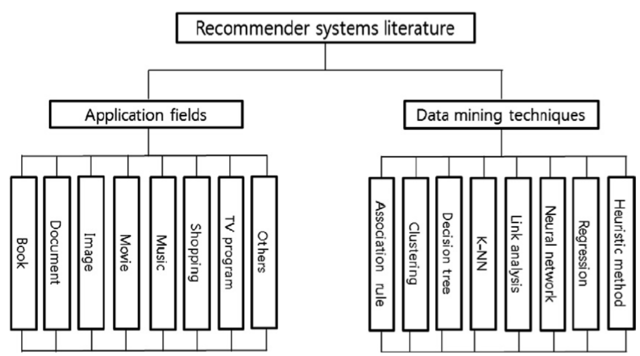

Figure 8: Classification framework of recommender systems (from [29])

It is interesting to note that none of them target at critical or safety critical domain. [14] presents the evaluation of a recommender system for single pilot operations but no information is given about the design and development of the underlying system and nor about its user interface and interaction techniques.

We have considered several engineering approaches to examine these issues. First, the ICO user interface design techniques enables to develop usable and reliable interaction techniques [24]. Complemented with task modelling (for describing operators goals and tasks to be performed to reach these goals), it can be used to analyze user's task w.r.t. system's behavior [22]. At last, task models can also be used to assess whether the user or the system should handle a particular task in a particular context [22]. All of these techniques aim at finding the optimal collaboration solution between the user and the system but were not applied with a recommender systems, even with the AMAN (Arrival Manager) advisory tool for air traffic control which could be also considered as a simple recommender system [23].

However these approaches do not deal with the possible dynamic change of behavior of the system, especially if it has machine learning capabilities (reinjecting operators' selections in the items information). Additionally, considering that the safety-critical user interfaces require additional design and development paths, we identified the following set of issues that must be considered if the system is (partly) autonomous:

- What is usability of a recommender system in a critical context and how to evaluate it (as operators follow extensive training and have deep knowledge of the behavior of the supervised systems),
- How to guarantee the safety and dependability of the

possible interactions when browsing recommended items,

- How to guarantee the safety and dependability of the underlying recommender system behavior,
- How to analyze and prevent operators' errors,
- How to assess and design responsibility, authority and liability between the recommender system and the operators (for instance in aircraft the entire authority belong to the captain and not to the first officer),
- How to design and specify interaction techniques where autonomous behavior from the system interfere with operator input (including the question on how to model that formally [7]),
- How to design interaction so that the operators can foresee the systems' future steps and states and the impact of selecting one recommendation instead of another one,
- How to design interactions when the automation (recommendation) can fail and how to notify the operators about degradation of the recommender system (for each of the stages in Figure 3),
- How to enhance and evaluate aspects of user experience, while fulfilling the constraints of a safety-critical system which has to be secure, safe, reliable and usable.

Summary and Conclusion

This position paper proposed to consider recommender systems as partly-autonomous systems. We have demonstrated that their behavior is similar to the ones of autonomous systems and that existing classifications in that domain are applicable to recommender systems.

We have shown on a simple example from the aviation domain that current systems exhibits some of the characteristics of recommender systems. We have also highlighted design and development issues that currently prevent recommender from being deployed in the context of safety critical systems but we have also highlighted some of the problems to be addressed.

Future work deals with the definition of engineering approaches for building reliable and fault-tolerant recommender systems following what has been done in the past for interactive cockpit applications as presented in [16] and [36]. It is important to note that trade-off between properties (such as usability and dependability as presented in [17]) will also be present in the case of recommender systems in safety critical applications.

References

[1] Johnny Accot, Stéphane Chatty, Sébastien Maury, and Philippe Palanque. 1997. Formal transducers: models of devices and building bricks for the design of highly interactive systems. In *Design, Specification and Verification of Interactive Systems' 97*. Springer, 143–159.

[2] Johnny Accot, Stéphane Chatty, and Philippe Palanque. 1996. A formal description of low level interaction and its application to multimodal interactive systems. In *Design, Specification and Verification of Interactive Systems' 96*. Springer, 92–104.

[3] Julie A Adams, Curtis M Humphrey, Michael A Goodrich, Joseph L Cooper, Bryan S Morse, Cameron Engh, and Nathan Rasmussen. 2009. Cognitive task analysis for developing unmanned aerial vehi-

cle wilderness search support. *Journal of cognitive engineering and decision making* 3, 1 (2009), 1–26.

[4] ARINC. 2002. ARINC 661 Cockpit Display System Interfaces to User Systems. ARINC Specification 661. (2002).

[5] Rémi Bastide, Philippe Palanque, Duc-Hoa Le, Jaime Muñoz, and others. 1998. Integrating rendering specifications into a formalism for the design of interactive systems. In *Design, Specification and Verification of Interactive Systems' 98*. Springer, 171–190.

[6] Olivier Bau and Wendy E Mackay. 2008. OctoPocus: a dynamic guide for learning gesture-based command sets. In *Proceedings of the 21st annual ACM symposium on User interface software and technology*. ACM, 37–46.

[7] R Bernhaupt, M Cronel, F Manciet, C Martinie, and P Palanque. 2015. Transparent Automation for Assessing and Designing better Interactions between Operators and Partly-Autonomous Interactive Systems. In *Proceedings of the 5th International Conference on Application and Theory of Automation in Command and Control Systems*. ACM, 129–139.

[8] Jesús Bobadilla, Fernando Ortega, Antonio Hernando, and Abraham Gutiérrez. 2013. Recommender systems survey. *Knowledge-Based Systems* 46 (2013), 109–132.

[9] Matthew L Bolton, Ellen J Bass, and Radu I Siminiceanu. 2013. Using formal verification to evaluate human-automation interaction: A review. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 43, 3 (2013), 488–503.

[10] Bureau d'Enquêtes et d'Analyse. 2012. *Rapport final-Accident survenu le 1er juin 2009 à l'Airbus A330-203 immatriculé F-GZCP exploité par Air France vol AF 447 Rio de Janeiro-Paris*. Technical Report. Tech. rep., République Française, Ministère de l'Ecologie, du Développement durable et de l'Énergie.

[11] Alphonse Chapanis. 1996. *Human factors in systems engineering*. John Wiley & Sons, Inc.

[12] Carlos Iván Chesnevar and Ana G Maguitman. 2004. Arguenet: An argument-based recommender system for solving web search queries. In *Intelligent Systems, 2004. Proceedings. 2004 2nd International IEEE Conference*, Vol. 1. IEEE, 282–287.

[13] Mary L Cummings and Sylvain Bruni. 2009. Collaborative Human–Automation Decision Making. In *Springer handbook of automation*. Springer, 437–447.

[14] Arik-Quang V Dao, Kolina Koltai, Samantha D Cals, Summer L Brandt, Joel Lachter, Michael Matessa, David E Smith, Vernol Battiste, and Walter W Johnson. 2015. Evaluation of a recommender system for single pilot operations. *Procedia Manufacturing* 3 (2015), 3070–3077.

[15] Andy Dearden, Michael Harrison, and Peter Wright. 2000. Allocation of function: scenarios, context and the economics of effort. *International Journal of Human-Computer Studies* 52, 2 (2000), 289–318.

[16] Camille Fayollas, Jean-Charles Fabre, Philippe Palanque, Eric Barboni, David Navarre, and Yannick Deleris. 2013. Interactive cockpits as critical applications: a model-based and a fault-tolerant approach. *International Journal of Critical Computer-Based Systems 17* 4, 3 (2013), 202–226.

[17] Camille Fayollas, Célia Martinie, P Palanque, Yannick Deleris, J-C Fabre, and David Navarre. 2014. An Approach for Assessing the Impact of Dependability on Usability: Application to Interactive Cockpits. In *Dependable Computing Conference (EDCC), 2014 Tenth European*. IEEE, 198–209.

[18] Paul M Fitts. 1951. Human engineering for an effective air-navigation and traffic-control system. (1951).

[19] Björn Johansson, Åsa Fasth, Johan Stahre, Juhani Heilala, Swee Leong, Y Tina Lee, and Frank Riddick. 2009. Enabling flexible manufacturing systems by using level of automation as design parameter. In *Winter Simulation Conference*. Winter Simulation Conference, 2176–2184.

[20] Bart P Knijnenburg, Martijn C Willemsen, Zeno Gantner, Hakan Soncu, and Chris Newell. 2012. Explaining the user experience of recommender systems. *User Modeling and User-Adapted Interaction* 22, 4-5 (2012), 441–504.

[21] Joseph A Konstan, Bradley N Miller, David Maltz, Jonathan L Herlocker, Lee R Gordon, and John Riedl. 1997. GroupLens: applying collaborative filtering to Usenet news. *Commun. ACM* 40, 3 (1997), 77–87.

[22] Célia Martinie, Philippe Palanque, Eric Barboni, Marco Winckler, Martina Ragosta, Alberto Pasquini, and Paola Lanzi. 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*. IRIT Press, 50–59.

[23] Celia Martinie, Philippe Palanque, Alberto Pasquini, Martina Ragosta, Eric Rigaud, and Sara Silvagni. 2012. Using complementary models-based approaches for representing and analysing ATM systems' variability. In *Proceedings of the 2nd International Conference on Application and Theory of Automation in Command and Control Systems*. IRIT Press, 146–157.

[24] David Navarre, Philippe Palanque, Jean-Francois Ladry, and Eric Barboni. 2009. ICOs: A model-based user interface description technique dedicated to interactive systems addressing usability, reliability and scalability. *ACM Transactions on Computer-Human Interaction (TOCHI)* 16, 4 (2009), 18.

[25] Everett Palmer. 1995. Oops, it didn't arm'- A case study of two automation surprises. In *International Symposium on Aviation Psychology, 8 th, Columbus, OH*. 227–232.

[26] Raja Parasuraman and Victor Riley. 1997. Humans and automation: Use, misuse, disuse, abuse. *Human Factors: The Journal of the Human Factors and Ergonomics Society* 39, 2 (1997), 230–253.

[27] Raja Parasuraman, Thomas B Sheridan, and Christopher D Wickens. 2000. A model for types and levels of human interaction with automation. *IEEE Transactions on systems, man, and cybernetics-Part A: Systems and Humans* 30, 3 (2000), 286–297.

[28] Raja Parasuraman and Christopher D Wickens. 2008. Humans: Still vital after all these years of automation. *Human Factors: The Journal of the Human Factors and Ergonomics Society* 50, 3 (2008), 511–520.

[29] Deuk Hee Park, Hyea Kyeong Kim, Il Young Choi, and Jae Kyeong Kim. 2012. A literature review and classification of recommender systems research. *Expert Systems with Applications* 39, 11 (2012), 10059–10072.

[30] Ryan W Proud, Jeremy J Hart, and Richard B Mrozinski. 2003. *Methods for determining the level of autonomy to design into a human spaceflight vehicle: a function specific approach*. Technical Report. DTIC Document.

[31] James Reason. 1990. *Human error*. Cambridge university press.

[32] Nadine B Sarter, David D Woods, and Charles E Billings. 1997. Automation surprises. *Handbook of human factors and ergonomics* 2 (1997), 1926–1943.

[33] Thomas B Sheridan and Raja Parasuraman. 2005. Human-automation interaction. *Reviews of human factors and ergonomics* 1, 1 (2005), 89–129.

[34] Thomas B Sheridan and William L Verplank. 1978. *Human and computer control of undersea teleoperators*. Technical Report. DTIC Document.

[35] Kirsten Swearingen and Rashmi Sinha. 2001. Beyond algorithms: An HCI perspective on recommender systems. In *ACM SIGIR 2001 Workshop on Recommender Systems*, Vol. 13. Citeseer, 1–11.

[36] A Tankeu-Choitat, David Navarre, P Palanque, Yannick Deleris, Jean-Charles Fabre, and Camille Fayollas. 2011. Self-checking components for dependable interactive cockpits using formal description techniques. In *Dependable Computing (PRDC), 2011 IEEE 17th Pacific Rim International Symposium on*. IEEE, 164–173.