Application of Human-Autonomy Teaming (HAT) Patterns to Reduced Crew Operations (RCO)

R. Jay Shively[1],* Summer L. Brandt[2], Joel Lachter[2], Mike Matessa[3], Garrett Sadler[4], and Henri Battiste[4]

[1]NASA Ames Research Center, Moffett Field, USA
robert.j.shively@nasa.gov
[2]San Jose State University, Moffett Field, USA
{summer.l.brandt, joel.lachter}@nasa.gov
[3]Rockwell Collins, Moffett Field, USA
mike.matessa@rockwellcollins.com
[4]NVH Human Systems Integration, Moffett Field, USA
garrett.g.sadler@nasa.gov, henri.battiste@gmail.com

Abstract. Unmanned aerial systems, advanced cockpits, and air traffic management are all seeing dramatic increases in automation. However, while automation may take on some tasks previously performed by humans, humans will still be required to remain in the system for the foreseeable future. The collaboration between humans and these increasingly autonomous systems will begin to resemble cooperation between teammates, rather than simple task allocation. It is critical to understand this human-autonomy teaming (HAT) to optimize these systems in the future. One methodology to understand HAT is by identifying recurring patterns of HAT that have similar characteristics and solutions. This paper applies a methodology for identifying HAT patterns to an advanced cockpit project.

Keywords: Design Patterns · Human-Autonomy Teaming (HAT) · Reduced Crew Operations (RCO)

1 Introduction

In this paper we propose the use of design patterns to aid in characterizing commonly occurring human-autonomy teaming (HAT) situations. The concept of design patterns, originally introduced by Christopher Alexander [1] in the context of architectural design, and extended to the domain of software engineering by Beck and Cunningham [2], provides abstractions that capture a general repeatable solution to commonly occurring problems. That is, they provide descriptions or templates for how to solve a problem that can be used in many different situations. While not finished designs that can be transformed directly into code, these design patterns have served as aids to more efficient development.

More specifically, design patterns can speed up the development process by providing tested, proven development paradigms. Effective software design requires considering issues that may not become visible until later in the implementation. Reusing design patterns helps to prevent subtle issues that can cause major problems and improves code readability for coders and architects familiar with the patterns. In addition, patterns allow developers to communicate using well-known, well-understood names for software interactions.

We believe that the design of HAT solutions might also benefit from the use of patterns. In this paper, we will discuss the methodology for developing such patterns and examine example applications to a project investigating reduced crew size on commercial airlines. Because this project involves replacing one member of a tightly coupled team with automation, it provides a very ripe environment in which to apply this methodology.

2 Design Patterns for Reduced Crew Operations

NASA is currently investigating the feasibility of reduced crew operations (RCO) for transport category aircraft. RCO envisions having one pilot on board domestic flights, and two pilots on board long-haul operations, where one of the two pilots is often off of the flight deck resting in the bunk. An important element of NASA's RCO research seeks to develop a concept of operation (ConOp) that covers the roles and responsibilities of the principal human operators, the automation tools used by them, and the operating procedures for human-human and human-automation interaction. The human-automation function allocation, in particular, is an ongoing NASA focus, drawing upon insights gathered from subject matter experts in industry, academia and government during technical interchange meetings and from empirical human-in-the-loop research [3,4,5].

The proposed NASA RCO ConOp [6] includes three basic human roles: the pilot on board (POB), the dispatcher, and when necessary, a ground pilot. The POB (unless incapacitated) would serve as the captain and pilot-in-command. As such, s/he would determine when to call on automation and ground support. The POB's main tasks would be to manage risk and resources (both human and automation).

Onboard automation would assist the POB with many tasks currently performed by the pilot monitoring, such as flight management system (FMS) input, assisting with checklists and validating inputs.

Ground-based automation would assist the dispatcher in a variety of tasks. Dispatch tasks would be similar to current operations (e.g., preflight planning, monitoring aircraft positions, and enroute reroutes), but for those tasks currently performed jointly with the pilot, the dispatchers (aided by automation) would absorb some of the POB's workload (e.g., creating new flight plans, vetting them with air traffic control (ATC) and uplinking them to the aircraft). In this ConOp, automation would assist the dispatcher with creation of pre-flight briefings, flight path monitoring, selection of divert airports, and optimizing reroutes. Automation would also be responsible for monitoring many flights and alerting the dispatcher to aircraft needing assistance. In addition, the POB could call the dispatcher for consultation on important decisions where s/he might previously have consulted the first officer (e.g., diagnosing an aircraft system caution light or determining the fuel consequences of a holding instruction).

Under high-workload or challenging off-nominal operating conditions, such as an engine fire or cabin depressurization, where the flight's needs exceed the capacity of a dispatcher responsible for many other aircraft, a ground pilot would be assigned to the flight for dedicated piloting support. The ground pilot would have remote access to fly the aircraft as needed. Similarly, if the POB was found to be incapacitated (by automation or the dispatcher), a ground pilot would be assigned to that aircraft and assume the role of pilot-in-command.

The remainder of this paper will discuss the methodology for developing design patterns for RCO.

3 Steps to Build a Pattern

Schulte [7, 8], in conjunction with Neerincx and Lange [9], proposed using a set of primitives to build HAT patterns. They proposed three types of agents: 1) human operators, 2) intelligent/cognitive agents, and 3) automated tools. The agents, which can either be co-located or distributed, can be connected by cooperative, supervisory, or communications links.

As described by Schulte [7, 8], a critically important preliminary task in the construction of HAT patterns is the identification of the *Work Objective.* The *Work Objective* identifies the aspects that initiate and characterize the mission or purpose of the work. The *Work Objective* provides a black box description of the *Work Process*, which includes informational inputs (e.g., ATC clearance), environmental inputs (e.g., airspace) and supply inputs (e.g., fuel). The *Work Process*, utilizing all of these inputs and the *Work Objective*, produces a *Work Process Output* (e.g., reducing target speed) on the *Work Object* (e.g., speed of aircraft) that distributes meaningful physical and conceptual actions to human-automation team members within the overall *Work Environment.*

Using these primitives, we have been working to build patterns that describe HAT in the RCO context. An initial use case was developed, and by walking through each step, the agents and links required to depict one such pattern were identified.

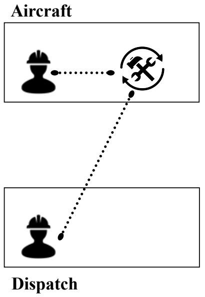
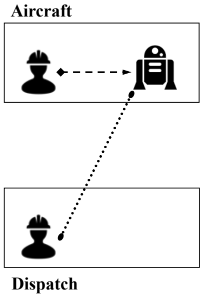
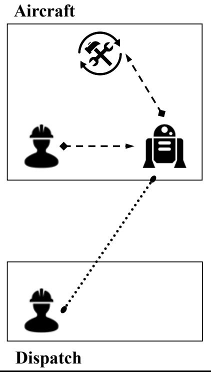
3.1 Initial Use Case 1: Fuel Leak

FLYSKY12 is en route from SFO to BOS. There is one POB and a dispatcher flight following.

In this initial use case, there is an onboard fuel leak. The *Work Objective* is to manage the fuel leak. The *Work Process* consists of the steps necessary to resolve the situation. Here, the output is a divert to an alternate airport. Fig. 1 provides a legend for the steps and links detailed in Fig. 2. The culmination of steps produced our initial RCO design pattern.

Agents	Links
👷 Human Operator	●··············● Communication Only
🤖 Intelligent/Cognitive Agent	◄– – – – – –► Supervisory
🛠 Automated Tools	●━━━━━━━━━━● Cooperative

Fig. 1. Legend for RCO design pattern steps; Cooperative and supervisory links imply communication

Step 1. **Fuel leak in the left fuel tank.** Onboard automation detects fuel imbalance and alerts POB and dispatcher. *This requires a communication link. Automation is shown as a tool because this could be something as simple as a sensor.*	
Step 2. Diagnosis. POB requests automation diagnose fuel imbalance. Automation reports to POB a leak in left tank. *The POB is asking for diagnosis, indicating supervisory control and that the automation has a certain level of intelligence, thus we have redrawn the automation as a cognitive agent.*	
Step 3. Fault management. POB requests that agent manage fuel. Agent opens the cross feed and turns off the pumps in the right side to draw fuel from the left. *The POB is delegating control of the fuel system to the agent. The agent uses onboard tools to accomplish the task.*	

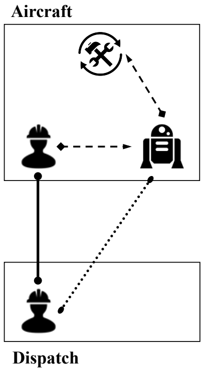
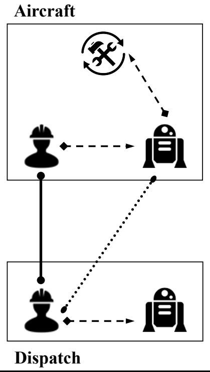
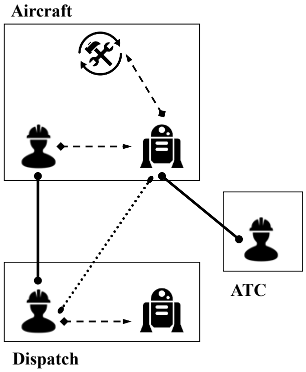
Step 4. Decision to divert. POB contacts dispatch about need to divert. *There is coordination between POB and dispatcher.*	
Step 5. Divert planning. Dispatcher requests divert planning from dispatch automation. *The dispatcher is delegating to the automation. Divert planning automation is shown as an agent because it uses multiple strategies to accomplish the task.* **Step 6. Digital datalink.** Dispatcher uplinks flight plan to POB. POB inspects the flight plan and agrees. *The dispatcher and POB cooperate to agree on the flight plan.*	
Step 7. Execution. POB requests agent coordinate divert with ATC**.** Agent reports divert is approved**.** POB tells agent to execute. *The agent cooperates with ATC. The POB is jointly responsible for safety of flight with ATC. In this case s/he has delegated the responsibility for working with ATC to the agent.*	

Fig. 2. RCO initial use case design pattern steps

4 Use Cases

Having developed a base HAT pattern for RCO, it now needs to be determined if it is general enough to account for other use cases. In this next section, additional use cases are examined.

4.1 Use Case 2: Thunderstorm

Initial Conditions. FLYSKY12 is en route from SFO to ORD. There is one POB and a dispatcher flight following. The *Work Objective* is to avoid a thunderstorm. Again, the *Work Process* consists of the steps necessary to resolve the situation with the output to divert to an alternate airport. Fig. 3 represents the final design pattern.

Step 1. Detection and alerting of thunderstorm. Dispatch automation informs dispatcher of convective cell growing on flight path of FLYSKY12. *This requires a communication link between dispatch automation and the dispatcher (covered by supervisory link in the pattern).*

Step 2. Dispatcher informs POB of cell. *This requires a link between the dispatcher and the POB. This link is as a cooperative link (as in the pattern) because, by regulation, the dispatcher and POB share responsibility for safe operation of the flight (including detecting and responding to thunderstorms).*

Step 3. Modification of flight plan. Dispatcher requests modified flight plan from dispatch automation. Dispatch automation returns modified flight plan. *The delegation of flight path planning to the automation requires a supervisory link. As with the previous use case, this planning requires consideration of multiple strategies making this automation an agent.*

Step 4. Dispatch uplinks modified flight plan. *Uses the link between dispatch and POB from Step 2.*

Step 5. POB requests clearance for flight plan from ATC. *POB and ATC are both responsible for safety of flight and thus this is a cooperative link. This differs from the pattern above where the link from ATC to the aircraft went to the agent.*

Step 6. ATC rejects clearance. ATC tells POB that aircraft must take additional six-minute delay for new arrival slot coming into ORD. *Cooperative link from Step 5.*

Step 7. Planning for delay. POB asks automation for alternatives to take six-minute delay. Automation provides two alternatives: a) Slow down, saves fuel but risks further movement/growth of cell b) Hold past cell, more fuel burn but lower risk of further deviations. *Like Steps 2 and 3 in the previous use case, POB is delegating this task to the automation, requiring a supervisory link. The automation is developing multiple strategies for taking the delay, making it an agent.*

Step 9. POB requests clearance from ATC, modified with holding after passing cell; ATC approves request. *Same cooperative link from Steps 2 and 5.*

Step 10. POB tells agent to implement the new clearance. Agent sets autopilot in accord with clearance. *Once again, the POB delegates tasks to the agent. As in the previous use case, the agent uses tools to perform the task.*

This use case is also well captured by the pattern developed in our initial use case. The only modifications are that in the initial use case the POB delegates negotiation with ATC to the agent, while in this use case he or she negotiates directly and that the onboard automation never communicates directly with the dispatcher. This is perhaps unsurprising since the basic structure of this pattern is specified in our RCO ConOp. The POB and the assigned dispatcher are jointly responsible for the flight, assisting each other in a cooperative relationship. Similarly, the POB and the ATC responsible for the sector of airspace containing the aircraft have complementary roles in assuring safety of flight, and thus must also cooperate. Further, our ConOp specifies that both dispatch and the POB acquire significantly enhanced automation. Thus, in most situations the operators, tools, agents, and their underlying relationships are fixed by our ConOp. Further, at a high level, the *Work Objective* remains constant for RCO: getting the aircraft to the best airport possible for the airline (usually its destination and ideally on time) while maintaining safety of flight. The relevant informational, environmental, and supply inputs also remain constant (although possibly with different weightings) across operations. Of course, a number of specifics could change depending on the situation. For example, while the POB and dispatch are jointly responsible for a flight, it is not necessary that dispatch be contacted in every situation (e.g., if ATC issued a two-mile deviation for traffic).

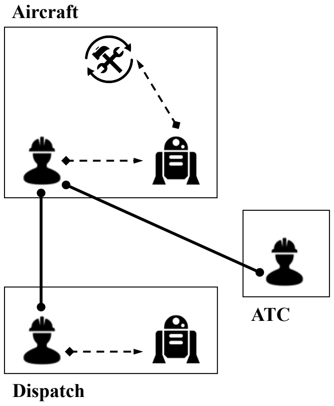


Fig. 3. Use case 2 design pattern

4.2 Use Case 3: Non-cooperative Pilot

There are, however, use cases for which we believe more significant changes to this pattern would be necessary. In particular, pilots may become incapacitated, or, in rare but well publicized instances, become threats to the aircraft themselves (e.g., Jet Blue 191, Germanwings 9525, EgyptAir 990). If we hope to guarantee safety and security in these cases, the dispatcher and possibly the cognitive agent will need significant authority to effectively supervise the POB. Here we outline a possible use case involving such a situation.

Initial Conditions. FLYSKY12 DEN-DCA on final into DCA. There is one POB and a dispatcher flight following. The *Work Objective* is to complete a safe flight with a potentially incapacitated pilot. Again, the *Work Process* is the steps necessary to resolve the situation with the output being the onboard agent assuming the pilot-in-command role. Fig. 4 represents the final design pattern.

Step 1. Pilot takes aircraft off course. At WIRSO (424 feet) POB decouples autopilot and turns north (toward the White House) instead of south. Onboard automation detects deviation from flight plan and alerts POB and dispatcher. *Same communication links from Step 1 of the initial use case.*

Step 2. ATC intervenes. ATC directs POB to correct course. *As with previous use cases, ATC is also responsible for safety of flight, indicating a cooperative link.*

Step 3. Automation calculates point of no return, informs pilot that s/he will be locked out if no corrective action is taken by that time (in this case the time would be minimal; the White House is less than 30 seconds from WIRSO). *As the aircraft approaches restricted airspace, the automation takes on additional authority, marking a change in the relationship between the agent and the POB: the agent is now supervising the POB rather than being supervised by the POB.*

Step 4. POB locked out. No corrective action is taken. The agent locks out POB and alerts the dispatcher. (Dispatch assumed to have the power to return control to POB, but does not here.) *The agent continues its supervisory role relative to the POB.*

Step 5. Agent squawks 777 and corrects course. *The agent has supervisory control over the FMS and other flight deck controls as in the previous use cases.*

Step 6. Establishing link with dispatcher. Agent informs dispatcher of corrective action. *This requires only a communication link between the agent and the dispatcher as in the previous use cases. However, it is presumed that, at this point the agent would either form a cooperative link with the dispatcher or the dispatcher would take over supervision of the agent.*

In this use case, major changes were needed to accommodate the events. Yet this change is only reflected in the subtle change from the POB and the cognitive agent cooperating with the agent supervising the POB, requiring that the agent have a great deal of autonomy and authority. Intuitively, this change in authority is a major change in the pattern; however, the level of authority (although implied through delegation) has not been an explicit component of these design patterns. Perhaps to fully explore some of this design space, it will need to be added.

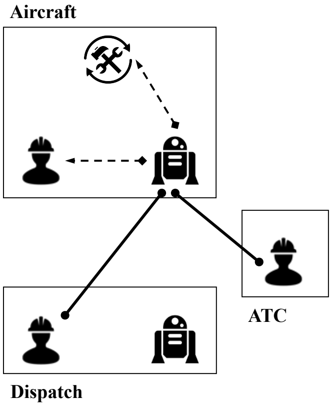


Fig. 4. Use case 3 design pattern

5 HAT Measures

Our HAT design patterns effectively captured the similarities between our first two use cases, as well as the structural differences between those and the final use case with the non-cooperative pilot. However, the reader may be concerned that in developing these patterns we have made a number of arbitrary distinctions. In fact, we spent considerable time debating which piece of automation was a cognitive agent versus a tool and the type of connections between the various agents. To aid in making these decisions, we turned to a number of metrics that have been used by researchers in related fields. Specifically, we looked at the degree to which agents exhibited situation awareness indexed by the levels of situation awareness described by Endsley [10], the management capabilities of agents as indexed by Sheridan's levels of automation [11], and decision-making ability assessed using the Non-Technical Skills framework (NOTECHS) categories [12]. In addition to categorizing the decision-making authority of the agent, NOTECHS have several other scales that allow us to assess whether communications links involve Management, Joint Decision making, and/or Cooperation.

We used these measures to give a more quantitative assessment of the automation in our initial use case. In Fig. 5, we re-examined each step, giving the reasoning behind these assessments. A new design pattern was drawn in Fig. 6 to reflect the additional measures.

Step 1. **Fuel leak in the left fuel tank.** Onboard automation detects fuel imbalance and alerts POB and dispatcher.	No CRM skill indicated
Step 2. Diagnosis. POB requests automation diagnose fuel imbalance. Automation reports to POB a leak in left tank.	Management skill in a supervisory relationship between POB and agent (labeled M), with the agent having a capability to identify problems (Decision Making level one, labeled D1)
Step 3. Fault Management. POB requests that agent manage fuel. Agent opens the cross feed and turns off the pumps in the right side to draw fuel from the left.	Management in a supervisory relationship between agent and aircraft (labeled M), with the agent informing the POB only if necessary (Sheridan management level nine, labeled M9)
Step 4. Decision to divert. POB contacts dispatch about need to divert.	A relationship between POB and dispatcher with Cooperation (awareness of needs of other), Decision Making (eliciting divert options), and Management (delegating divert location task to the dispatcher), labeled C/D/M

Step 5. Divert planning. Dispatcher requests divert planning from dispatch automation.	Management in a supervisory relationship between dispatcher and agent (labeled M), with the agent having a Decision Making ability to select an option (NOTECHS level three, labeled D3)
Step 6. Digital datalink. Dispatcher uplinks flight plan to POB.	Continued relationship between POB and dispatcher with Cooperation (awareness of needs of other), Decision Making (eliciting divert options), and Management (delegating divert location task to the dispatcher), labeled C/D/M
Step 7. Execution. POB requests agent coordinate divert with ATC. Agent reports divert is approved. POB tells agent to execute.	Management in a supervisory relationship between POB and agent (labeled M). Cooperative Decision Making between the agent and ATC (labeled D), with the agent having a Decision Making ability to select an option (NOTECHS level three, labeled D3)

Fig. 5. Re-examination of RCO design pattern using additional measures

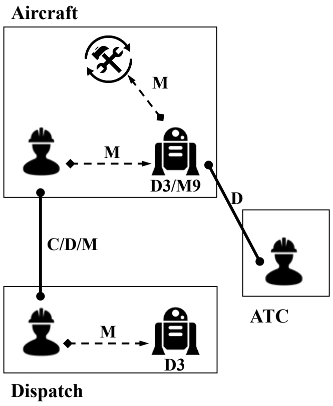


Fig. 6. RCO design pattern using additional measures

6 Discussion

In this relatively new field of human-autonomy teaming, this analysis suggests that defining design patterns may help describe and prescribe human-autonomy relationships. This paper attempts to build a pattern to describe the human-autonomy teaming resident in a reduced crew operations design. Building on a use case, researchers were able to identify a basic pattern (see Fig. 1). This pattern describes the agents (human and otherwise) and the logical connections between them. Once this base pattern was defined, new use cases evaluated its generality. For nominal and routine off-nominal operations, most of the relationships were captured. However, for extreme use cases (non-cooperating pilot), new relationships needed to be added. Further, it became clear that the dimension of authority needed to be added to fully describe the environment. It may be that a family of patterns is required to fully describe complex situations in all contexts, but that a single basic pattern may suffice for normal operations.

A challenge with the present exercise was getting the correct level of detail in defining HAT design pattern elements. Highly general elements can gloss over critical distinctions, while narrowly defined elements can result in overly complex and hard to generalize patterns. The cooperative link is a case in point. In this exercise we realized that there were two varieties of this link, one reflecting relatively unstructured *collaboration* between humans and/or agents working on a single task (e.g., a ground operator and a pilot simultaneously and collaboratively searching for the best divert airport); and one reflecting a more structured *coordination* while working on separate subtasks (e.g., an automated agent developing a list of divert options and an operator culling this list while generating new criteria/constraints for the automated agent). Answering whether or not to include distinctions such as these will probably require multiple efforts to generate and apply HAT design patterns.

We have seen that a design pattern (or family of design patterns) can be used to describe this environment and its relationships, but how can it be of use? One way is to re-use patterns when developing system designs in other (new) environments. We can use patterns to prescribe the relationships and level of automation required to achieve design goals. Design patterns can also be used in a diagnostic manner. The non-cooperating pilot example showed that the cognitive agent needed a higher degree of authority than had previously been assigned. Exercising the use cases can determine if a pattern (or existing system) is able to execute that case and can diagnose where additional autonomy and/or authority might be required.

Members of NATO HFM-247 working group are applying this methodology to their individual projects to determine the overall generalizability and utility. If successful, this may be a significant step forward in understanding human-autonomy teaming.

Acknowledgements. This study was supported by the NASA Safe and Autonomous System Operations project and Reduced Crew Operations subproject. We would like to thank Dr. Walter Johnson for his support and insightful edits.

References

1. Alexander C., Ishikawa S., Silverstein M.: A Pattern Language: Towns, Buildings, Construction. Oxford University Press (1977)

2. Beck K., Cunningham W.: Using Pattern Languages for Object-Oriented Programs*. In: Proceedings of* OOPSLA *'87, Specification and Design for Object-Oriented Programming* (1987)

3. Comerford, D., Brandt, S.L., Wu, S.-C., Mogford, R., Battiste, V., Johnson, W.W.: NASA's Single Pilot Operations Technical Interchange Meeting: Proceedings and Findings, Report No. NASA-CP-2013-216513, NASA Ames Research Center, CA (2012)

4. Lachter, J., Brandt, S.L., Battiste, V., Ligda, S.V., Matessa, M., Johnson, W.W.: Toward Single Pilot Operations: Developing a Ground Station. In: Proceedings of the International Conference on Human-Computer Interaction in Aerospace, Santa Clara, CA (2014)

5. Brandt, S.L., Lachter, J., Battiste, V., Johnson, W.W.: Pilot Situation Awareness and Its Implications for Single Pilot Operations: Analysis of a Human-In-The-Loop Study. In: Proceedings of the 6th International Conference on Applied Human Factors and Ergonomics. Las Vegas, NE (2015)

6. Bilimoria, K.D., Johnson, W.W., Shutte, P.C.: Conceptual Framework for Single Pilot Operations. In: Proceedings of the International Conference on Human-Computer Interaction in Aerospace, Santa Clara, CA (2014)

7. Schulte, A.: Human Systems Engineering (HSE) Approach to Human-Autonomy Teaming, NATO STO HFM-247 Task Group Presentation (2015)

8. Schulte, A., Donath, D., Lange, D.S.: Design Patterns for Human-Cognitive Agent Teaming. In: Conference on Engineering Psychology & Cognitive Ergonomics. Lecture Notes on Computer Science, Springer (2016)

9. Neerincx,M., Lange, D.: HAT Patterns, NATO STO HFM-247 Task Group Presentation (2015)

10. Endsley, M.R.: Toward a Theory of Situation Awareness in Dynamic Systems. Hum. Factors 37, 32–64 (1995)

11. 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 Transactions, 30(3), 286-297 (2000)

12. Flin, R., Martin, L., Goeters, K.-M., Hörmann, H.-J., Amalberti, R., Valot, C., Nijhuis, H.: Development of the NOTECHS (Non-Technical Skills) System for Assessing Pilots' CRM Skills. Hum. Factors Aero. Safety 3, 95-117 (2003)