Adaptive Human-Automation Cooperation: A General Architecture for the Cockpit and its Application in the A-PiMod Project

Denis Javaux*, Florian Fortmann†, Christoph Möhlenbrink‡
*Symbio Concepts & Products, Bassenge, Belgium, Email: denis.javaux@symbio.pro
†OFFIS, Oldenburg, Germany, Email: florian.fortmann@offis.de
‡DLR, Braunschweig, Germany, Email: christoph.moehlenbrink@dlr.de

Abstract—**The design of future aircraft cockpits will be based on a cooperative team perspective of the human crew and the automation. The team perspective requires to rethink the interaction between the human crew and the automation. It further requires to develop a human-machine system architecture that supports this perspective. This paper describes a general architecture for adaptive human-automation cooperation in the cockpit. It relies on an analysis of the nature of the flight for better integration of the crew and cockpit automation in the joint, cooperative and adaptive completion of the flight. The architecture has been instantiated in the European project A-PiMod, which aims at developing adaptive multi-modal cockpits to support the interaction between the human crew and the automation.**

Keywords–Human-machine cooperation; adaptive systems; automation design.

I. INTRODUCTION

Improving aircraft safety is one of the main challenges to cope with the expected increase of future air traffic. Modern aircraft are highly complex socio-technical systems, comprised of a human crew and the automation. The trend towards more automation has changed the task of the human crew from manual to supervisory control [1], which has led to novel error sources. Studies have shown that 60-80% of aviation accidents are caused by human errors [2] [3], such as automation surprises [4], opacity [5], erroneous mental models [6], degraded situation awareness [7], and out of the loop problems [8]. These problems have significantly contributed to major incidents and accidents, despite the high level of training of the human crew. This hints at a disconnection between the human crew and the automation, inherent to the way automation is currently designed. To address this design problem, several authors [9] [10] argue that automation should be seen as a team player. Both, the human crew and the automation should constitute a deeply integrated team that achieves the mission safely and adaptively in all circumstances.

In an unpublished work, the first author has been investigating for years possible new architectures for human-automation cooperation in the cockpit. Results showed that cooperative human-automation systems are implicit in many operational settings, including the cockpit, and that many Human Factors issues plaguing them were likely the results of the non-acknowledgment of that implicit nature. These systems should be understood, modeled and explicitly, purposefully and rationally designed as explicit human-automation cooperative systems.

In this paper, we propose a general architecture for adaptive human-automation cooperation that precisely shows how to build such a human-automation team, based on an in-depth analysis of (1) the nature of the flight and its execution, and (2) adaptive, cooperative human-machine systems. The architecture is currently instantiated under the umbrella of the European Research & Development project A-PiMod [11], which consists of a consortium of partners from the research and the industry.

This paper is structured as follows. In Section II, we elaborate on the nature of the flight and its execution. In Section III, we describe the general architecture. In Section IV, we describe the application of the architecture within the A-PiMod project. In Section V, we finally provide a conclusion on our work.

II. THE NATURE OF THE FLIGHT AN ITS EXECUTION

In order to analyze, model and design an airliner cockpit as an explicit human-automation cooperative system it is necessary to understand the very nature of the flight, in abstract and purely functional terms. What is a flight and what does it entail from the point of view of a controller in charge of executing that flight.

A. Mission Level Tasks: What the Flight is About

Performing a flight is about going from origin to destination, safely and efficiently, while adapting to current contingencies (weather changes, system failures, crew incapacity, etc.). Indeed, the flight plan (F-PLN) cannot always be flown as intended and has to be altered, radically modified, or even aborted. Therefore, the F-PLN is a dynamic structure that is permanently adapted. It is akin to the most general flight task, that has to be performed by the aircraft (A/C). We call this task the Mission Level (ML) task. The ML task can be decomposed into several ML subtasks, that correspond to the different flight phases. Figure 1 shows the general structure of the ML task, represented as a graph.

Performing the ML task consists of progressing over that graph and taking appropriate alternative branches if necessary (e.g., to return to departing airport). The graph (the mission) is executed by some form of control logic. This control logic may be on the ground (e.g., if the A/C is a remotely controlled unmanned aerial vehicle), on-board (e.g., if the aircraft is completely autonomous, manually controlled by one to many human pilots, under shared control of a team of some form of automation and a human crew), or distributed between ground and on-board entities (e.g., if the A/C is a remotely controlled unmanned aerial vehicle with some form of automation on-board). All these control logic designs are functionally equivalent and perform the same functions. They differ solely in

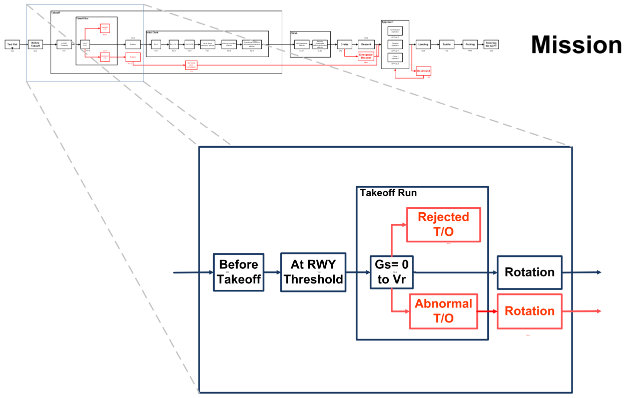

Figure 1. Representation of the ML task as a graph. The graph shows the structure of the flight, including its phases, and adaptation to contingencies (highlighted in red).

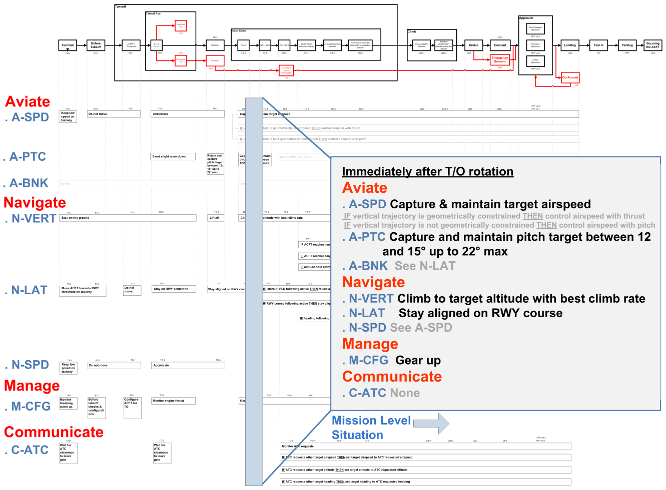

Figure 2. Translation of current ML subtask into mission execution tasks.

terms of their implementation (human or automation) and their physical distribution (ground or on-board).

The general architecture proposed in this paper relies on the deep recognition of that functional equivalence. It prompts at understanding contemporary cockpits as dynamic functional systems that implement the control logic needed to execute the mission, including its monitoring and adaptation whenever necessary.

B. Cockpit Level Tasks: What the Cockpit has to do

In modern commercial aviation, the execution, monitoring and adaptation of the ML task is achieved by the human crew and automation, seen as a set of cooperative human and machine agents in the cockpit. To achieve the ML task, the cockpit has to perform the following three types of Cockpit Level (CL) tasks:

(1) Mission (F-PLN) Monitoring and Adaptation Tasks: The F-PLN is not a static structure. During a given flight, it is frequently adapted, to cope with Air Traffic Control (ATC) requests, or because of contingencies that induce its modification (e.g., bad weather at destination airport, and weather avoidance during the flight). The cockpit therefore also has to permanently monitor the mission, assess the current circumstances, and decide if changes need to be made to the F-PLN. These are mission monitoring and adaptation tasks.

(2) Mission (F-PLN) Execution Tasks: The cockpit has to take as input the current ML subtask and perform corresponding lower level mission execution tasks (flight control, A/C configuration, interaction with ATC, etc.). The mission execution tasks can be organized along the familiar categories ”Aviate”, ”Navigate”, ”Communicate” and ”Manage”. In the schema of Figure 2, the ”Aviate” category contains, e.g., three flight control tasks: airspeed control (A-SPD), pitch control (A-PTC), and bank control (A-BNK). Similarly, the ”Navigate” category incorporates the flight navigation tasks vertical navigation (N-VER), lateral navigation (N-LAT), and ground speed control (N-SPD). The graph exactly shows which subtasks have to be executed to complete the mission. For example, at rotation during takeoff, A-SPD requests an accelerating airspeed, A-PTC a rotation to reach a target pitch between 15° and 22°, and A-BNK a bank angle of about 0° (wings level). Thus, taking the mission (F-PLN) as input, specific mission execution tasks are derived. These tasks specify what the A/C has to do to execute the current mission. If ever the mission needs to be modified, the mission execution tasks are dynamically updated to reflect the new mission.

(3) Task Distribution Tasks: The CL tasks have to be performed by the given control logic, in this case the aircraft cockpit as a whole, including the human crew and the automation. Some processing is needed to decide who will perform them. In today’s cockpits for example, mission (F-PLN) execution is mostly (99% of the time) performed by the Auto Flight System (AFS). Mission monitoring and adaptation is mostly achieved by the human crew, with some assistance from the Flight Management System (FMS). This allocation, however, is very static and this is detrimental to the cockpit adaptability and resilience. One of the objectives of the general architecture proposed in this paper is to permanently suggest a suitable distribution of CL tasks, based on the state, capabilities and workload of the agents, especially the human crew. This distribution is achieved by the task distribution tasks.

C. Agent Level Tasks: Executing Cockpit Level Tasks

CL tasks are executed by the agents in the cockpit after being distributed to them. We call the concrete distribution of these tasks the Agent Level (AL) tasks. As mentioned above, F-PLN execution in contemporary cockpits is mostly achieved by the automation, F-PLN monitoring and adaptation by the human crew with some assistance by the automation, and task distribution is achieved by the human crew and by mode control logics inherent to the automation. The AFS, including the autopilot (AP), Auto-Throttle (ATHR) and FMS indeed have specific internal modes. Each mode achieves one or more CL tasks (e.g., the speed mode on Airbus aircraft regulates airspeed). These systems have some mode transition logics that define how they switch between modes, and therefore between tasks. They implement a form of implicit and automatic task (re-) distribution in the cockpit.

One of the objectives of this paper is to suggest that there is a disconnection between these implicit task distributions by automation and the ones to which the human crew contribute,

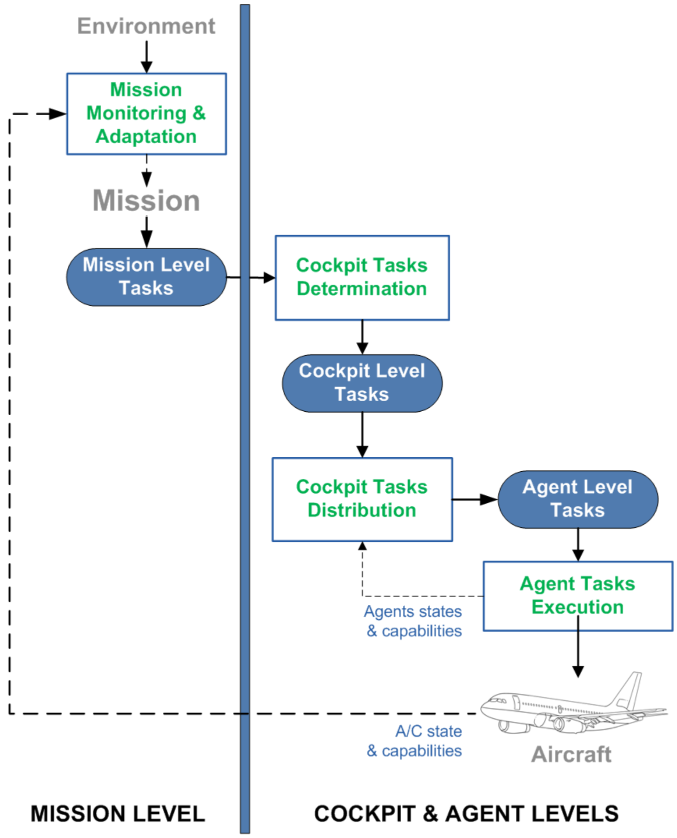

Figure 3. General Architecture for Adaptive Human-Automation Cooperation.

something typically manifested by the now infamous automation surprises. Automation surprises occur, e.g., because the human crew and the automation do not cooperatively define how tasks should be distributed, something the proposed architecture aims to improve upon.

III. AN ARCHITECTURE FOR MISSION, COCKPIT AND AGENT LEVEL TASKS

From this deeper understanding of the F-PLN and its execution, a general architecture for the distributed, cooperative and adaptive execution of ML, CL, and AL tasks in modern aircraft cockpits can be derived. It allows the execution by a cooperative human-automation system. The architecture goes beyond the scope of cockpits and is usable in most frameworks where a mission has to be achieved by a collective of human and machine agents. The architecture derives very naturally from the distinction above between ML, CL, and AL tasks (see Figure 3). It relies on four components that process them specifically:

(1) Mission Monitoring and Adaptation: The progress over the Mission (F-PLN) is assessed and as well as the context in which the Mission is executed. Context includes, e.g., weather near the A/C, along the foreseen F-PLN and at destination, the state of installations at destination (e.g., landing assistance systems such as the ILS), and the availability of the intended runway. It also includes the state of the A/C itself. Any abnormal contingency in the context will prompt for an adaptation of the Mission (F-PLN) (e.g., to bypass bad weather) or more drastic changes such as diversion to another airport. The output of this component is the current Mission task, a F-PLN that is always safe and flyable, based on the initial FPLN and permanent adaptation to external contingencies (and ATC requests).

(2) Cockpit Tasks Determination: The tasks the cockpit seen as a whole has to perform at all times are produced. There are two types of cockpit tasks: 1) mission independent tasks: these tasks have to be performed permanently by the cockpit: Mission (F-PLN) monitoring and adaptation tasks, Task distribution tasks and Agent level tasks: Executing cockpit level tasks; 2) mission dependent tasks: they are directly derived from the current Mission Task (a safe and flyable F-PLN) and implement the flight plan in terms of concrete actions. A possible strategy for producing them is to rely on the dimensions suggested in the Figure 2 (Aviate, Navigate, Communicate, Manage), which allow easily translating the F-PLN into lower level tasks such as having specific airspeed, ground speed, thrust, heading, pitch, bank, altitude, descent angles, communication occurrences, and actions on A/C systems such as the gear, flaps, and slats.

(3) Cockpit Tasks Distribution: Once cockpit tasks have been determined, they have to be distributed to the different agents available in the cockpit, that is the crew (PF and PNF) and various automated systems (e.g., AFS, automated fuel monitoring, etc.). This therefore takes as input the cockpit tasks to distribute and the state and capabilities of the agents in the cockpit (e.g., vigilance state of the crew, state of fatigue, situation awareness, workload, state of automation, etc.). In a normal flight today, most of the time, the Mission (F-PLN) execution tasks are assigned to the AFS, while most mission independent tasks are assigned to the crew with some assistance of cockpit automation systems (envelope protections, collision avoidance systems, etc.). This is particularly true of the Mission (F-PLN) monitoring and adaptation tasks.

(4) Agent Tasks Execution: The tasks assigned to the agents are then executed by them. Thus Mission (F-PLN) execution is today mostly achieved by the AFS and the other mission independent tasks by the crew with assistance from some automated systems.

Each of the four components in the architecture should be seen as a cooperative human-machine system in its own respect: each of them is made of human and machine agents cooperating to perform the work of the component. This is one of the key ideas of the architecture. Another important idea is that the agents in question are functional agents. A single physical agent can embody several functional agents (e.g., participate to several functional agents, in more than one, and possibly all of the four components). In the extreme case of fully manual flight, there are only four functional agents, one for each component, and they are all achieved by a single physical agent: the pilot. The pilot superposes the four functions. As will be seen later, in modern cockpits, the crew also superposes participation to all four components, into two single individuals: the pilot flying (PF) and pilot non-flying (PNF). This superposition idea is the key to cooperative system design.

As seen in Figure 3, the whole architecture is mostly divided into two sections: the Mission Level is about everything occurring ”outside the cockpit” and is about determining the Mission the logic in control of its execution has to achieve (cf. the notion of functional equivalence above. Other architectures, e.g., an autonomous drone, would be absolutely identical here). The Cockpit and Agent Levels are about the peculiar proposed implementation and how it achieves the mission (in term of the functional approach, all implementations would differ here).

Two main control loops exists within the architecture (beside the control loop closed by the agents on the A/C): a loop to monitor and adapt the mission and a loop to monitor and adapt the task distribution within the cockpit. They provide the adaptive capabilities expected from the architecture: to external circumstances (e.g., change of F-PLN in case of weather change) and to internal circumstances (e.g., change of task distribution because of high workload for one of the human crew). The whole architecture hints are the inherently mission-oriented character of future cockpit architectures. Future cockpits should be about the safe and adaptive completion of the mission, by an adaptive and cooperative system of human and machine agents (on-board, and/or on the ground).

IV. APPLICATION OF THE ARCHITECTURE WITHIN THE A-PIMOD PROJECT

A-PiMod (Applying Pilot Models for Safer Aircraft) is a European project that aims at developing adaptive automation for a multi-modal cockpit. Indeed, todays automation is indifferent to the emotional and cognitive state of the crew. Automation only supports the crew based on explicit and static task assignments, with no adaptive capabilities, even though it is capable of higher or lower levels of support if needed or when the capabilities of the crew are challenged. A novel approach to adaptive automation is needed and it must be applicable for real-time operations. Automation should be seen as a partner in the global endeavor of flying the aircraft with the human crew; they should adapt to each other and to the context, aiming at maintaining safety at all times as a team. The objective of A-PiMod is therefore to provide adaptive task distributions between the crew and automation, based the crew’s state (workload, situation awareness, or fatigue, etc.) and behavior (e.g., a critical task that is not performed by the crew will be taken over by automation). In order to provide means of improvement to the safety of flight, especially in times of continuously increasing performance levels, automation and information provision to the flight deck, A-PiMod works on adaptive automation and an adaptive multi-modal cockpit.

The A-PiMod consortium comprises eight European partners from 6 countries. A-PiMod will last from September 2013 to August 2016. A-PiMod is followed yearly by an advisory group of highly qualified pilots, Human Factors researchers and industry experts. During the project, the partners have defined a framework for adaptive automation based on the general architecture for adaptive human-automation cooperation shown in Figure 3. The A-PiMod project includes partners with competences for the design of a multi-modal cockpit, for building crew state inference models, with expertise in real-time risk assessment and training. The A-PiMod project is continuously going along with validation activities and addressing safety as an operational concept.

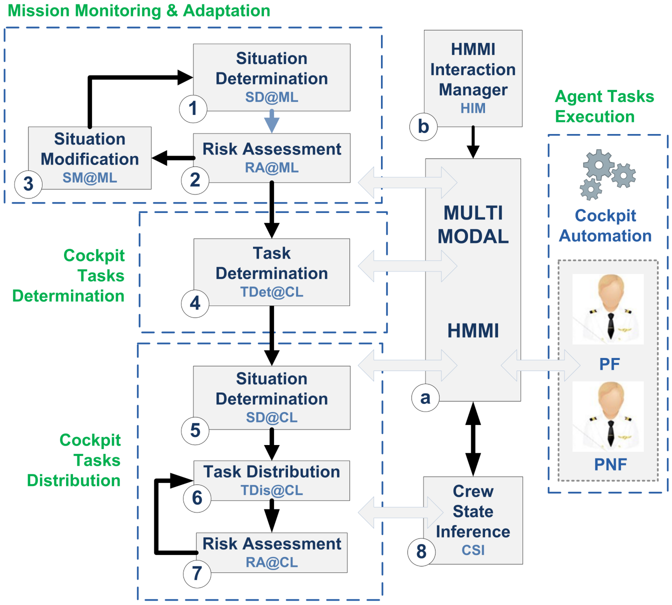

Figure 4. A-PiMod architecture.

The A-PiMod architecture is based on 8 components and 2 separate software modules, arranged within the three macro-components (Mission Monitoring and Adaptation, Cockpit Tasks Determination, Cockpit Tasks Distribution, and Agent Tasks Execution) of the general architecture previously shown in Figure 4. A peculiarity of the A-PiMod architecture, inherited for the general architecture, is the inherently cooperative nature of the components: the components are not only software. In the A-PiMod architecture, a component is made of a software module and of the human crew (PF and PNF). Thus, each component is a small cooperative system in itself. These components are accompanied by two exclusive ”software only” modules, which realize the interaction between the human crew and the automation. In the following, the components (1-8) and the exclusive software modules (a and b) of the instantiated A-PiMod architecture are described.

(1) Situation Determination at Mission Level: The situation determination at Mission Level component (SD@ML) is in charge of determining the current state of the mission and the context in which it is executed. This includes the progress on the F-PLN (mission phase/sub-phase), the state of the A/C and its systems and the environment in which the A/C operates (e.g., weather, runway availability at destination airport, and ATC).

(2) Risk Assessment at Mission Level: The risk assessment at Mission Level component (RA@ML) is in charge of determining the risk of not being able to achieve the mission as intended (e.g., can the current F-PLN be flown safely to the destination?). It uses as input the output of the situation determination at Mission Level component (SD@ML).

(3) Situation Modification at Mission Level: If the risk level is deemed unacceptable, control is passed to the situation modification at Mission Level component (SM@ML). This component is in charge of reducing the risk associated with the current situation to an acceptable level. For example, this will entail solving any threatening issue with the A/C systems (e.g., engine fire) or modifying the F-PLN, e.g., to avoid bad weather, or chose an alternate destination.

(4) Task Determination at Cockpit Level: When the risk level is back to acceptable levels, the cockpit level tasks for the current situation are determined by the task determination at Cockpit Level component (TDet@CL). This includes all F-PLN execution tasks, all F-PLN monitoring and adaptation tasks and all task distribution tasks (e.g., choosing which cockpit agent has to do what).

(5) Situation Determination at the Cockpit Level: The situation determination at the Cockpit Level component (SD@CL) then assesses the state of the cockpit, in particular the state of the agents, human and machine (e.g., crew and automation), in terms of availability and current capabilities (e.g., crew is fatigued).

(6) Task Distribution at Cockpit Level: The cockpit level tasks and the cockpit state are then processed by the task distribution at cockpit level component (TDis@CL) to produces one or more tentative distribution of the tasks to the cockpit agents.

(7) Risk Assessment at the Cockpit Level: The risk assessment at the Cockpit Level component (RA@CL) then assesses the risk(s) associated with these distribution(s). This risk evaluation is based on the state, workload and capabilities of the agents (e.g., the crew and automation, such as the AFS). A task distribution will only be selected if the associated risk is deemed acceptable, and if several distributions exists, the one with the lowest risk will be selected. If no acceptable distribution can be found, this information is passed back to the task determination at Cockpit Level component (TDet@CL) to state that the requested set of cockpit level tasks cannot be achieved safely by the cockpit. This information is then transferred to the situation modification at Mission Level component (SM@ML), which will typically produces an alternate F-PLN (because the previous one could not be flown safely by the cockpit). For example, this would happen if one of the crew was incapacitated and the cockpit workload was likely to be so high (e.g., due to bad weather at destination airport) that a diversion to another airport would be safer (and flyable by the diminished cockpit).

(8) Crew State Inference: Adaptivity of the task distribution within the cockpit heavily depends on the capability of the task distribution and risk assessment components at the Cockpit Level (TDis@CL and RA@CL) to know the current and future state of the crew. This information is provided by an eighth component, the Crew State Inference component (CSI). The CSI permanently monitors the crew and infers their current state, such as vigilance, workload and situation awareness. As all components in the architecture, the component is made of the crew and of a sophisticated software module. The software module produces its own inferences but the crew can alter them if needed, or even indicate they are fatigued or incapacitated before the module detects it. The CSI module infers intentions, situation awareness, and workload, using a combination of well-studied technologies, notably Bayesian and cognitive models.

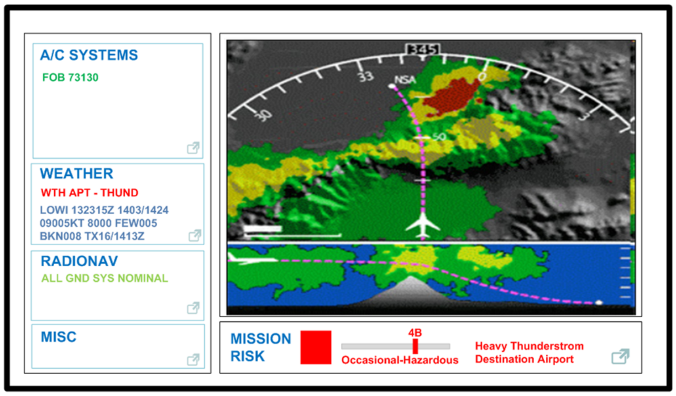

Figure 5. Paper prototype of the Mission Level display. It shows a map with the current flight plan and weather information, and associated mission risks.

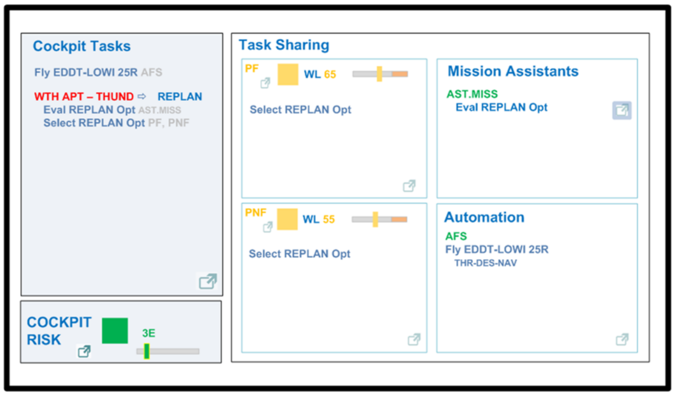

Figure 6. Paper prototype of the Cockpit Level display. It shows the current task distribution between the human crew and the automation, the workload for PF and PM and the associated cockpit risks.

(a) Human-Machine Multi-Modal Interface: The Human-Machine Multi-Modal Interface (HMMI) plays a role in communicating information to the crew and for supporting the interaction and cooperation between the crew and the software modules within the components (e.g., supporting the joint modification of the F-PLN by the SM@ML module and the crew). Besides the traditional input modalities, the HMMI relies on speech, gesture, and touch input. Further eye movements are recorded as a measure of attention. In A-PiMod, the HMMI for presenting information and supporting human-automation cooperation is currently under development. The current tentative solutions shown in Figure 5 and Figure 6 rely on two dedicated displays: the ML display and the CL display. The two displays support the interaction and cooperation between the human crew and all software modules in the architecture. They support the whole collaborative execution, monitoring and adaptation chain of the mission by the human crew and the modules, providing adaptive task distribution at all stages, based on the state, workload and capabilities of the cockpit agents. This provides adaptive capabilities to the HMMI, where the goal is to ensure the information displayed is indeed processed as intended.

(b) HMMI Interaction Manager: To drive the HMMI finally, the A-PiMod architecture sports the HMMI Interaction Manager. The HMMI Interaction Manager is a pure software module (e.g., not a component in which the crew intervene). Its task is to handle the interaction between the modules and the human crew, in particular by considering the current state and actions of the crew (provided by the CSI). The HMMI Interaction Manager gets requests from the modules to display information (on the ML and CL displays). It then displays the

corresponding information, but does not stop there. It starts monitoring if the information is perceived by the human crew (or if a task to be triggered by that information is executed). If not, it will enter an escalation strategy, e.g., by enhancing the salience of the information, and possibly resorting to alarms if necessary.

To implement the A-PiMod architecture, the project partners have specified and developed a series of software modules that act as team players with the crew to perform and manage the flight. These modules are integrated into several demonstrators that are used to conduct extensive validation sessions at DLR premises, of the underlying adaptive automation concepts and of the demonstrators themselves. Many of the modules in the A-PiMod architecture are driven by rule production systems. This allows for real-time behavior, deterministic execution and their certifiability.

V. CONCLUSIONS

We believe the general cockpit architecture above and its instantiation in the framework of A-PiMod has the potential to improve the safety of future aircraft. It provides a complete adaptive execution and adaptation chain for the mission, based on 4 main core components (instantiated into 8+2 components and software modules in A-PiMod). The architecture smoothly adapts to changes at the Mission level (e.g., bad weather) and at the Cockpit and Agent levels (e.g., incapacitation). In A-PiMod the later is achieved through a multi-modal HMMI and a CSI module. Each component is a fully cooperative system made of the crew and dedicated software agents (modules in A-PiMod). This allows task sharing within the components that range from full manual execution to full automatic execution. The software agents (modules) and the crew basically contribute the same tasks. The software agents can be seen - and should be designed - as cognitive agents [12]. This makes human interaction and cooperation with them far easier and robust. The architecture integrates all state transitions for automation systems in the cockpit into a single task distribution component, itself dealing with the allocation of tasks to the crew and automation. This makes the global behavior of automation far more integrated, easier to develop and debug, and synchronized with the crew during operators (reduction of automation surprises). The architecture provides a framework for aircraft that can be flown in full (or assisted) manual or full automatic control, with many intermediary configuration between which it is easy and safe to transition in flight. The architecture is also ideal for progressively, non disruptively and safely developing aircraft that implement more automation. Given the cooperative nature of the components in the two architectures (general and A-PiMod) it is possible at any time to revert in-flight to mixed or full manual modes.

ACKNOWLEDGMENT

The A-PiMod project is funded by the European Commission Seventh Framework Programme (FP7/2007-2013) under contract number: 605141 Project A-PiMod.

REFERENCES

[1] T. Sheridan, Telerobotics, Automation, and Human Supervisory Control. MIT press, 1992.

[2] C. Billings, Flight Deck Automation: Promises and Realities. NASA Ames Research Center, 1989, ch. Toward Human Centered Automation, pp. 167–190.

[3] R. Amalberti, “Automation in aviation: A human factors perspective,” Handbook of aviation human factors, 1999, pp. 173–192.

[4] N. B. Sarter, D. D. Woods, and C. E. Billings, “Automation surprises,” Handbook of human factors and ergonomics, vol. 2, 1997, pp. 1926–1943.

[5] C. Billings, Aviation automation: the search for a human-centered approach, ser. Human factors in transportation. Lawrence Erlbaum Associates Publishers, 1996.

[6] N. B. Sarter and D. D. Woods, “How in the world did we ever get into that mode? mode error and awareness in supervisory control,” Human Factors: The Journal of the Human Factors and Ergonomics Society, vol. 37, no. 1, 1995, pp. 5–19.

[7] M. Endsley, “Automation and situation awareness. automation and human performance: Theory and applications,” Parasuraman, R., Mouloua, M.(eds.), 1996, pp. 163–181.

[8] M. R. Endsley and E. O. Kiris, “The out-of-the-loop performance problem and level of control in automation,” Human Factors: The Journal of the Human Factors and Ergonomics Society, vol. 37, no. 2, 1995, pp. 381–394.

[9] F. Flemisch, S. Meier, J. Neuhöfer, M. Baltzer, E. Altendorf, and E. Özyurt, “Kognitive und kooperative systeme in der fahrzeugführung: Selektiver rückblick über die letzten dekaden und spekulation über die zukunft (cognitive and cooperative systems in vehicle control: Review of past and speculation of the future),” Kognitive Systeme, 2013.

[10] R. Parasuraman, M. Barnes, K. Cosenzo, and S. Mulgund, “Adaptive automation for human-robot teaming in future command and control systems,” DTIC Document, Tech. Rep., 2007.

[11] A-PiMod, “Applying pilot models for safer aircraft,” February 2015. [Online]. Available: http://www.apimod.eu

[12] R. Onken, “Cognitive cooperation for the sake of the human-machine team effectiveness,” DTIC Document, Tech. Rep., 2003.