Designing a Support System to Mitigate Pilot Error While Minimizing Out-of-the-Loop-Effects

Nikolaus Theißing(✉) and Axel Schulte

Institute of Flight Systems,
Universität der Bundeswehr München, Neubiberg, Germany
{nikolaus.theissing,axel.schulte}@unibw.de

Abstract. This article describes a design pattern for human-autonomy teaming in which a human operator is supported by two cognitive agents. The first agent automates delegated tasks as a subordinate in a hierarchical relationship. The second agent serves as an assistant system. Its task is the mitigation, i.e. prevention and correction, of human erroneous behavior. Error-correcting automation can be prone to complacency effects. The assistant system is therefore specifically designed to avoid potential out-of-the-loop phenomena. The design pattern is implemented for the use case of unmanned air reconnaissance using a single-operator ground control station. The results of an experimental campaign confirm the validity of the approach.

Keywords: Human-autonomy teaming · Human error · Error mitigation · Assistant system · Cognitive agent · Complacency

1 Introduction

Complex automation in the aviation domain can improve the efficiency of flights and enable new fields of application. It does not, however, necessarily protect the human-machine-system from human erroneous behavior. Dangerous or otherwise sub-optimal commands can still cause unintended behavior of the underlying automation, especially in a supervisory-control-relationship. A high degree of automation may even increase the risk of erroneous behavior by introducing effects of automation-induced error [1, 2].

The challenge is now to employ automation in the human-machine-system that brings the desired benefits (increased effectivity and efficiency) and minimizes the amount of erroneous behavior of the pilot.

In this article, a design pattern is described that uses two distinct automation systems: A subordinate automation operated in a supervisory-control-relationship [3] is complemented by an assistant system that is designed to support the human pilot during the mission. The assistant system shall mitigate erroneous behavior of the pilot by intervening in dangerous situations or when the underlying automation is not operated properly (by harmful commands or command omissions). Alerts, messages, suggestions,

D. Harris (Ed.): EPCE 2016, LNAI 9736, pp. 439–451, 2016.
DOI: 10.1007/978-3-319-40030-3_43

and overrides are then used by the assistant system to help the pilot transfer the current (dangerous) situation to a normative (safe) one.

When designing an assistant system of the described kind, it is a challenge not to induce human erroneous behavior. A system that corrects errors immediately and reliably may cause complacency effects in the pilot: The pilot may put an overly high trust in the automation, therefore neglect his or her own monitoring tasks and, as a consequence, lose vigilance or situation awareness [4].

To avoid such effects, the assistant system must be designed specifically against the induction of such out-of-the-loop-effects [5]. This article proposes a technique for assisting the pilot step by step and thereby keeping him or her vigilant with respect to the task as much as possible. The assistant system described in the design pattern is supposed to mitigate erroneous behavior on the one hand and to keep the unwanted out-of-the-loop effects as low as possible on the other hand.

The design pattern will be defined in the following section. The subsequent section will describe the application of the design pattern to the domain of unmanned reconnaissance flights. After that, the results of an experimental evaluation will be shown.

2 Design Pattern "Step-by-Step Error Correction"

2.1 Separation of Two Automation Units

A key element of the design pattern is the separation of the automation used to operate the vehicle from the automation supporting the pilot.

Figure 1 shows the configuration of the work system. The work system notation [6] differentiates between the set of worker on the left-hand side and the tools on the right-hand side. Among other aspects, a worker is characterized by the access to the overall mission goal and the authority to modify that goal and use the available tools to achieve it. The tools, on the other hand, are subordinate to the worker.

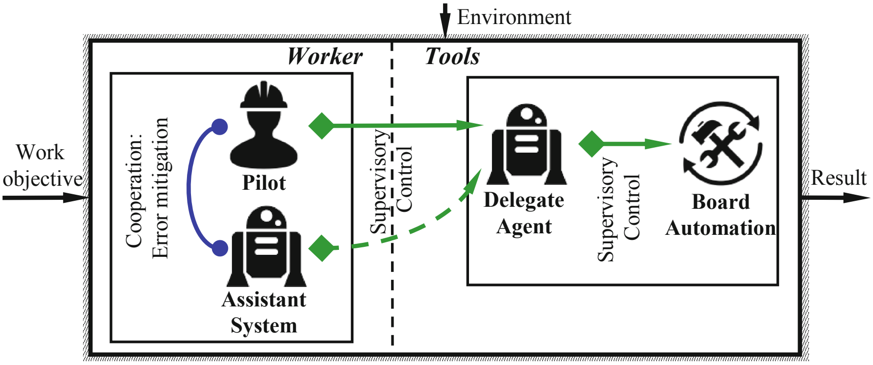

Fig. 1. The basic elements of the design pattern in the work system notation [6]

In the described design pattern the pilot, as a worker, operates the underlying automation in a supervisory-control-relationship. The underlying automation consists of a subordinate cognitive unit, referred to as the *delegate agent*, operating the conventional board automation. In addition to the human pilot, a second cognitive unit is located within the group of workers. That cognitive unit, referred to as *the assistant system*, supports the pilot during the mission management process. Onken and Schulte introduced this configuration as one which fully exploits the options of 'dual-mode cognitive automation' [7].

Each cognitive unit has a distinct purpose:

- Task of the delegate agent: Control of the conventional automation, and thereby, reduction of human taskload
- Task of the assistant system: Mitigation, i.e. prevention or correction, of erroneous behavior of the pilot

Whereas the delegate agent is subordinate to the human pilot (it does what it is told), the assistant system shall work in a cooperative way (without a hierarchical gap, on its own initiative, in pursuit of the overall mission goal). The clear separation between the underlying automation and the cooperative support system offers the following advantages:

- The aircraft is operable even with a complete assistant system shutdown. In case of malfunction the assistant system can, as a last resort, be safely deactivated.
- The board automation of an existing aircraft needs not (or only with minor changes) be modified but only complemented.
- The assistant system and the board automation may be physically detached.
- The functional separation may facilitate the development and certification process.

The pilot controls the underlying automation via a human-machine-interface (HMI) that is not explicitly depicted in the figure. Whereas the assistant system can be built to interact with the pilot in an arbitrary way, it is typically integrated into the existing interface.

2.2 The Role of the Delegate Agent

The role of the delegate agent is the control of the conventional automation of the controlled vehicle. The delegate agent is controlled by the human supervisor and, in turn, controls the underlying conventional automation in another supervisory relationship [8]. This layer of agent supervisory control enables the human pilot to delegate certain higher cognitive tasks (i.e., planning, scheduling, decision making) to the agent. This provides the pilot an automation span of control beyond that of the conventional board automation.

Delegating the control of the conventional automation to an agent can yield the following advantages:

- Independence from an active data connection: The delegate agent is usually controlled with single, discrete commands (as opposed to e.g. the immediate control exerted by flying via stick and throttle). A disruption of the data connection leaves the aircraft functional, albeit with old commands.

- Reduction of workload: The possibility to delegate even higher cognitive tasks to the machine enables the pilot to free his or her mental resources for other tasks.

The role of the delegate agent is to do what it is told without questioning its commands. Of course, depending on the design of the agent, detailed feedback to the pilot may be used to indicate abnormal situations e.g. via alerts or warnings. Also, the given commands may (implicitly) contain directives on how to react to certain situations. The agent will, however, be bound to the given commands. The rationale behind this behavior is to ensure that the pilot is always in full control of his or her vehicle. This implies that the agent will execute erroneous commands regardless of the resulting danger. The prevention and correction of such commands is the task not of the subordinate agent but of the assistant system.

2.3 The Role of the Assistant System

The delegate agent attempts to execute the given commands. It is the task of the human pilot to provide correct commands to the agent at any given time. If the given commands are erroneous, the behavior of the agent is likely to be erroneous, too.

The role of the assistant system is to mitigate erroneous behavior, i.e. to prevent it from occurring or, if that is too late, attempt to correct it and minimize its effects.

Erroneous behavior of the pilot can be classified in two types: Either the pilot gives a wrong command to the agent (error of commission) or the pilot fails to give a command to the agent although it would be necessary (error of omission) [9].

A command omission can be prevented by making sure that the right command is given to the agent in time. A wrong command can (assuming that its effects are not immediately disastrous) be corrected by giving a correcting counter-command to the agent in time. The resolution to both classes of erroneous behavior (wrong commands and omitted commands) can therefore be condensed into one principle: Ensure that a certain command is given to the agent in time.

Goal and Constraints. The assistant system shall attempt to keep up and restore, if necessary, the following goal state:

- Goal: At any given time the automation controlled by the pilot (the agent) is provided with correct commands.

That goal is constrained by the requirement that the human pilot, and not the assistant system, shall be the primary entity in charge. One reason, in addition to the legal and moral responsibility for the vehicle, is the superior knowledge of the human pilot: The pilot can be assumed to be the most valuable and reliable source of knowledge, decisions, and initiative in the system. To keep these assets available, the cognitive resources of the pilot (e.g. situation awareness, vigilance) have to be protected from negative influences such as out-of-the-loop effects.

Therefore the assistant system shall pursue its goal under the following constraints:

- Constraint 1: The given commands shall reflect the intent of the human pilot as closely as possible.
- Constraint 2: The performance of the human pilot regarding his or her cognitive resources shall be kept as high as possible.

Requirements. The constraints require the assistant system to behave in a way that will, on the one hand, eventually resolve the dangerous situation. On the other hand, the human pilot must be involved as much as possible to keep him or her in the control loop. In essence, the assistant system shall intervene if necessary, but the share of work contributed by the human pilot shall be as high as possible.

The following requirements to the behavior of the assistant system can be derived:

1. The human pilot shall be given as much time as possible to find own solutions.
2. Dangerous situations shall be resolved before damage is inflicted.
3. Interventions shall provide input that helps with the current problem.
4. The input given by an intervention shall not exceed the current problem.

The requirements call for an escalating behavior of the assistant system. Schulte demands that an assistant system behave according to an escalating scheme: The assistant system shall let the pilot do his or her tasks without intervening. Only if necessary shall the system successively guide, relieve, and – if everything fails – override the pilot [6].

This scheme will reflect in the following strategy that implements the given requirements and shall offer precise rules for the behavior of the assistant system that can be used for a practical implementation.

Strategy. At any given moment (e.g. in each computation cycle) the assistant system shall act according to the following strategy:

1. Determine if the current situation is dangerous and, if so, at what time damage (a violated threshold of certain performance parameters) will be inflicted. A situation is dangerous if the further development, without intervention by the pilot or the assistant system, leads to damage.
2. Determine what the pilot must do to resolve the dangerous situation. The resolution typically consists of giving a certain command (sequence) to the delegate agent. This action is the desired action that will be enforced by the assistant system.
3. Estimate the current cognitive state of the pilot. The cognitive state includes mental resources such as situation awareness, vigilance, workload, and focus. It also includes the state of information processing, i.e. the current task and the associated cognitive processes. This estimate should be based on a model of the pilot's information processing.
4. Compute the transitions of the pilot's cognitive state leading from the current state to the resolution of the dangerous situation, and identify the conditions for each transition: What steps will the pilot's mind have to go through to effect the desired action, beginning with its current state? These steps and estimates of their duration

(with buffers and worst-case assumptions) should be based on a model of the pilot's information processing.

5. Arrange these computed mental steps along a timeline, beginning with the rightmost one. Arrange them in a way that the final step (the desired action) takes place immediately before the moment of damage, i.e. barely in time. The position of the left end, i.e. the starting point of the sequence, will then determine the point in time at which the pilot has to begin working on the problem.
6. Determine: Is the starting point of the sequence in the future?
   (a) Yes: There is still time left for the pilot to find own solutions. Do nothing.
   (b) No: The pilot should have reacted by now. Intervene by enforcing the current transition, i.e. the first step, using any available means.

Figure 2 shows an illustration of this strategy for a simple example. New commands have to be entered in time to avoid a collision with another aircraft (1,2). The pilot is currently analyzing the tactical map (3). The sequence of thought and action leading from this current task to the desired action step (4,5) will likely not be completed in time (6a). Therefore, the assistant system intervenes (6b): It enforces the transition from the pilot's current mental state to the next state (the detection of a relevant change in the tactical environment) by alerting the pilot about an incoming other aircraft.

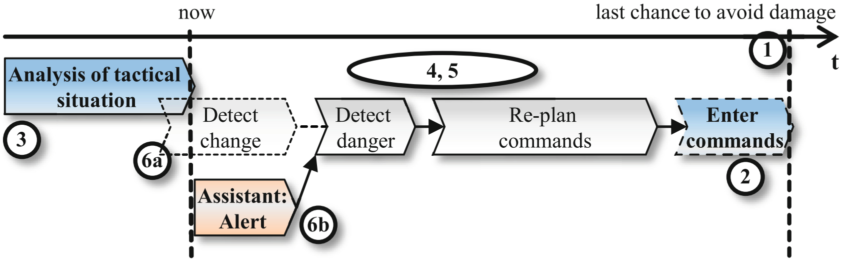

Fig. 2. The strategy of the assistant system for planning and scheduling its interventions

2.4 Functional Architecture of the Assistant System

To implement the shown intervention strategy the assistant system needs the following capabilities:

- Monitoring of the environment and detection and analysis of danger
- Monitoring of the pilot and interpretation of the observed data with respect to the pilot's cognitive state
- Planning and scheduling of interventions
- Execution of interventions, i.e. of the actual interaction with the pilot

Figure 3 shows the functional system architecture of the assistant system as a network of modules implementing the capabilities and exchanging the respective data.

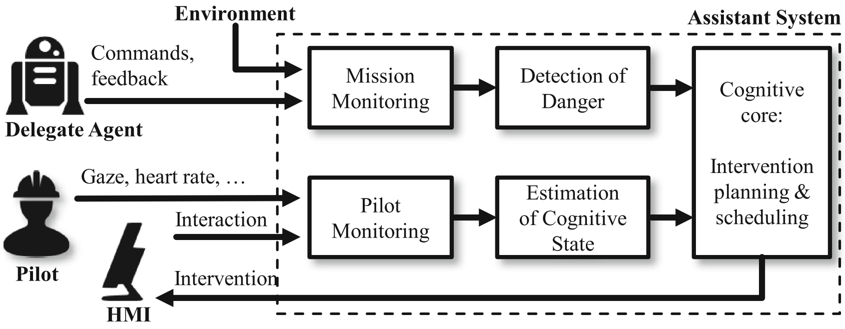

Fig. 3. The inputs, outputs and processing components of the assistant system

3 Application to Unmanned Air Reconnaissance

This section will provide an exemplary use case in which the described design pattern has been practically implemented and evaluated. The pattern has been applied to the domain of unmanned air reconnaissance conducted by a single human pilot.

3.1 Domain Description

In the given work domain it is the task of a single human pilot to gain reconnaissance information on certain objects (buildings, persons, vehicles) in an area that may possibly contain hostile forces. The information can be obtained by interpreting imagery and video data gathered by sensors that are attached to an unmanned aircraft. A single human pilot has the task to manage the flight control and sensor control of that aircraft from a ground control station.

The reconnaissance targets are given to the pilot beforehand, but may change dynamically during the mission. The execution is constrained by airspace regulations (boundaries and corridors), threats (possibly unexpected hostile air defenses), and resource limitations (fuel). The pilot has to carry out the tasks of flight management, sensor management and interpretation of the sensor data in parallel. The pilot is therefore supported by automation applied according to the described design pattern.

3.2 System Architecture

Figure 4 depicts the system architecture of the human-machine-system. In the ground control station the human pilot commands and controls the aircraft via a graphical user interface (GUI). The assistant system shares that GUI to monitor the pilot's interaction with the system and to intervene if necessary. The pilot and the assistant system exert supervisory control over the delegate agent referred to as *decision engine*. The decision engine conducts the flight and sensor control of the aircraft according to tasks received from the ground control station. The air and ground segment communicate via an air data link.

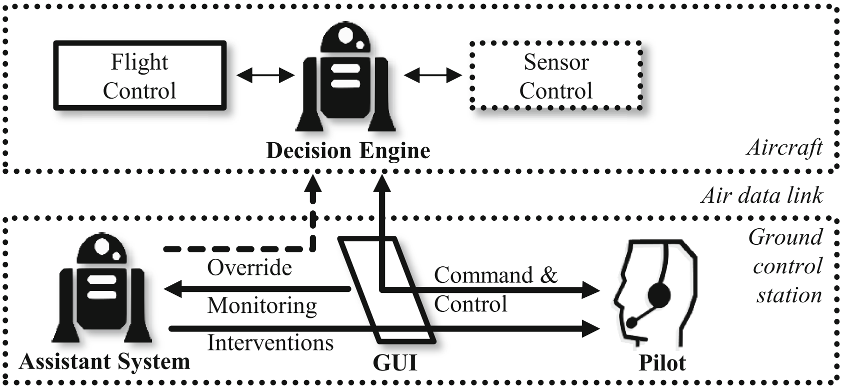

Fig. 4. The functional architecture of the unmanned reconnaissance system

3.3 Decision Engine

The decision engine implements the principle of task-based guidance [8]. Instead of detailed manual control commands, the human pilot can assign a task or a sequence of tasks to the decision engine and monitor the execution. A task is an abstract high-level command describing an action and, if necessary or desired by the pilot, parameters. Examples are “Land at the home base”, “Find vehicles in area X”, or “Find vehicles in area X using manual sensor guidance”. A sequence of such tasks, defining the actions to be carried out, can be commanded to the decision engine.

After receiving a new or modified task sequence the decision engine plans the execution of the tasks by complementing missing steps and breaking the tasks down to elementary operations. It then executes the tasks and provides the human pilot with feedback about the currently processed task, its execution state, and the current route.

3.4 Assistant System

The assistant system shall prevent the effects of erroneous behavior of the human pilot. These effects can be, among others:

- Violation of airspace regulations by the aircraft
- Loss of aircraft by exhaustion of fuel reserves during the flight
- Loss of aircraft by entry into the threat radius of hostile air defense sites
- Ineffective reconnaissance (inadequate fulfillment of the mission objective)

To avoid these effects, the assistant system can intervene. It is integrated into the control station’s systems and has access to the pilot’s GUI. Depending on the conditions of the mental step the pilot shall be supported with, the assistant system can display general alerts and iconic or textual messages, highlight certain screen elements, direct the pilot’s attention to other screens, or override commands.

The assistant system obtains the information necessary to plan, schedule and execute an intervention from the subsystems of the GUI. Figure 5 depicts the system architecture of the assistant system. Information about the tactical environment and the state of the decision engine is analyzed for dangerous situations according to a model of the mission domain. A model of the pilot's behavior is employed in the component inferring the pilot's gestures from the observed input and estimating the pilot's mental state. The estimation component uses a Colored Petri Net [10] as an on-line simulation of the pilot's behavior. The cognitive core plans and schedules the interventions according to the strategy described above based on another model of the pilot's behavior. The NDDL-based framework EUROPA [11] is used for the implementation of the planning and scheduling process.

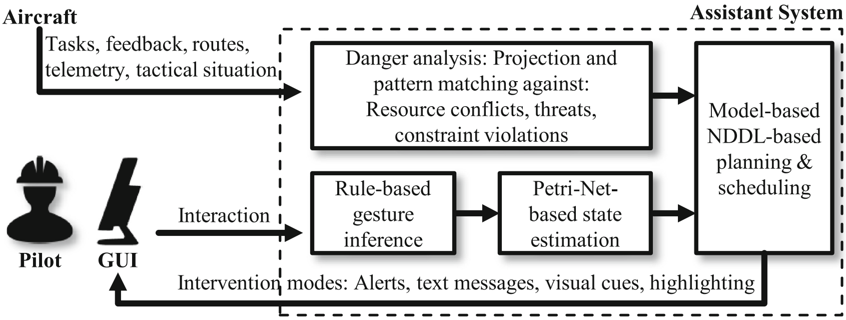
Aircraft
Assistant System
Tasks, feedback, routes, telemetry, tactical situation
Danger analysis: Projection and pattern matching against: Resource conflicts, threats, constraint violations
Model-based NDDL-based planning & scheduling
Rule-based gesture inference
Petri-Net-based state estimation
Interaction
Pilot
GUI
Intervention modes: Alerts, text messages, visual cues, highlighting

Fig. 5. Components and information flow of the implemented assistant system

4 Experimental Evaluation

To evaluate the effectiveness of the design pattern, a human-machine experimental campaign was conducted. The experiment had two goals:

- Show that the assistant system reduces the negative impact of human erroneous behavior on the overall mission performance. This is the primary system goal.
- Show that the intervention strategy fulfills its requirements. This shall justify the usage of the described strategy. After all, a system with a simpler strategy (e.g. 'override immediately upon danger') may yield the same results concerning mission performance at first, but have negative impact on the pilot's mental state.

4.1 Setup

In the experiment a group of test subjects conducts a series of reconnaissance missions with the described system in the role of the pilot. The missions are simulated (flight dynamics, tactical environment and sensor imagery are generated by a virtual environment), but the real hard- and software of the ground control station is used.

After the introduction to the system and a training mission, each subject conducts three missions. The first and third missions are carried out with a deactivated assistant system (as a baseline configuration A) whereas during the second mission, the assistant system is active (configuration B). This ABA-configuration is used to average out the influence of training effects by comparing B to the average of A measured before and after.

The missions are designed to be similar enough to be comparable to each other but sufficiently different to avoid habituation effects on the subjects. The objective of each mission is the reconnaissance of certain areas. Ancillary tasks are the monitoring of the tactical map, the detection of targets of opportunity, and the conduction of radio dialogues. These tasks are constrained by threats and resource limitations as described above.

The tasks and constraints are chosen to generate very difficult missions. The reason is that the assistant system only acts in situations of danger, which should be a rare exception in normative work situations. Therefore, in this experiment dangerous situations are artificially created. The tasks given to the subjects are time-consuming, demand multi-tasking, and create a high mental workload. During the missions, the tactical situation changes dynamically. The changes include spontaneous threats and blocked airspaces. The resulting high degree of difficulty is supposed to evoke a lot of erroneous behavior in the subjects.

The investigated hypotheses state that the loss of mission performance (i.e. the damage inflicted by erroneous behavior) is reduced by the application of the assistant system, whereas the performance in the primary reconnaissance task, the workload, and the situation awareness of the pilot will generally be unaffected.

The dependent variable of performance loss is operationalized by a penalty score for airspace violations, resource limit violations and neglected threats. The situation awareness is determined by SAGAT questionnaires [12] after each mission and an evaluation of the subjects' behavior during the missions. The workload is represented by the result of NASA-TLX questionnaires [13]. The task performance is determined by a score for the detection and classification of reconnaissance targets and targets of opportunity.

The duration of the experiment is too short to gain evidence about the influence of the assistant system on the pilot's cognitive state in the long term. The experiment does, however, allow the investigation of the behavior of the assistant system (does it do what it is supposed to?) and the reactions of the subjects.

4.2 Results

The described experiment was conducted with a group of 17 test subjects comprising officers and cadets of the German Armed Forces. Each subject had several years of military experience and either an academic background or practical experience (as a pilot or unmanned system operator) in aviation.

Figure 6 shows the measured values for the dependent variables. As the graphs suggest, the performance loss (in the upper right graph) is the only measured variable with a significant change between the configurations. With an active assistant system (mission 2) the measured performance loss is significantly lower than that observed without assistance (missions 1 and 3). The remaining investigated variables cannot be distinguished with statistical significance [14].

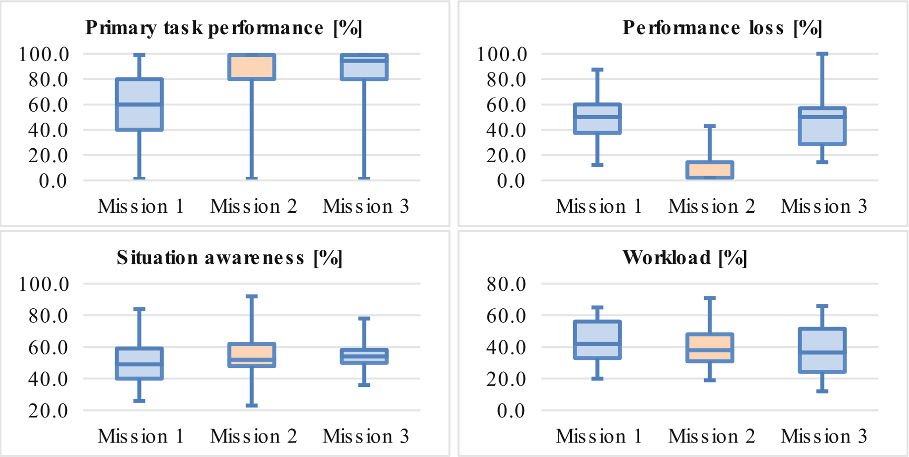

Fig. 6. Box-Whisker-plots of the measured variables comparing the configurations without assistance (missions 1 & 3) to those with an active assistant system (mission 2)

The hypotheses stated above can therefore be accepted: The application of the assistant system was shown to lower the performance loss whereas no significant effects on the remaining variables could be observed.

During the experimental campaign, 31 interventions of the assistant system were encountered. An analysis of the behavior of the assistant system and the reaction of the subjects shows that in 42 % of the cases the assistant system had to escalate to the final level of overriding the pilot to avoid damage. In the remaining cases, a lower number of interventions were necessary. 36 % of the situations only required the first two steps (general warning and highlighting of a changed element) for the pilot to find the rest of the solution on his or her own. It can be assumed that in most of those cases a static level of intervention would have been either unnecessarily explicit (giving the pilot more information than necessary) or insufficient for solving the problem.

5 Conclusion

A design pattern for human-autonomy teaming has been presented that combines the delegation of functionality to underlying automation with cooperative support by an assistant system. An experimental campaign was conducted to evaluate an implementation of the design pattern. The functionality of the assistant system has been verified regarding the desired immediate effect of error mitigation and the immediate impact of the intervention strategy on the pilot's behavior has been investigated.

Further research will have to provide evidence that the intervention strategy of the assistant system has the desired effect on the pilot's mental state in the long term. The current experiment could only monitor the immediate effects on the subjects' behavior. A long-term study will have to show that the effects of this assistant strategy sustain the pilot's vigilance and situation awareness more than simple strategies (e.g. 'warn immediately') would.

An issue not addressed in this article is that of the functional limitations of the assistant system. In situations for which the system is not designed an intervention may be counterproductive. As a remedy, the assistant system has to employ knowledge about its own limitations. It needs the capability to detect if the current situation is still within its scope. If not, it needs to react appropriately, e.g. by notifying the pilot and then remaining silent.

The presented design pattern originated from the domain of aviation. The concept can, however, be transferred to any work environment in which a human operator supervises complex automation. Suitable work domains include (semi-)autonomous driving with a driver assistant system or industrial operation of complex machinery.

References

1. Bainbridge, L.: Ironies of automation. Automatica. **19**, 775–779 (1983)
2. Sarter, N.B., Woods, D.D., Billings, C.E.: Automation surprises. In: Salvendy, G. (ed.) Handbook of Human Factors & Ergonomics, 2nd edn, pp. 1–25. Wiley (1997)
3. Sheridan, T.B.: Task analysis, task allocation and supervisory control. In: Helander, M.G., Landauer, T.K., Prabhu, P.V (eds.) Handbook of Human-Computer Interaction, pp. 87–103. Elsevier Science B. V. (1997)
4. Parasuraman, R., Molloy, R., Singh, I.L.: Performance Consequences of Automation-Induced 'Complacency'. Int. J. Aviat. Psychol. **3**, 1–23 (2009)
5. Kaber, D., Endsley, M.: Out-of-the-loop performance problems and the use of intermediate levels of automation for improved control system functioning and safety. Process Saf. Prog. **16**(3), 126–131 (1997)
6. Schulte, A., Donath, D., Lange, D.S.: Design patterns for human-cognitive agent teaming. In: Harris, D. (ed.) EPCE 2016. LNCS(LNAI), vol. 9736, pp. 231–243. Springer, Heidelberg (2016)
7. Onken, R., Schulte, A.: System-Ergonomic Design of Cognitive Automation. Springer, Heidelberg (2010)
8. Clauß, S., Schulte, A.: Task delegation in an agent supervisory control relationship capability awareness in a cognitive agent. In: 2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 825–830. IEEE (2014)
9. Hollnagel, E.: Looking for errors of omission and commission or The Hunting of the Snark revisited. Reliab. Eng. Syst. Saf. **68**, 135–145 (2000)
10. Jensen, K., Kristensen, L.M.: Coloured Petri Nets: Modelling and Validation of Concurrent Systems. Springer, Heidelberg (2009)
11. Barreiro, J., Boyce, M., Do, M., Frank, J., Iatauro, M., Kichkaylo, T., Morris, P., Ong, J., Remolina, E., Smith, T., Smith, D.: EUROPA : a platform for AI planning, scheduling, constraint programming, and optimization (2004)
12. Endsley, M.R.: Situation awareness global assessment technique (SAGAT). In: Proceedings of the National Aerospace and Electronics Conference (NAECON), New York, pp. 789–795 (1988)

13. Hart, S.G., Staveland, L.E.: Development of NASA-TLX (Task Load Index): results of empirical and theoretical research. Adv. Psychol. **52**, 139–183 (1988)
14. Theißing, N., Liegel, A., Schulte, A.: Verhindern von Pilotenfehlern durch ein zustandsadaptives Assistenzsystem. In: Grandt, M., Schmerwitz, S. (eds.) Kooperation und kooperative Systeme in der Fahrzeug- und Prozessführung, pp. 97–114. Deutsche Gesellschaft für Luft- und Raumfahrt - Lilienthal-Oberth e.V, Bonn (2015)