A Design Pattern for Working Agreements in Human-Autonomy Teaming

Robert S. Gutzwiller(✉), Sarah H. Espinosa, Caitlin Kenny, and Douglas S. Lange

Space and Naval Warfare Systems Center Pacific, 53560 Hull Street, San Diego, CA 92152, USA
{gutzwill, shunt, ckenny, dlange}@spawar.navy.mil

Abstract. Humans and machines are increasingly reliant upon each other in complex environments and military operations. The near future suggests human understanding of machine counterparts is a required, paradigmatic element. Knowing how to engineer and design for these environments is challenging. The complexity between levels of automation, human information processing, and function allocation authority issues in an adaptive system make it unlikely to find a "one-size-fits-all" approach. There may still be general strategies for engineering in these cases; for example, collaborating and coordinating are familiar requirements of all human team activities, and extend to human-automation teaming. Here, we outline what we believe is one so-called "design pattern" for working agreements. We use the loose structure of prior software design patterns to organize our thoughts on why working agreements are necessary, where and how they are applicable, what instantiating them requires, and how to measure their effectiveness. By choosing the design pattern structure, we end up carefully describing what might work best and what the limits are toward improving human-machine teaming.

Keywords: Human-machine teaming · Working agreements · Design patterns · Human automation interaction

1 Introduction on Working Agreements

In the modern era of software development, many problems are recurring. Fortunately, these problems have repeatable solutions. The accepted format for conveying these solutions has been termed a "design pattern" [1, 2], and can be applied across the spectrum of software engineering, including implementation and structural design [3]. We believe these can be extended in spirit to describing human-autonomy teaming patterns.

Ostensibly, these patterns are useful for multiple reasons. First, they provide the mid-level conceptual representations for common problems. These allow for a sharing mechanism among programmers and designers that focuses less on specifics and more on generalities. Second, the patterns are not so high-level as to be useless in application: having viewed one pattern, a programmer can begin to work, whereas high-level representations require more effort and investment to flesh out. Patterns may help define key hardware and software requirements as well.

D.N. Cassenti (ed.), *Advances in Human Factors in Simulation and Modeling*,
Advances in Intelligent Systems and Computing 591, DOI 10.1007/978-3-319-60591-3_2

These representations are common in software design, but not yet in the interaction between humans and autonomy. However, recently there have been several efforts to create patterns for methods to solve human-autonomy teaming challenges [4, 5]. Indeed, this method may help bridge disparate research efforts by outlining general solutions to problems plaguing human-autonomy partnerships. The lineage of design patterns suggests that human-autonomy teaming patterns should focus on particular techniques, their intention, motivation, applicability, participants, consequences, implementation, known uses, and related patterns.

In this first attempt at a design pattern, are interested in an emerging method for man-machine teaming, working agreements [6]. We use the format of the chapters from Gamma [3] in the next sections where they apply, borrowing from the software community and using it to develop the conceptual representation of working agreements more fully.

1.1 Background and Intent

Working agreements define how and when human-automation teams split tasks, and in what situations responsibilities over these tasks may change. Most of the power of using agreements is in reducing the mismatch between different agents' expectations about the state of the world, and how agents will interact; and the true state or operating model of the agents, which drive intention, and action. The mismatch between how agents represent other agents expected behavior might affect both the human and any automated participants. Often times this mismatch creates serious safety and performance failings, such as the grounding of the ship Royal Majesty. The inability of the crew to confirm that differential modes of automated navigation and GPS navigation information were active and valid [7] created serious issues. The combination of mode confusion, automation surprise (because of mismatched user-machine models of the environment and how each agent should work), and embedded automated commands operating silently, led a simple GPS receiver loss while leaving a harbor to metastasize into a catastrophic grounding of the ship many days later [7].

Working agreements may improve the transparency of automated decision-making systems by clearly indicating, as well as pre-determining what tasks each human and automated agent in a team performs, and under what conditions. Working agreements can even specify particular methods for solutions that could be used, or could not be used or allowed.

Working agreements seem especially appropriate for teaming, as team situations require communication and collaboration [8–10] in a way that is facilitated by (a) articulating tasks and functions, (b) properly allocating those functions among members, and (c) a display of the allocation of tasks among the team [11, 12]. For example, managing tasks is a source of difficulty for supervisory control of unmanned systems. Automation may reduce load by taking control of some of the operators' tasking (thus reducing demand). The tasks themselves form the backbone of an agreement, which can be configured and determine the behavior of the agents.

Still, certain tasks may be off-limits to certain actors (for example, weapons release). A working agreement could specify tasks that the automation cannot perform. However, a conditional property may modify such an agreement – which can be specified as well. The key is that these function allocations are brought to the surface of human-automation interactions, where they are visible.

1.2 Motivation

Working agreements help make the rules and constraints of human-machine teams explicit. It is especially important to articulate constraints when the roles and responsibilities of automation "adapt" to changing conditions of the operator and/or the environment. The notion of adapting responsibilities between humans and automation differs from traditionally "static" automation (e.g., [13]). Adaptive automation attempts to deal with the complexity and context-specific dynamics inherent in real-world system operations [13]. These situations are often large, complex, and rapidly overwhelm the limitations of human memory and attention [14]. Such adaptive automation seems to be the future of most systems due to greater success than seen in other automation implementations [15]. As working agreements help track and specify the changes in authority and responsibility that accumulate across adaptive systems, they are naturally beneficial.

Many have written about the intricacies of determining how and when automation changes its behaviors [16–18]. In considering how systems should adapt and interact with operators, working agreements could become a centerpiece in making explicit the capabilities and limitations of the automation (and aiding in developing operator expectancies by making them explicit). As a direct effect, human memory demand is reduced through a better, more available representation of what the automation is doing, and when. Most important is that the negotiation between systems and operators about task and function allocations can happen before workload reaches a high level [19], when most traditional solutions would take effect.

There are several key characteristics to a working agreement in human-machine systems; the use of adaptive or adaptable "mixed initiative" automation with multiple actors; a base model to represent the automation capabilities and limitations; and a triggering model in the case of adaptation for representing how and when to shift responsibilities between the human operator and the system. Adaptation in this sense represents an update of who is responsible for what, in real time, and adjusting the associated information displays, task sequences, and task deadlines as required [20].

Explicit definition of goals and tasks, and how they are shared and split between the agents, are necessary steps toward developing good teamwork and communication (e.g., [21, 22]). These same sets of goals and tasks can populate a working agreement structure [6, 23]. They are, in some respects, necessary before a working agreement can be developed, and are at least one way to abstract the tasks, and the methods to accomplish them, in a manner that a machine may understand and process [24].

In the case of working agreements, automation transparency should also be increased. The task-based nature of working agreements help indicate in clear language what the automation should be doing in any foreseeable condition. Tasks and methods to perform them can also be specified in system language, providing some shared understanding. Working agreements will not solve emergent issues that complex systems may create which are not foreseeable. Different methods may be useful for creating teamwork between humans and artificial neural network-created behaviors, for example [25].

Limitations of the automation can also be articulated in clear task-based language. The system can no longer hide which tasks it is, and is not, performing or going to perform. Allocations are displayed to the operator prior to performance, aiding in transparency and operator trust.

Working agreements may also be available during or displayed after performance to aid in training and building of user mental models. The agreements themselves may be useful when aggregated across a range of users; for example, did Bob set up a different agreement than Jill? What conditions prompted the difference? Were there any performance effects as a result? While there are a myriad of ways to improve operator trust or acceptance of automation behaviors, working agreements may be one of the easier methods to implement, once a task taxonomy is created and methods of performance are catalogued.

1.3 Applicability – When Should You Use This Design Pattern?

The working agreement pattern should be useful when there is at least one human and one computer "agent," each of which may be asked to perform tasks and occasionally share or trade responsibility for their execution. Such configurations are often referred to as adaptive automation [15] or sometimes adaptable automation [17], or mixed-initiative automation systems. In each, there are definable tasks that either the human or the automation can perform. Additionally, their name hinges on the locus of the allocation authority [26], either human "adaptable," or computer "adaptive," systems.

The allocation agent is that which decides how to allocate task responsibility between the participants. The allocation authority, if a computer, can assign incoming and existing tasks based on any number of inputs of varying complexity, including human workload estimation [18], [27] or state changes in the system given a set of parameters. For example, the flight management system – autopilot – in some airliners takes action at the point in landing where human reaction time is sub-par and unsafe. The function is allocated to the computer and taken out of the hands of the human, based on a condition; [28]. If the allocation authority is a human, then the human directs other humans and automation to have responsibility over certain tasks.

1.4 Participants

The human operator determines and sets the base configurations of the working agreements. These include more than just the function allocation elements. Configuration of the working agreement also involves determining the transition points and conditions that may serve to alter function allocations and authority changes to particular elements, classes of elements or more. The involvement of actual end users in this effort is critical to avoid confusing future operators, or anchoring design around inappropriate elements or triggers. Multiple cognitive task analysis methods are available to help determine these [29–31].

As we will discuss in a later section, human-human working agreements are already a familiar concept in some domains. The extension of the design pattern to human-agent and human-autonomy teams is a natural one, and is being explored.

An intelligent agent must also be able to interpret the human's determination and setting of default working agreement configurations, including function allocation, and transitioning conditions (usually based on if - then logic).

1.5 Structure

While we will usually think of working agreements as dictated by the human to the autonomy, they can also be considered a negotiation between the two in which the human has the upper hand and final say. The autonomy may reject roles through considerations of an inability to perform a particular function, and/or due to current configuration. One could also allow agents to bid for particular tasks. While most often this pattern may have a strictly one-way decision process, one can envision a two-way negotiated agreement fully deserving of that title.

Additionally, once a working agreement is set, its presence allows the human and autonomy to work as teammates rather than necessitating a strict supervisory control abstraction, if the agreement has been so specified.

Figure 1 uses a graphical language defined in [5] to illustrate the team structure being employed. In the figure, we see that the human and autonomy are teammates, both using a common collection of tools to perform work processes. While many implementations may look more supervisory in nature (which could be illustrated by a directional edge between human and autonomy) we diagrammed the more general case.

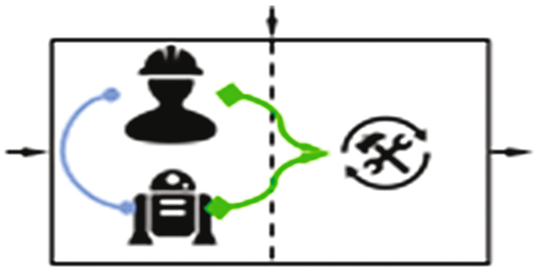

Fig. 1. Working agreements are at their most general a teaming relationship between human and autonomy. [From 4]

The working agreements themselves can be implemented with a variety of structures. However, the recommendation is to use a structure as shown in the following figure. In Fig. 2, we show that an agreement is between two or more agents. Agreements are associated with tasks to be performed. In particular, we are interested in situations where a human is entering into an agreement with an autonomous artificially intelligent agent.

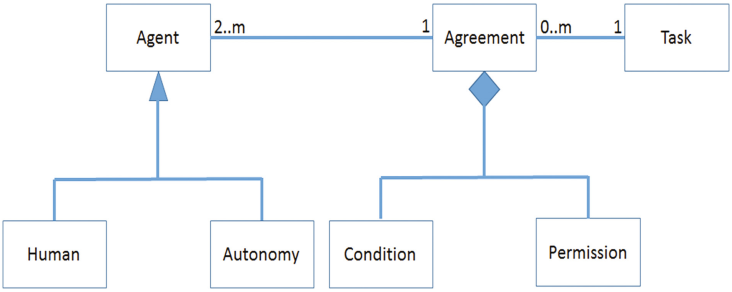

Fig. 2. Working agreement structure using unified modeling language.

We use a task, method breakdown shown in Fig. 3, for an unmanned systems command and control task allocation. Methods can be associated with agents and we can collect data about performance of agents for methods and tasks. Key to working agreements are the conditions under which each set of permissions is valid, and thus, whether the implementation is useful (as discussed next in Sect. 1.6).

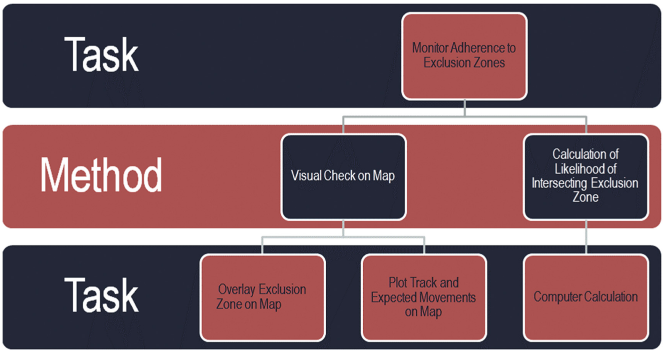

Fig. 3. Example task-method breakdown, as in [21].

1.6 Consequences – What Happens with Pattern Use?

Working agreements are theorized to provide several values to human-autonomy partnerships. A few theories include that agreements should (1) improve teamwork through transparency, and roles clarification; (2) provide a stable representation of the goals and limits of multiple partners in an environment, thereby lessening the burden on memory and training of the human user; and (3) inspire the refinement and definition of tasks and conditions. Better allocation will take advantage of differential skill sets between man and machine, and take into account the information and understanding needed by a human teammate.

Evaluating working agreements is, in essence, about the assessment of the whole system of systems incorporated in the environment. Naturally, performance is a key parameter. However, additional conditions should be examined, and we consider a few of them here (though this is not an exhaustive list).

First (1) whether there are emergent factors not captured by the task-methods hierarchy should be determined. Emergent factors could easily disrupt performance of the whole system or any of the parts; for example, user fatigue may disrupt whole system performance independent of any helpful effect of a working agreement. Second, (2) is to ensure each party's adherence to the agreement. If an agreement is violated, it might be important to understand the conditions surrounding the violation, but it also means the agreement was not *actually* in place. Third (3), a determination of agent comprehension of the agreement should be made. A user that does not understand the limitations imposed by an agreement will not act properly in the environment, so any performance consequences will not be the result of the agreement. Finally, (4) assessments must ensure there was correct and timely recognition of any triggering elements, events, conditions, and contexts that might comprise the agreement itself. The need for recognition applies to all agents in the agreement.

In order to evaluate the result of a working agreement when it is functioning properly, we turn to the measurement of human-automation function allocation [32]. As agreements define function allocation schemes, once a set of requirements are met, they can be evaluated by a grouping of measures advocated by Pritchett et al., of which we deemed the following as crucial to working agreements:

1. Assessments of user workload, and task loading. This measure may be facilitated by the specific task-based design and task-methods hierarchy breakdown. For example, certain task agreements may generate more or less demand on a human user, or create bandwidth overload on networked systems. There may be optimal agreements, then, that moderate workload and system performance.
2. A measure of authority as compared to a measure of responsibility. In other words, is there an imbalance between the controlling agent and the agent responsible? A working agreement, properly implemented, should directly improve this measure.
3. The stability of the performance and system environment, to the extent it is operationally relevant.
4. The measurement of interruptions, particularly interruption frequency and probability, as well as their effects on operator performance. For example, some function allocation schemes require repeated authorization or input from the human operator, sometimes to the detriment of their performance. Does a particular working agreement improve on, or make this situation worse?
5. The capabilities and limitations of automation and the behaviors that result when they are exceeded by the environment or the user. These may improve, to the extent they are captured in a working agreement, although this is unknown.
6. System cost and performance; is the working agreement improving either, as the result of its effective implementation?
7. How well does the human adapt to any context changes? Some working agreement settings may allow the user to exert more control or allocate more attention to tactics, for example, while others may mean the human is performing more menial labor.

We also suggest that measures of the human's trust in the autonomy may improve through increasing the transparency of function allocation. Of course, one of the gaps identified here is that working agreements and human-autonomy teams in general have not been adjudicated by these measures. Working agreement assessment is a move toward requiring measures of function allocation, and experimental exploration of design.

1.7 Implementation

Working agreement employment will require determination of what tasks are present to be split between human and machine. This is a fundamental function allocation task, and has puzzled engineers of complex systems before. However, here and elsewhere we believe we have a good approach in the application of a task-method hierarchy [24].

We are using this method to determine tasks and methods within a multiple unmanned system command and control environment. Here, supervisory control may be shared between humans and autonomy, creating unique teams [11]. It required the definition of tasks (which were in part derived from subject matter experts), as well as determining the possible conditions surrounding whether automation was capable or should be allowed to perform them.

Working agreements should always be developed in conjunction with user-centered design practices and testing. No amount of a priori assumptions by designers will ever compensate for hands-on, iterative testing with actual users of a system. It is unlikely that all possible effects from all possible combinations of agreements can be understood before users attempt to interact with them. As the autonomy should be viewed as a teammate, the display of the working agreement conditions and status is key in informing the user about the agreement. Efforts should be made to make sure this display is user-friendly and unambiguous through usability assessment and experimentation [33].

It is also critical to ensure that interacting with the agreement occur before operations begin, allowing operators to orient themselves to the situation and conditions. Orientation, as it facilitates communication and coordination, is key to teaming. One could argue that by centering on tasks, a working agreement helps define the work for the team; doing so before work begins may be the most effective approach.

Conditions in which the agreement language itself is unclear may arise. Clarity should be a design goal and should be facilitated by the task-method basis for the agreements and by user-centered design. For example, in air traffic control, there are specific assignments and air space coverage areas that are referenced in agreements, which delineate "who does what" in a verbiage that all parties understand. In the Consequences section above, we discussed some ways to measure the effectiveness of working agreement implementation; these methods may also capture something like language ambiguity problems, whether it is reflected in user trust, particular agreement configuration patterns, workload, and stability.

Implementation may be limited to where there exist definable, separable tasks, roles, and responsibilities. While this will cover most forms of human-automation teaming at a high level, it will not cover others. A good example of something difficult is the automation present in fly-by-wire systems used in high-performance aircraft which smooth human operator input and maintain flight stability; since the "parts" are not separable it would be very difficult to split tasks out. A similar problem exists when examining basic control of robotic systems that use automation to control robotic arms with high degrees of freedom. In essence, it may be difficult to implement working agreements in automated systems that do not allow for part-task separations.

As noted elsewhere, separating tasks may have important implications for training [34, 35]. Part-task training methods characterize how many automated systems are created, with the automation taking some tasks out of the users' hands. Naturally, the shift of "who does what" between operators and automation affects *what* users learn to perform [36]. Accordingly, the implementation of working agreements may help structure examinations of training with automation. Working agreements could be chosen and enforced experimentally, and measured along with training and skill acquisition of the users, to determine which agreements improve operations, and train the best skills for a given domain [36].

1.8 Known Uses and Related Patterns

The domain of air traffic control (ATC), which depends on proper coordination of aircraft flight paths between multiple human controllers for the duration of a flight, has historically used a working agreement model. The official term is letter of agreement, and they are employed by both the Federal Aviation Administration and by European Organization for the Safety of Air Navigation (EuroControl) [37]. Letters of agreement follow a standard format to supplement procedures that address the needs of particular sectors of airspace. In two ways, these agreements are adhering to our earlier principles for working agreements: (1) that language has to be clearly understood by the parties involved; and (2) that agreements foster the articulation of tasks to enable themselves.

While letters of agreement can cover a variety of scenarios, in general they serve to "(a) define responsibilities and coordination requirements, (b) establish or standardize operating methods, (c) specify special operating conditions or specific ATC procedures, or (d) delegate responsibility for an ATC service" [38]. These letters of agreement are negotiated by the parties involved, which for ATC could mean two airspace sectors, but also between entities [39]. In either case, the agreements define the purpose, scope, responsibilities, and procedure for both parties. Henceforth, parties are equipped with the necessary awareness to coordinate, delegate, and safely proceed with shared ATC procedures. Air traffic controllers are familiar with these agreements between sectors of airspace, and are trained to split tasks within the same sector. A good example of this behavior is within teams assigned to some En Route sectors, who split work between a primary Radar position (R-side) and a secondary Radar Associate operator (D-Side). The Associate supports the R-side operator by taking over housekeeping tasks, such as aircraft handoffs and assisting in separation assurance [40].

Effective teaming and negotiation of the agreement helps define a task allocation schema. In this schema, the D-side acts as a collaborator to help solve difficult conflicts. Much like a working agreement, each team must uniquely determine task allocation and conflict resolution strategies (such as a preference for issuing orders to aircraft to change vector, versus directing them to change altitude, in order to avoid collisions).

Close collaboration and transparency between an R-side and D-side is critical, as not all of these agreements are formal.

As the push toward automation of the national airspace gains momentum, the D-side operation is a natural candidate for supportive automation. With automation systems, the agreements would need to be made formal to enable teamwork. The NASA Ames Airspace Operation Laboratory has been investigating the allocation of tasks between a human air traffic controller and ground based separation assurance automation since 2008 [41]. These experiments specifically target increasing the total number of aircraft in a sector at any given time, while maintaining safety. Several experiments yielded confidence that separation assurance automation is a viable candidate for mediating workload in future, denser airspaces [41]. However, the relationship between the R-side air traffic controller (human) and automation for the D-side is still under investigation.

A 2010 experiment highlighted the human's willingness to delegate housekeeping tasks to the automation [41]. One notable finding was that controllers preferred authority over decision-making tasks, such as approving weather reroutes. However, the majority of controllers indicated post–study that they would still share tasks with the automation, instead of completing them in isolation. This collaboration paradigm mimics the current R-/D-side collaboration between human controllers. It also suggests that whether or not the task-element contains decision-making points, may influence whether an operator will trust that element in the agreement.

Transparency and goal alignment are critical in human-machine teaming. More focused analyses of the data from the 2010 study offered additional insights. Operators have specific preferences in conflict resolution (which the automation in the study did not take into account). Controllers often wanted a different type of conflict resolution than the automation could provide. The automation went about resolving conflicts in a very different manner [42], which violated controller expectancies.

Unlike a human D-Side who could be told a specific way to do a task, or at least could be negotiated with, controllers had to work around the limitations of the automation. Controllers were given no insight or interface for delegating tasks or the methods used to perform them. Implementing working agreements specifically would allow the users to articulate how they want to limit or expand the behavior of the automation, solving some of these problems, and building toward improved human-machine teams.

2 Conclusions

We have written extensively here regarding the use of working agreements between humans and automation. We believe that such agreements are a good way to organize the work of the various agents often in play in complex environments. It is a way to articulate who is doing what work, and the conditions that dictate change. By giving users the power (or denying them the power) to alter explicit expectancies about automation behaviors, better teaming should arise. It is important this power is available a priori, before the performance conditions are put upon the users.

As discussed above, working agreements themselves can be evaluated and assessed. Measurability is not a trivial property, as approaches that can be measured can also be definably improved. What works for some conditions may not work for others; what works for some populations of users may not for others.

The task specification exercise required to formulate agreements provides a valuable reason to articulate tasks that populate a given domain. It is not easy, but benefits the human factors assessment of the user needs and roles, and will help specify the different roles and capabilities of automated teammates.

Acknowledgements. The views expressed in this article are the sole views of the authors and do not reflect official policy or the opinions of the US Government or Department of Defense. This work was funded in part by a NISE ONR project from SPAWAR Pacific, and from the Autonomy Research Pilot Initiative IMPACT project.

References

1. Smith, R.: Panel on design methodology. In: OOPSLA 1987 (1987)
2. Beck, K., Cunningham, W.: Using pattern languages for object-oriented program. In: OOPSLA 1987 (1987)
3. Gamma, E., Helm, R., Johnson, R., Vlissides, J.: Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley, Boston (1995)
4. Lange, D.S., Gutzwiller, R.S.: Human-autonomy teaming patterns in the command and control of teams of autonomous systems. In: HCII 2016 (2016)
5. Schulte, A., Donath, D., Lange, D.S.: Design patterns for human-cognitive agent teaming. In: Harris, D. (ed.) Engineering Psychology and Cognitive Ergonomics, pp. 231–243. Springer, Cham (2016)
6. de Greef, T., Arciszewski, H., Neerincx, M.: Adaptive automation based on an object-oriented task model: implementation and evaluation in a realistic C2 environment. J. Cognit. Eng. Decis. Mak. **4**(2), 152–182 (2010)
7. Lützhöft, M.H., Dekker, S.W.A.: On your watch: automation on the bridge. J. Navig. **55**(1), 83–96 (2002)
8. Cuevas, H.M., Fiore, S.M., Caldwell, B.S., Strater, L.: Augmenting team cognition in human-automation teams performing in complex operational environments. Aviat. Space Environ. Med. **78**(5), B63–B70 (2007)
9. Malin, J.T., Schreckenghost, D.L., Woods, D.D., Potter, S.S., Johannesen, L., Holloway, M., Forbus, K.D.: Making intelligent systems team players: case studies and design issues. Volume 1: human-computer interaction design. NASA Technol. Memo. 104738, 1–276 (1991)
10. Klein, G., Bradshaw, J.M., Feltovich, J.M., Woods, D.D.: Common ground and coordination in joint activity. In: Rouse, W.B., Boff, K.R. (eds.) Organizational Simulation, pp. 139–184. Wiley, Hoboken (2005)
11. Gutzwiller, R.S., Lange, D.S., Reeder, J., Morris, R.L., Rodas, O.: Human-computer collaboration in adaptive supervisory control and function allocation of autonomous system teams. In: Shumaker, R., Lackey, S. (eds.) Virtual, Augmented and Mixed Reality, pp. 447–456. Springer, Cham (2015)

12. Gutzwiller, R.S., Lange, D.S.: Tasking teams: supervisory control and task management of autonomous unmanned systems. In: International Conference on Virtual, Augmented and Mixed Reality, pp. 397–405 (2016)
13. Sheridan, T.B., Verplank, W.L.: Human and computer control of undersea teleoperators, Man-Machine Syst. Lab, Dep. Mech. Eng. MIT. Grant N00014-77-C-0256 (1978)
14. Cummings, M., Mastracchio, C., Thornburg, K., Mkrtchyan, A.: Boredom and distraction in multiple unmanned vehicle supervisory control. Interact. Comput. **25**(1), 34–47 (2013)
15. Kaber, D.B.: Adaptive automation. In: Oxford Handbook of Cognitive Engineering. In: Lee, J., Kirlik, A. (eds.) Oxford University Press, Oxford, March 2013
16. Parasuraman, R., Bahri, T., Deaton, J.E., Morrison, J.G., Barnes, M.: Theory and design of adaptive automation in aviation systems. NAWCADWAR Technical Report-92033-60, pp. 1–44 (1992)
17. Scerbo, M.W.: Theoretical perspectives on adaptive automation. In: Parasuraman, R., Mouloua, M. (eds.) Automation and Human Performance: Theory and Applications, pp. 37–63. Erlbaum, Mahwah (1996)
18. Kaber, D.B., Endsley, M.R.: The effects of level of automation and adaptive automation on human performance, situation awareness and workload in a dynamic control task. Theor. Issues Ergon. Sci. **5**(2), 113–153 (2004)
19. Arciszewski, H., de Greef, T., van Delft, J.: Adaptive automation in a naval combat management system. IEEE Trans. Syst. Man Cybern. Part A Syst. Hum. **39**(6), 1188–1199 (2009)
20. Feigh, K., Dorneich, M.C., Hayes, C.C.: Toward a characterization of adaptive systems: a framework for researchers and system designers. Hum. Factors **54**(6), 1008–1024 (2012)
21. Klein, G., Woods, D.D., Bradshaw, J.M., Hoffman, R.R., Feltovich, P.J.: Ten challenges for making automation a team player in joint human-agent activity. IEEE Intell. Syst. **19**, 91–95 (2004)
22. Hollnagel, E., Woods, D.: Joint Cognitive Systems: Foundations of Cognitive Systems Engineering. Taylor & Francis, Boca Raton (2005)
23. Johnson, A.W., Oman, C.M., Sheridan, T.B., Duda, K.R.: Dynamic task allocation in operational systems: issues, gaps, and recommendations In: IEEE Aerospace Conference, pp. 1–15, March 2014
24. Lange, D.S., Gutzwiller, R.S., Verbancsics, P., Sin, T.: Task models for human-computer collaboration in supervisory control of teams of autonomous systems. In: International Inter-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support, pp. 97–102 (2014)
25. Gutzwiller, R.S., Reeder, J.: Human interactive machine learning for trust in teams of autonomous robots. In: IEEE CogSIMA (2017)
26. Sheridan, T.: Adaptive automation, level of automation, allocation authority, supervisory control, and adaptive control: distinctions and modes of adaptation. IEEE Trans. Syst. Man Cybern. Part A Syst. Hum. **41**(4), 662–667 (2011)
27. Wilson, G., Russell, C.: Performance enhancement in an uninhabited air vehicle task using psychophysiologically determined adaptive aiding. Hum. Factors **49**(6), 1005–1018 (2007)
28. Parasuraman, R., Riley, V.: Humans and automation: use, misuse, disuse, abuse. Hum. Factors **39**(2), 230–253 (1997)
29. Crandall, B.W., Klein, G.A., Hoffman, R.: Working Minds: A Practitioner's Guide to Cognitive Task Analysis. MIT Press, Cambridge (2006)
30. Clark, R., Feldon, D., van Merriënboer, J.J.G., Yates, K., Early, S.: Cognitive Task Analysis, vol. 2006 (2006)
31. Endsley, M.R., Jones, D.: Designing for Situation Awareness: An Approach to Human-Centered Design, 2nd edn. CRC Press, New York (2012)

32. Pritchett, A.R., Kim, S.Y., Feigh, K.M.: Measuring human-automation function allocation. J. Cogn. Eng. Decis. Mak. **8**(1), 52–77 (2013)
33. Kortum, P.: Usability Assessment: How to Measure the Usability of Products, Services, and Systems. Human Factors and Ergonomics Society, Santa Monica (2016)
34. Wightman, D.C., Lintern, G.: Part-task training for tracking and manual control. Hum. Factors **27**(3), 267–283 (1985)
35. Naylor, J.C., Briggs, G.E.: Effects of task complexity and task organization on the relative efficiency of part and whole training techniques. J. Exp. Psychol. **65**(3), 217–224 (1963)
36. Gutzwiller, R.S., Clegg, B.A., Blitch, J.G.: Part-task training in the context of automation: current and future directions. Am. J. Psychol. **126**(4), 417–432 (2013)
37. European Organisation for the Safety of Air Navigation: Common Format Letter of Agreement Between Air Traffic Services Unites (ASM.ET1.ST015 DEL01/02 Ed. 4) (2012)
38. US Department of Transportation Federal Aviation Administration: Air Traffic Organization Policy: Section 3. Letters of Agreement (LOA) (2010)
39. Federal Aviation Administration Los Angeles ARTCC: Los Angeles Air Route Traffic Control Center, CA, U.S.A; Tijuana Terminal Radar Approach Control: Letter Of Agreement (1990)
40. US Department of Transportation Federal Aviation Administration: Air Traffic Organization Policy (2015)
41. Prevot, T., Homola, J., Martin, L., Mercer, J., Cabrall, C.: Toward automated air traffic control—investigating a fundamental paradigm shift in human/systems interaction. Int. J. Hum. Comput. Interact. **28**(2), 77–98 (2012)
42. Mercer, J., Gomez, A., Homola, J., Prevot, T.: A closer look at automation behavior during a human in the loop simulation. In: 33rd IEEE/AIAA Digital Avionics Systems Conference (2014)

http://www.springer.com/978-3-319-60590-6

Advances in Human Factors in Simulation and Modeling
Proceedings of the AHFE 2017 International
Conference on Human Factors in Simulation and
Modeling, July 17–21, 2017, The Westin Bonaventure
Hotel, Los Angeles, California, USA
Cassenti, D.N. (Ed.)
2018, XVI, 597 p. 242 illus., Softcover
ISBN: 978-3-319-60590-6