Human-Computer Collaboration in Adaptive Supervisory Control and Function Allocation of Autonomous System Teams

Robert S. Gutzwiller, Douglas S. Lange(✉), John Reeder,
Rob L. Morris, and Olinda Rodas

Space and Naval Warfare Systems Center Pacific, San Diego, CA, USA
{robert.s.gutzwillerl,doug.lange,john.reeder,
rob.morris,maria.rodas}@navy.mil

Abstract. The foundation for a collaborative, man-machine system for adaptive performance of tasks in a multiple, heterogeneous unmanned system teaming environment is discussed. An autonomics system is proposed to monitor missions and overall system attributes, including those of the operator, autonomy, states of the world, and the mission. These variables are compared within a model of the global system, and strategies that re-allocate tasks can be executed based on a mission-health perspective (such as relieving an overloaded user by taking over incoming tasks). Operators still have control over the allocation via a task manager, which also provides a function allocation interface, and accomplishes an initial attempt at transparency. We plan to learn about configurations of function allocation from human-in-the-loop experiments, using machine learning and operator feedback. Integrating autonomics, machine learning, and operator feedback is expected to improve collaboration, transparency, and human-machine performance.

Keywords: Autonomics · Autonomous systems · Supervisory control · Task models

1 Introduction

Human roles in unmanned vehicle command and control (C2) are in flux, transitioning between responsibility for one situated vehicle, to a supervisor responsible for many abstracted, automated vehicles [1, 2]. This supervisory role is very similar to military command and control in terms of providing commanders intent to agents, planning and monitoring their actions, intervening if necessary, and learning from feedback about how the mission is being performed [3, 4]. The rapid advance of technology and recent autonomous ship trials prove such role transitions are imminent for the warfighter [5].

Understanding and improving human-machine teaming is needed to aid future supervisors [6]. In the current paper, we provide the first steps for our adaptive function allocation methods. These help guide our ultimate goal of autonomics implementation for human-machine cooperation with teams of autonomous heterogeneous vehicles.

Many systems benefit from human-centered integration [7–9], although automating tasks occasionally has notable downsides, such as lowered situation awareness,

R. Shumaker and S. Lackey (Eds.): VAMR 2015, LNCS 9179, pp. 447–456, 2015.
DOI: 10.1007/978-3-319-21067-4_46

increased workload and increased operator complacency [10–13]. Costs of automation are sometimes a function of human attention allocation. A complacent operator allocates attention differently than an attentive one who is constantly checking up on automated systems or teams [13].

As a human supervisor, tasks require four general information processing phases; acquiring information, analyzing it, coming to a decision and executing it. Automation of a task within each phase may vary across a single automated system [14], but there usually are tradeoffs in performance and attention as a result. These tradeoffs are more favorable for higher degrees of automation of information acquisition and analysis, compared to the automation of decisions - but there are always tradeoffs ([15, 16]). In part, adaptive systems are an answer to this problem. However, an open question is about how to divide functions and tasks between automation and humans. In the unmanned system domain, especially, this is a pressing issue because automation is necessary but the demand to maintain awareness for the task, and transparency in the available automation, are both high.

2 Methods

We have chosen a "playbook" approach to the supervision of multiple unmanned vehicles [17–19] as one potential solution in a wide spectrum of control abstractions, in part because decision making is left to the operator (choosing plays), and because of the documented benefits of these systems in the literature [19–21]. These systems can be made to adapt to the environment, operator and the tasks at hand [20]. Adaptation in this sense represents updating "who does what" dynamically, in addition to determining task sequences and deadlines, and adapting information presentation [22]. These adaptive approaches typically outperform static levels-of-automation approaches [8, 21].

Function allocation schemes should reflect the contextual dynamics of real-world supervision. The lack of dynamics is exactly what is poorly addressed by static LOA perspectives [23, 24]. Therefore we focus on methods that allow for adaptation of task performance with automation, understanding that dynamics encompass a wide range of possibilities. Determining which are useful is something that machine learning may aid, but this is a long term goal not addressed in detail here. In order to implement an adaptive system in the context of multiple, heterogeneous vehicle command and control, we are also employing the use of autonomics to help achieve system goals.

Autonomic approaches manage complex systems such that they exhibit self-adaptation in response to demands on the system or degradation of performance. One such autonomics approach is the Rainbow autonomics framework (Fig. 1), developed at Carnegie Mellon University (CMU) [25]. Rainbow employs an architecture-based, self-adaptation approach that models the managed system through an architecture description language, receives information from the system through gauges from probes that read data from different points within the managed system, and then executes strategies. These strategies provide instruction on how the managed system should adapt to sensed changes in order to maintain "health" (which can be characterized with different metrics).

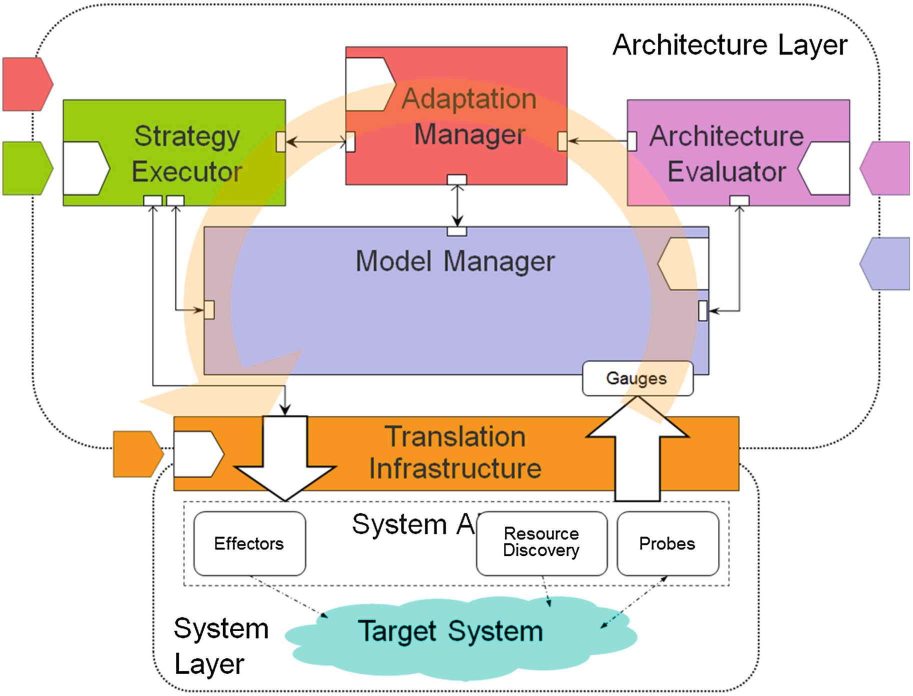

Fig. 1. The rainbow autonomics framework. [From 25]

In a system consisting of teams of unmanned vehicles under the supervisory control of a single operator, necessarily there must exist some level of autonomy for the vehicles to operate without the operator being responsible for remote operation [26]. As the number of vehicles, and teams of vehicles begins to proliferate, there also must be enough automation to assist the supervisor in managing a large complex situation, which rapidly overwhelms the limitations of memory, attention, and basic human performance (e.g., [27]). Adaptation becomes a question of whether autonomy should be in charge (and allocating tasks), and the relative level at which the autonomy should be operating within a range - from completely manual to largely automatic performance [20, 28, 29]. Movement along this range has been used to reduce workload for operators in demanding tasks.

The autonomics framework can play a central role in controlling such a “sliding scale” of autonomy, allowing for adaptive automation approaches in supervisory control. Because the primary role of Rainbow is to observe and maintain holistic system and mission health, autonomics can adjust the level of autonomy based upon multiple conditions within the system. These conditions may include operator workload [30] and attention allocation [31, 32], attention required for a task, risk associated with a task or with the autonomy, risk associated with failing to act, operator input on autonomy allowance [33], and other such determinable parameters.

For example, when autonomics detects an overloaded operator, it could set the autonomy level higher, potentially relieving the burden on the operator and allowing the system to maintain mission alignment. Such a relationship is not our starting point, though it has been done successfully elsewhere [30, 34]. Rather, our particular method allows for the inclusion of multiple, measurable factors, and for these factors to be weighted differently in whether the autonomy takes action through a particular strategy. For example, while adaptive function allocation may aid performance of certain types of tasks, such as information acquisition or analysis, it may harm others such as decision making [35]. Further, unique combinations of these measures may provide more effective transitions, both up and down on the sliding scale of automation by incorporating multiple factors, rather than a single measure (e.g., [28, 29], [36, 37]).

Two major challenges present themselves in this pursuit. First, as we consider the tasks present in the existing multi-vehicle control domain, we must be able to define them, identify them, and track their performance by both the human and the automation. And second, we must eventually demonstrate that we can effectively learn how to measure, utilize and weight the impacts of various conditions and factors within the human-machine system to adapt the allocation sets.

We address here the task-based aspect of our work, which encompasses the method used to breakdown the operating task structure into definable portions for later allocations. We will present these tasks within a display, called a task manager, used successfully in other ongoing research [38]. The task manager also provides us with an opportunity to address the transparency issues involved in automated function allocation, discussed in the sections below.

3 Task Model

3.1 Task Methods Hierarchy

Providing assistance to supervisors in managing and performing tasks requires assisting automation to have some model of the tasks involved. Task models have been well studied and reviewed. Our approach traces its roots to the task structures defined by Chandrasekaran et al. [39]. Chandrasekaran's task structure is a bipartite directed acyclic graph (DAG) of tasks, methods and subtasks. As in most DAG structures, these components may be applied recursively. The bipartite and recursive nature of this task structure results in tasks decomposed into methods, which in turn are composed of subtasks, which are simply tasks themselves.

Tasks represent elements that must be performed in order to execute the method. Methods are alternative approaches to completing a task. Therefore, the DAG has characteristics of decision trees as well (Fig. 2).

Task generation can occur either through user initiation, or agents that recognize particular properties within the system. When that happens, a task, defined in the structure above, gets created and then must be queued for a supervisor, or an autonomous agent assisting a supervisor. The systems first decisions are who will perform the task and by what method. Some methods may only be suitable for human efforts, and some for computer. Some tasks require collaborative efforts, where some of the sub-tasks are performed by human supervisors and others by automation.

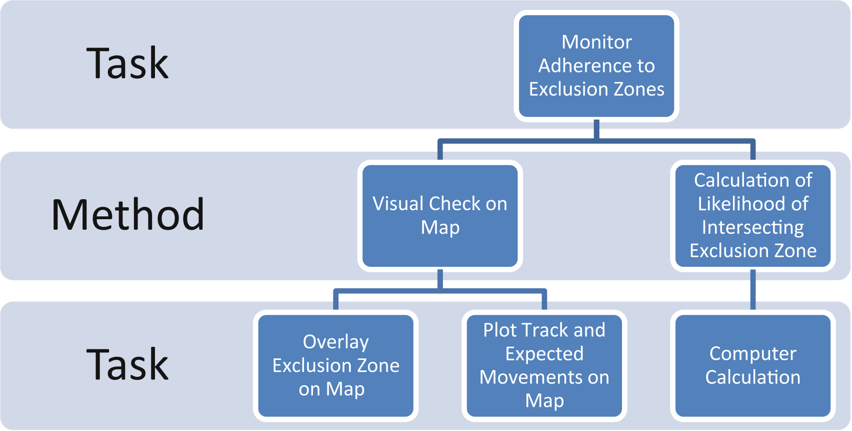
Task
Monitor Adherence to Exclusion Zones
Method
Visual Check on Map
Calculation of Likelihood of Intersecting Exclusion Zone
Task
Overlay Exclusion Zone on Map
Plot Track and Expected Movements on Map
Computer Calculation

Fig. 2. Example structure: the bipartite nature creates alternating levels of tasks and methods

Algorithms can be used to make initial decisions or suggestions concerning tasking. With proper instrumentation this decision can be made by considering properties of the environment; the attention available from the supervisor with the history of the supervisor's performance for this type of task, other ongoing tasks, their current workload, authorizations; the user-automation working agreements which help define and constrain who can do what (and when they can do it); and the mission's goals.

These granular tasks and their associated methods represent a breakdown of the multiple vehicle task space. They will be unique to a specific platform or unmanned C2 system; yet the principles on which our measurements, and the autonomics system are based may have wider applicability. For example, Rainbow can sit "on top" of other systems, an does not interfere with their operations – therefore it can be integrated somewhat easily into existing systems, and the majority of the work is in developing the appropriate probes and gauges for the system to sample information through.

3.2 Task Manager Display

To provide visibility to the user and ultimate control over tasking, our initial task management system was developed. It will allow the user, as well as the autonomic controls, to reprioritize tasking for the human and autonomous assistant. The display itself provides information on the current tasking for both entities (Fig. 3).

While allowing the user to dictate allocation necessarily adds to the load for the operator, it allows the operator awareness of the tasks "to be done" in the scenario as well. Even if no tasks were to be performed by an automated agent, the task management interface provides an aid to the operator concerning tasks to be completed (reducing the demands on prospective memory; [40, 41]). The use of such an interface is a starting point for incorporating basic intent inference, as was done in prior programs, allowing us to facilitate tasks when they are selected by the operator to be completed, by pulling up the relevant task information for a given method. The instrumentation of the task

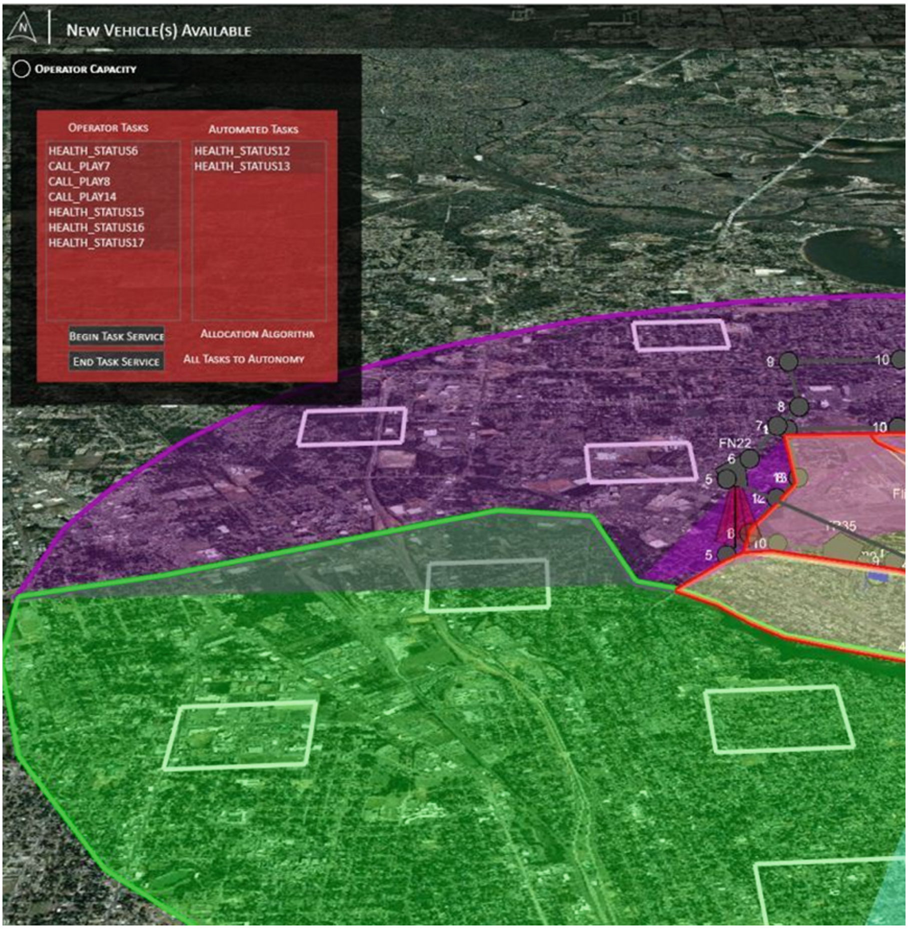

Fig. 3. Initial Prototype of Task Manager Displaying Simple User Interface with Task Queues

manager will also provide us, as researchers, the important data concerning task shedding and other task management behaviors including switching, and the effects of interruptions – both known issues in this domain.

Finally, the task manager also provides some insight into the operations of the autonomy. For example, if the autonomy senses operator overload and begins to task authorized tasks to the automation but the user is not actually overloaded, this will be self-evident by examining the automation's task queue. As we explore methods of prioritization within the queues themselves, potentially giving the automation authority to appropriately weight the importance of certain tasks for the human operator to aid in decision making, the interface will be expanded to incorporate and present information on how priority decisions were made. It may even be possible to allow the working agreement between the human and the automation to include an agreed-upon priority ranking to develop expectations between the two entities.

4 Risk-Attention Metric

Ultimately, if we are to choose between supervisor(s) and autonomous assistant(s) for performing tasks, we need to weigh the capabilities of each against current load, and utilize known human factors in the decision. The human factors utilized are meant to ensure that we maximize the collaborative decision making and performance ability of the human-computer team. One way to do that is to balance risk vs. attention. The fundamental idea is that if the most capable agent to do a particular action is available, i.e., the one that reduces risk in performance of the action, it would be the natural agent to select if all agents (human as well as autonomous) were completely available.

However, when available attention is not uniform or abundant, we need to balance taking on risk to manage attention. This is something humans naturally do, though they may not be optimal. This is further complicated by the risks posed by human performance factors, such as the need for situational awareness and understanding what the automation is doing.

If the workload is fairly light, we want the operator to pay attention to more tasks, despite the computer being able to help. This will help keep operator situational awareness up and prevent operators from becoming overly complacent concerning the remaining automation. If the workload is heavy enough to put operations at risk because of limited operator attention availability, then we want the computer (or other people) to perform more of the tasks. Therefore, we want to utilize scores concerning the ability of the computer to perform a task, relative to the ability of the human supervisors to perform the same task; however our approach differs in that it is also weighing contextual and adaptive factors, and that some operations will be limited by our incorporation of working agreements. The difference between which agents do what is essentially a measure of the risk in turning the task over to that agent. There is additional interest in using this metric to evaluate the teaming of humans and automation to perform certain tasks (e.g., a truly collaborative “method” for performance), which may rank lower on risk than either agent performing alone.

Measuring attention availability is more difficult. At its most straightforward, we can measure how far behind schedule the task queue has become. Every task will have some deadline (hard or soft), and tasks will have a distribution of predicted durations depending on who has been assigned the task. It becomes an easy matter to count the number of times that deadlines are missed over a particular time window as one fundamental attribute of performance: it is by no means the only one. More advanced methods that measure other actions by the operators, biometrics [32], and other characteristics of the tasks may be beneficial and could be incorporated when available.

As this is work in progress, we have not yet experimented to find the appropriate range of risk/attention values for maintaining situational awareness of both the automation, and the tasks themselves. This is the essence of the experimentation that we will begin with the prototype task manager in late 2015. As a final note, we have alluded to the use of machine learning to determine what an optimal configuration of function allocations may be, based on recent developments in measuring their efficacy [24]. These measures require user performance as well, but represent a further stage of development for this work, and one that holds great promise for helping to determine appropriate allocation schemas, and how to adapt them (yet has not been done before). It explores a range of potential solutions much more expansive than a typical analysis may uncover or engineering might suggest.

5 Summary

As C2 develops technologically, reliable methods for human interaction with an adaptive range of automation and autonomy must be established. Ultimately the attention allocation policy of a user plays a significant role in determining tradeoffs. As operators freely choose where to allocate attention, reduced workload provided by automated assistance leaves both negative and positive behaviors as possible outcomes.

We are of course wary of schemes which employ high degrees of automation, but, we also consider here that tradeoffs are a natural and persistent byproduct of function allocation [15]. By allowing an autonomics framework to aid in task management and providing an initial display of this aiding, we hope to provide adaptations and timely support for the system and the user that accurately reflect the state of both, in context of the mission. This method is sufficiently different from other methods of function allocation as to suggest it could be beneficial. It may allow for adaptation that is not opaque, and thus of great benefit.

Finally, we have touched on a relatively new concept of working agreements between what the autonomy can and can't do as set by the human [33]. This addresses the complexities of expectation-based faults, wherein the human expects the automation to perform (or does not). It also increases the collaboration of the human-system team. Overall, then, we are making progress toward more cooperation and more transparency in these types of systems.

Acknowledgements. This work was supported by the Space and Naval Warfare Systems Center Pacific Naval Innovative Science and Engineering Program. This work was also supported by the US Department of Defense Autonomy Research Pilot Initiative under the project entitled "Realizing Autonomy via Intelligent Adaptive Hybrid Control".

References

1. Sheridan, T.B.: Humans and automation: Systems design and research issues. Human Factors and Ergnomics Society Wiley, Santa Monica/New York (2002)
2. Finn, A., Scheding, S.: Developments and Challenges for Autonomous Unmanned Systems, p. 237. Springer-Verlag, Berlin Heidelberg (2010)
3. Willard, R.: Rediscover the art of command and control. Proc. - United States Nav. Inst. **128** (10), 52–54 (2002)
4. Sheridan, T.B., Parasuraman, R.: Human-automation interaction. Rev. Hum. Factors Ergonomics **1**(41), 89–129 (2005)
5. Tucker, P.: Inside the Navy's secret swarm robot experiment. Defense One, 05 October 2014
6. Chen, J.Y.C., Barnes, M.J.: Human–agent teaming for multirobot control: a review of human factors issues. IEEE Trans. Hum.-Mach. Syst. **44**(1), 13–29 (2014)

7. Endsley, M.R., Kaber, D.B.: Level of automation effects on performance, situation awareness and workload in a dynamic control task. Ergonomics **42**(3), 462–492 (1999)
8. 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)
9. Cummings, M.L., How, J., Whitten, A., Toupet, O.: The impact of human-automation collaboration in decentralized multiple unmanned vehicle control. In: Proceedings IEEE (2011)
10. Bainbridge, L.: Ironies of automation. Automatica **19**(6), 775–779 (1983)
11. Endsley, M.R., Kiris, E.O.: The out-of-the-loop performance problem and level of control in automation. Hum. Factors **37**(2), 381–394 (1995)
12. Kirlik, A.: Modeling strategic behavior in human-automation interaction: why an 'aid' can (and should) go unused. Hum. Factors **35**(2), 221–242 (1993)
13. Parasuraman, R., Manzey, D.H.: Complacency and bias in human use of automation: An attentional integration. Hum. Factors **52**(3), 381–410 (2010)
14. Parasuraman, R., Sheridan, T.B., Wickens, C.D.: A model for types and levels of human interaction with automation. IEEE Trans. Syst. Man Cybern. Part A Syst. Hum. **30**(3), 286–297 (2000)
15. Onnasch, L., Wickens, C.D., Li, H., Manzey, D.: Human performance consequences of stages and levels of automation: an integrated meta-analysis. Hum. Factors **56**(3), 476–488 (2014)
16. Rovira, E., McGarry, K., Parasuraman, R.: Effects of imperfect automation on decision making in a simulated command and control task. Hum. Factors **49**(1), 76–87 (2007)
17. Miller, C.A., Parasuraman, R.: Designing for flexible interaction between humans and automation: delegation interfaces for supervisory control. Hum. Factors **49**(1), 57–75 (2007)
18. Parasuraman, R., Galster, S., Squire, P., Furukawa, H., Miller, C.: A Flexible Delegation-Type Interface Enhances System Performance in Human Supervision of Multiple Robots: Empirical Studies With RoboFlag. IEEE Trans. Syst. Man Cybern. Part A Syst. Hum. **35**(4), 481–493 (2005)
19. Squire, P.N., Parasuraman, R.: Effects of automation and task load on task switching during human supervision of multiple semi-autonomous robots in a dynamic environment. Ergonomics **53**(8), 951–961 (2010)
20. Scallen, S.F., Hancock, P.A.: Implementing adaptive function allocation. Int. J. Aviat. Psychol. **11**(2), 197–221 (2001)
21. Parasuraman, R., Mouloua, M., Molloy, R.: Effects of adaptive task allocation on monitoring automation. Hum. Factors **38**(4), 665–679 (1996)
22. 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)
23. Kaber, D.B.: Adaptive Automation, In: Oxford Handbook of Cognitive Engineering, J. Lee and A. Kirlik, Eds. Oxford University Press, Oxford (2013)
24. Pritchett, A.R., Kim, S.Y., Feigh, K.M.: Measuring human-automation function allocation. J. Cogn. Eng. Decis. Mak. **8**(1), 52–77 (2013)
25. Garlan, D., Cheng, S., Huang, A., Schmerl, B., Steenkiste, P.: Rainbow: architecture-based self-adaptation with reusable infrastructure. IEEE Comput. **37**(10), 46–54 (2004)
26. Cummings, M.: Operator interaction with centralized versus decentralized UAV architectures. HAL Lab MIT Dep. Aeronaut. Astronaut., pp. 1–13, 2013
27. Cummings, M., Mastracchio, C., Thornburg, K., Mkrtchyan, A.: Boredom and Distraction in Multiple Unmanned Vehicle Superisory Control. Interact. Comput. **25**(1), 34–47 (2013)

28. 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, NJ (1996)
29. Johnson, A.W., Oman, C.M., Sheridan, T.B., Duda, K.R.: Dynamic task allocation in operational systems: Issues, gaps, and recommendations. In: 2014 IEEE Aerospace Conference, pp. 1–15 (2014)
30. Wilson, G., Russell, C.: Performance enhancement in an uninhabited air vehicle task using psychophysiologically determined adaptive aiding. Hum. Factors **49**(6), 1005–1018 (2007)
31. Ratwani, R., McCurry, J.M., Trafton, J.G.: Single operator, multiple robots: an eye movement based theoretic model of operator situation awareness. In: Present. 5th ACM/IEEE International Conference Human-Robot Interact, pp. 235–242 (2010)
32. Breslow, L., Gartenberg, D., McCurry, J.M., Trafton, J.: Dynamic operator overload: a model for predicting workload during supervisory control. IEEE Trans. Hum.-Mach. Syst. **44**(1), 30–40 (2014)
33. 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. Cogn. Eng. Decis. Mak **4**(2), 152–182 (2010)
34. Bailey, N.R., Scerbo, M.W., Freeman, F.G., Mikulka, P.J., Scott, L.A.: Comparison of a brain-based adaptive system and a manual adaptable system for invoking automation. Hum. Factors **48**(4), 693–709 (2006)
35. Kaber, D., Wright, M., Prinzel, L., Clamann, M.: Adaptive automation of human-machine system information-processing functions. Hum. Factors **47**, 730–741 (2005)
36. Rouse, W.: Adaptive aiding for human/computer control. Hum. Factors **30**(4), 431–443 (1988)
37. Inagaki, T.: Adaptive automation: sharing and trading of control, In: Handbook of Cognitive Task Design, Hollnagel, E. (ed.), LEA, pp. 147–169 (2003)
38. Rodas, M., DiVita, J., Morris, R.: Introducing a task prioritization tool in a complex supervisory control environment. Cogn. Methods Situat. Aware. Decis. Support, pp. 79–82, 2014
39. Chandrasekaran, B., Johnson, T.R., Smith, J.W.: Task-structure analysis for knowledge modeling. Commun. ACM **33**(9), 124–136 (1992)
40. Dodhia, R., Dismukes, R.: Interruptions create prospective memory tasks. Appl. Cogn. Psychol. **89**, 73–89 (2009)
41. Dismukes, R., Nowinski, J.: Prospective memory, concurrent task management, and pilot error. In: Kramer, A., Wiegmann, D., Kirlik, A. (eds.) Attention: From theory to practice, pp. 225–236. Oxford University Press, New York (2007)