Some Issues with Interaction Design and Implementation in the Context of Autonomous Interactive Critical Systems

Célia Martinie, Philippe Palanque & Martina Ragosta
IRIT-ICS, University of Toulouse 3, 118, Route de Narbonne
31062 Toulouse, France
{ martinie, palanque, ragosta}@irit.fr

ABSTRACT

Increasing analysis and decision capabilities in systems is a classical approach followed by designers and engineers to support operators in the command and control tasks of more and more complex systems. However, designing interfaces that will be used to operate intelligent systems and (partly) autonomous systems is a very complex activity altering in depth the development process of these systems. In the early days, the development of such systems was rather limited and thus easier to manage even though bad designs have been widely and frequently reported (e.g. [13, 15]). Nowadays, they can be integrated to several critical systems, such as Air Traffic Control management systems, as illustrated by the example presented in the case study. This position paper raises some issues related to the design and implementation of safety-critical user interfaces featuring intelligent and (partly) autonomous behaviors.

Keywords

Intelligent and autonomous systems, automation, user interaction, command and control.

INTRODUCTION

Currently, automation is one of the main means for handling increasing complexity in systems. Indeed, automation makes it possible for designers to transfer the burden of operators or users to a system. Parasuraman et al. have defined a classification [14] of the different levels of automation presented in Figure 1.

These levels of automation have been extensively used for assessing the automation level of command and control systems such as Air Traffic Management applications, aircraft cockpits or satellite ground segments (also called human in-the-loop systems). Thus automation is having a strong impact on workload, team size and human error. However, together with this high potential, automation raises complex issues related to design aspects of user interfaces and user interactions. Good designs are very difficult to perform and to assess especially when abstract properties such as predictability [2], transparency, honesty [3], usability or even user experience [4].

Command and control systems have to handle standard interactive objects as well as autonomous objects evolving in the same system but that are having potentially a complete independent behavior. In addition to that aspect, designers might want to empower the operators of such systems with (partly) automated tools to organize their work targeting at reducing users' workload and improving the overall performance.

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

Automation has already been studied in a number of disciplines and application fields: design, human factors, psychology, (software) engineering, aviation, health care, games [9] but the horizon for embedding them systematically into operational systems is not more than 10 years ahead. However, earlier adoptions of automation have not always been entirely successful as demonstrated by many studies in various application domains [13] or [15]. One distinguishing feature of the area of safety-critical systems is that system properties such as fault-tolerance [5], dependability [1] or usability all have to be treated on an equal basis.

This position paper focuses on **three design and engineering issues for the user interface of (partly) intelligent and autonomous systems**:

Non direct interaction: the engineering of the user interaction with autonomous systems (e.g. how to integrate autonomous behavior within the entire user interface), especially in cases were autonomous objects are represented on the user interface. For example, the representation (on an air Traffic Controller screen) of an unmanned aerial vehicle (UAV) with which no direct interaction is possible, together with aircrafts with which interaction is possible via an interactive screen (and that information is sent to the pilots).

User task migratability: design and engineering of user interfaces and interaction techniques for (partly) autonomous systems in charge of performing in an autonomous way tasks which were previously performed manually by operators. For instance, designing the user interface for an auto pilot in a plane or a cruise control must allow operators to set and/or control the auto pilot behavior.

Reconfigurable automation level at runtime: lies with the fact that automation cannot always be seen as static i.e. defined once for all for a given system but would preferably be dynamic i.e. evolving according to internal and external parameters such as operators' training, experience or workload, system state and for instance occurrence of failures or presence of alarms or environment information such as number of aircraft in the sector, types of aircrafts, presence of bad weather conditions, … It is when addressing such types of problem that the designers have to consider automation over the loop i.e. when automation takes place to adapt the system to better support the operators by means of personalization or by learning from users better strategies.

PROBLEM STATEMENT

One of the current challenges and goals for autonomous behavior are related to the analysis, planning, decision and action cycle as described in [14]. When interacting with a (partly) autonomous system the user interface and its underlying system should provide ways to support the operator in these activities. More precisely the operator should be supported:

- In analyzing the current context of the operations,
- In identifying one or several plans in order to carry on the current tasks or to handle unexpected events,
- In deciding amongst the various plans which one is the most appropriate,
- In inputting the plan into the system aspects such as reusing a previous entered plan or mending an existing one could significantly improve performance,
- In triggering the supervisory system to execute the plan which might include some degrees of autonomy (i.e. that the supervisory system has some delegated authority),
- In following the execution of the plan allowing the operator to know what has already been performed, what is the currently executed and what will be executed in the future steps.

Of course, the operator in charge of and responsible for the operations should always have the possibility to interrupt the current execution of the plan and possibly to resume it later on.

One solution to that problem could be to reduce the operator role to the one of automation overseer and thus only acting at a high (and abstract) strategic level. Such solution would make it very difficult (and nearly impossible) for the operator to come back to a more low (and concrete) tactical level especially in case of automation degradation or system failure. Thus, other solutions have to be indentified and designed requiring scientific means to assess:

- How the operator will be able to identify (from the currently available information about the autonomous system) new plans or modification to current potential plans (or potential configurations)?
- How the operator will be able to build new plans or configurations?
- How the operator will be able to assess beforehand the impact of a potential new plan or configuration?
- How the operator will be able to interact (both monitor and possibly interrupt) with the current configuration under "execution"? This interaction aspect can be particularly complex if, in a proactive system, the configurations are executed in an autonomous way by the supervision system. Additionally such interaction should be conformant to the ones used for normal interactions.

A SAFETY-CRITICAL CASE STUDY

This section refines the concepts presented above in the context of an Air Traffic Control (ATC) case study.

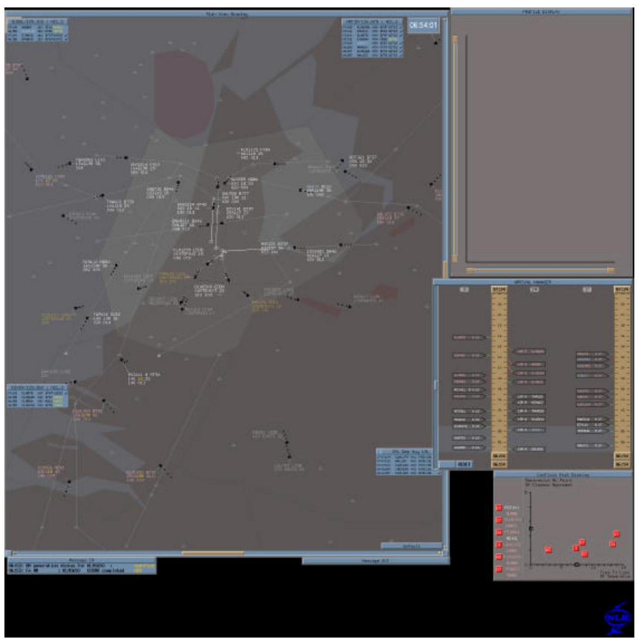

Figure 2. A screenshot of the user interface of ATC [6].

Presentation of the Air Traffic Control AMAN software

The case study is based on a command and control system for Air Traffic Control called AMAN (Arrival MANager) as proposed in [6]. Figure 2 is screenshot of such an interface which is composed of two parts, the left-hand side displaying a standard graphical interface for air traffic control (each icon representing the position of each aircraft in the sector and its flight information) and the right-hand side presenting the AMAN interface consisting of a list of advices proposed by the autonomous systems to the Air Traffic Controller. A detailed view of this AMAN interface is represented in Figure 3.

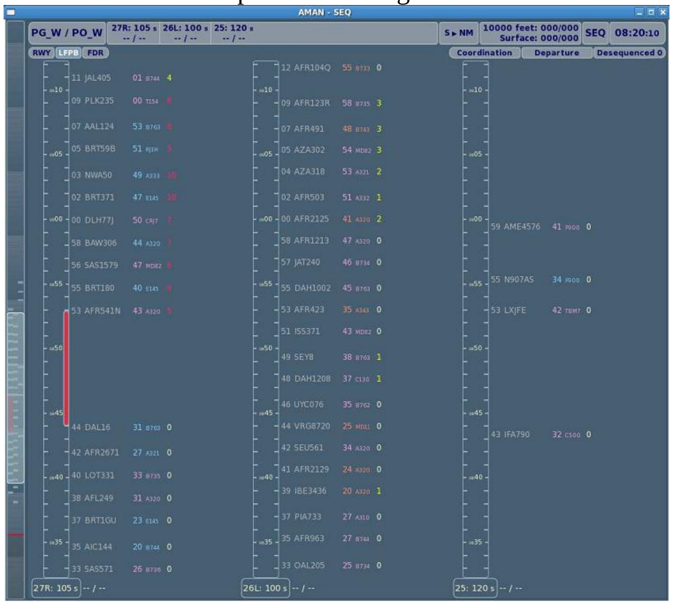

Figure 3. Detailed user interface of the AMAN system.

The AMAN system provides support to the controllers in establishing the optimal approach routes for partly autonomous systems (aircrafts). AMAN is a decision support tool for the person in charge of sequencing the aircrafts arrivals, which is called a sequence manager. According to [14] classification presented in Figure 1, the AMAN systems corresponds to levels 2-3 where the system computes alternatives and proposes them to the operator.

The sequence manager is permanently checking the list provided by AMAN but also analyzing if the provided list is accurate w.r.t. information provided by other persons and entities (executive controller, aircraft pilot, aircraft down going measures, planner…). At any time, if the sequence manager analyses that the displayed sequence is not relevant w.r.t. the current context, she/he can decide not to follow AMAN's advice.

Design opportunities for a learning version of the AMAN tool

As promoted by [16], we consider increasing the capabilities (analysis and interaction) of the AMAN tool so that it provides more accuracy and trust to the sequence manager.

Learning capability: as a first step, when the sequence manager does not follow AMAN's advice, the AMAN tool should be capable of learning why its advice has not been followed.

Explanations: AMAN tool should provide more information on the elements that were computed to determine the current sequence of aircrafts arrival. This feature would increase the transparency of the tool.

Disengagement capability: AMAN tool should provide a way for being disengaged by the user when the contextual information becomes wider and more complex than AMAN is capable to handle. In this kind of context, advices that have a high probability of being wrong or at least irrelevant should not be displayed (reconfiguration of the automation level).

As the AMAN tool is safety-critical and all of these opportunities have to be deeply analyzed w.r.t. dependability constraints as detailed in next section.

LESSONS LEARNED AND MAIN ISSUES

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 [12]. Complemented with task modelling, it can be used to analyze user's task w.r.t. system's behavior [11]. 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 [10]. All of these techniques aim at finding the optimal collaboration solution between the user and the system.

However these approaches don't deal with the possible dynamic change of behaviour of the system, especially if it has machine learning capabilities. 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 in a critical context and how to evaluate it,
- How to guarantee the safety and dependability of the possible interactions,
- How to analyze and prevent operators' errors,
- 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,
- How to design interactions when the automation can fail and how to notify the operators,
- 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.

CONCLUSION AND PERSPECTIVES

The research work presented here has been trying to bring together multiple perspectives for the design and evaluation of intelligent and (partly) autonomous interactive systems.

The contributions presented here have been focusing on the current practice in the safety-critical application domains and particularly in the Air Traffic Control management domain. We argue that automation issues in the safety-critical application domain overlap interaction issues for intelligent and autonomous systems. As of today, proposals have been made to assess and design analysis and decision levels for safety-critical intelligent systems (autonomous and non-autonomous), but engineering means of designing safety-critical interactive machine learning systems are rare, although required for certification and validation purpose.

AUTHORS' BACKGROUND AND INTERESTS

We have a long term experience in the area of interactive command and control systems including design and implementation issues of (partly-) autonomous systems. We have been working closely with main actors in the area of safety critical interactive systems including air-traffic management, aircraft cockpits and satellite ground segments. We have little expertise in the area of intelligent adaptive systems as the difficulties in terms of validation and certification prevent them for being used in safety critical contexts. However, in the domain of ATM for instance, traffic increase within the next few years requires looking at radically new solutions including higher automation levels and adaptive systems. We organized a SIG last year at CHI to try to foster interests of CHI community on automation issues [8]. We could share that perspective and the conclusions of the SIG during the workshop. Participation to the workshop would also be beneficial to our work by exchanging with knowledgeable researchers in the area of intelligent systems.

ACKNOWLEDGMENTS

This work is partly funded by CNES (National Space Studies Center) and Eurocontrol/SESAR research network HALA! (Higher Automation Levels in Aviation).

References

1. Avizienis A., Laprie J-C., Randell B., Landwehr C: Basic Concepts and Taxonomy of Dependable and Secure Computing. IEEE Trans. Dependable Sec. Comput. 1(1): 11-33 (2004).

2. Dix A. J. Formal methods for interactive systems. Academic Press 1991: 1-369.

3. Graham, C., & Cockton, G. (1996). Design principles for Interactive Software. London: Chapman & Hall

4. Law E L-C, Roto V., Hassenzahl M., Arnold P.O.S. Vermeeren, and Kort J. 2009. Understanding, scoping and defining user experience: a survey approach. 27th int. conf. on Human factors in computing systems (CHI '09). ACM, New York, NY, USA, 719-728.

5. Randell B., "System structure for software fault tolerance," IEEE Trans. Soft. Eng., vol. SE-1, no. 2, pp.220-232, 1975.

6. NLR PHARE Demonstration 3, Final Report, PHARE/NLR/PD3-4.4.2/FR;1.0, February 2000.

7. 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) ACM, p243 – 250

8. Bernhaupt R., Boy G., Feary M., and Palanque P. 2011. Engineering automation in interactive critical systems. In Proceedings of the 2011 extended abstracts on Human factors in computing systems (CHI EA '11). ACM, New York, NY, USA, 69-72.

9. Billings, C. E. Aviation Automation: the Search for a Human Centered Approach, Lawrence Erlbaum Associates, Inc., Mahwah, NJ, 1997.

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

11. Martinie, C., Palanque, P., Barboni, E., Winckler, M., Ragosta, M., Pasquini, A., Lanzi, P. Formal Tasks and Systems Models as a Tool for Specifying and Assessing Automation Designs. Int. Conf. on Application and Theory of Automation in Command and Control Systems, Barcelona, Spain, 2011.

12. Navarre, D., Palanque, P., Ladry, J., and 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 (Nov. 2009), pp. 1-56.

13. Palmer, E. "Oops, it didn't arm." - A Case Study of Two Automation Surprises. In Proc of 8th International Symposium on Aviation Psychology, Ohio State University, Columbus, Ohio, 1995.

14. Parasuraman, R.; Sheridan, T.B.; Wickens, C.D. "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.

15. Sarter, N. D., Woods D. How in the World Did I Ever Get Into That Mode? Mode Error and Awareness in Supervisory Control, Human Factors, 37, p., 5-19, 1995.

16. Stumpf, S., Sullivan, E., Fitzhenry, E., Oberst, I., Wong, W.-K. and Burnett, M. Integrating rich user feedback into intelligent user interfaces. Proc. IUI, ACM (2008), 50-59.