Adaptive Automation in a Naval Combat Management System

Henryk F. R. Arciszewski, Tjerk E. de Greef, and Jan H. van Delft

Abstract—**There is a continuing trend of letting fewer people deal with larger amounts of information in more complex situations using highly automated systems. In such circumstances, there is a risk that people are overwhelmed by information during intense periods or, on the other hand, do not build sufficient situational awareness during periods of slack to deal with situations where human intervention becomes necessary. A number of studies show encouraging results in increasing the efficiency of human–machine systems by making the automation adapt itself to the human needs. Current literature shows no examples of adaptive automation in real operational settings, however. We introduce a fine-grained adaptation methodology based on well-established concepts that is easy to comprehend and likely to be accepted by the end user. At the same time, we let the machine operate like a virtual team member in that it continuously builds its own view of the situation independent from the human. Working agreements between human and machine provide lower and upper bounds of automation that are in advance determined by the end user so that unwanted appropriation of responsibility by the machine is avoided. The framework is domain neutral and therefore thought to be applicable across a wide range of complex systems, both military and civilian. It gives researchers an architecture that they can use in their own work to get adaptive automation up and running quickly and easily.**

Index Terms—**Adaptive automation, automation levels, human factors, human–machine system, object-oriented task allocation, working agreements.**

I. Introduction

In many domains (air traffic control, military command and control, and crisis management), humans are assisted by computer systems during their assessment of the situation and their subsequent decision making. A continuous technology push has led to innovative but, at the same time, complex systems. Technological development has enabled humans to work more efficiently or effectively (or both) using such systems. In such information-rich and dynamic environments, however, a competition for the users' attention is going on between numerous different information items, at times leading to a cognitive overload. This overload originates in the limitations of human attention and constitutes a well-known bottleneck in human information processing. Research has indicated repeatedly, moreover, that aiding the crew by as much automation as technologically feasible does not necessarily lead to a better performance. Prolonged periods of low activity (i.e., underload), on the other hand, lead to performance degradations because the operator gets out of the information processing loop as he or she becomes a passive monitor. Taking both underload and overload into account, it is important the keep the human within a bandwidth of workload for optimum performance. Parasuraman *et al.* [1] and Endsley *et al.* [2] offer a good overview of potential pitfalls. In order to reach a truly optimal human–machine combination, continuous research is required to attain the right balance between technologically feasible levels of automation (LoAs) on the one hand and human requirements and responsibilities on the other hand.

Various studies have been conducted that provide indications on the *level of control* that can be allocated toward a human or a system (see [3] for a detailed overview). Since 1951, various suggestions have been proposed starting with Fitts's list [4], continuing with the well-known taxonomies of Sheridan and Verplank [5] and Endsley [6] and finishing with the widely accepted model of Parasuraman *et al.* [1]. They describe a model where information processing is divided into four stages (information acquisition, information analysis, decision and action selection, and action implementation) and where each of these stages is automated at a different level of autonomy. In the military world, this four-stage information processing loop is usually referred to as the OODA (Observe, Orient, Decide, Act) loop first introduced by Boyd [7, ch. 24]. Parasuraman *et al.* [1] propose to choose a type and level of automation based on primary (e.g., human performance consequences) and secondary (e.g., automation reliability and costs of action) criteria. They argue for the application of higher LoAs when applied to the sensory and action levels (information acquisition and action implementation) compared to those of the cognitive levels (information analysis and decision and action selection) [1], [8], [9]. High LoAs in the information analysis phase can severely impact the situational awareness of the human and make it difficult for him or her to monitor proper system behavior and to correct system errors when these occur [10]. Likewise, high LoAs in the decision-making phase make it difficult to ensure that proper decisions are indeed being made.

Manuscript received March 6, 2008. First published September 1, 2009; current version published October 16, 2009. This work was supported by the Netherlands Ministry of Defense Research and Development organization under Programs V055 and V206. This paper was recommended by Associate Editor M. Dorneich.

H. F. R. Arciszewski is with the Netherlands Organization for Applied Scientific Research (TNO), Defence, Security and Safety, Information and Operations, The Hague, 2597 AK, The Netherlands (e-mail: henryk.arciszewski@tno.nl).

T. E. de Greef was with the Netherlands Organization for Applied Scientific Research (TNO), Defence, Security and Safety, Human Factors, 3769 ZG Soesterberg, The Netherlands. He is now with the Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, 2600 GA Delft, The Netherlands (e-mail: t.e.degreef@tudelft.nl).

J. H. van Delft was with the Netherlands Organization for Applied Scientific Research (TNO), Defence, Security and Safety, Human Factors, 3769 ZG Soesterberg, The Netherlands. He is now with the ForceVision Department, Royal Netherlands Navy, Den Helder, 1780 CA, The Netherlands (e-mail: jan.van.delft@forcevision.nl).

Digital Object Identifier 10.1109/TSMCA.2009.2026428

Even if the system designer gets the amount of automation right, however, (highly) varying circumstances will still produce a (highly) varying workload. A so-called adaptive system [11], in which the division of labor between human and machine is flexible and responsive to task or human demands, is thought to represent a better solution to the problem of function allocation than the static ones currently in use.

Adaptive automation is based on the idea of supporting the human only at those moments in time where his performance is in jeopardy. W. B. Rouse introduced *adaptive aiding* in 1988 [12, p. 431] as a first description of adaptive automation. Rouse stated that adaptive aiding is *a human–machine system-design concept that involves using aiding/automation only at those points in time when human performance needs support to meet operational requirements*. Whether one uses the terms adaptive automation, dynamic task allocation, dynamic function allocation, or adaptive aiding, they all reflect the dynamic reallocation of work in order to improve operator performance or to prevent performance degradation.

In addition to adapting, we believe that the information system should aim to take over those jobs that are less critical in terms of severity or responsibility or that are more repetitive and monotonous (provided it can handle them, of course). Furthermore, the attention of the user should be guided to more relevant high-priority tasks. An information processing system that fulfills all these requirements will aid the user in doing his or her job with maximum effectiveness.

As indicated, the concept of adaptive automation represents the best match between task demands on the one hand and the cognitive resources of the human on the other hand [10], [13]. As a result, various sources in various domains have argued that adaptive automation enhances performance [10], [14], reduces workload [11], [15], and improves situation awareness [16]. These results highlight some of the potential advantages of adaptive automation. Research on adaptive automation has focused on either low-fidelity controlled laboratory experiments [8], [10], [14], [16], [17] or high-fidelity studies in the field of air traffic control [15]. Adaptive automation in laboratory settings allows only a fairly limited scope of assessment and decision making to the human operators. Only limited research has dealt with applying adaptive automation in complex environments like command and control.

This paper describes the deployment of adaptive automation to such a complex domain, more specifically to naval command and control (C2). The work has been part of a larger research program that investigates adaptive teams and adaptive automation for the Royal Netherlands Navy (RNLN). The RNLN is preparing for a future in which a large variety in missions will have to be undertaken and executed in new and demanding environments with smaller crews. The last years have seen a marked shift of operational deployment from open ocean ("blue water") to littoral waters ("brown water") in the vicinity of hostile territory where missions are largely in support of land operations. Littoral operations are characterized by an extended range of threats. Aside from the danger from traditional platforms (military ships and aircraft), the operational area is covered by land-based weapons (guns and missile launchers), and there is an increased chance of asymmetric attacks by small surface vessels and ditto civilian aircraft. Situation assessment is made more difficult due to the presence of numerous neutral and civilian entities, smaller detection ranges (and, thus, reaction times), and stricter rules of engagement. In addition, the amount and complexity of available information continually increase because of, among other things, better sensors and communication and information technology. At the same time, crews are being scaled down due to increasing maintenance and personnel costs.

In order to keep abreast of these developments, the RNLN needs to have flexible teams that can adapt to dynamic operational situations. The advantages of adaptive teams are a better chance of fulfilling mission goals and a more efficient deployment of personnel. Adaptive automation, in turn, is intended to aid the crew in this continuing adjustment to the changing environment.

Our immediate goal was to stretch the envelope of laboratory studies and to come up with a controlled naturalistic environment that offers challenges and opportunities close to the operational circumstances described earlier. The main focus therefore has been to develop a framework architecture that makes adaptive automation realizable for the whole chain of information processing and decision making. In other words, we have attempted to find an approach to adaptive automation that could be implemented in a "real" environment in the foreseeable future. Based on this framework, a demonstration system was built where human-in-the-loop experiments could be performed. In this way, we could assess the viability of the framework vis-à-vis its aims, and we could conduct experiments with our "brand" of adaptive automation and see whether it would live up to its expectations. We started with a number of premises. First, we chose not to design a whole new architecture, but instead, we aimed for the addition of adaptive behavior to existing operational systems. There is a multitude of domain algorithms and knowledge in the real world, and adaptive automation should be applicable without forcing complete redesigns and rewrites of existing software. Analogous to the well-established and utilized paradigm of separating the business model, viewer, and controller [18], [19], we have kept the domain algorithms and the adaptation mechanisms apart. Second, when control is dynamically shifted between the operator and the system, as is the case with adaptive automation, an important question is to what extent should we shift control. We concur with Parasuraman *et al.* [1] that adaptive automation should be flexible enough to accommodate different LoAs for the different stages of the information processing loop. We have therefore striven for a framework that allows such different LoAs. Furthermore, we think that the user rather than the system designer should have the final say as to what LoAs the system is authorized to reach. Parasuraman *et al.* [1] discuss the question of authority using the evaluation criteria *reliability* and *costs of actions*. We believe that human control is particularly important in domains where incorrect decisions lead to outcomes with a high toll (e.g., military command and control). This requires that the automation levels be phrased in terms understandable to the users. Our final starting point was to think about the machine as a virtual team member with responsibilities rather than as a tool [20], [21], where work is divided according to prior agreements and where performance is evaluated after the action. Regular interaction should improve the overall human–machine performance over time.

After an introduction to the domain in Section II, Sections III–VI describe a task allocation model that uses an object-oriented approach and four clearly distinguished LoAs. How such task allocation can be made adaptive is the subject of Section VII. Section VIII shows how our approach to adaptive automation can be interpreted as a set of working agreements in the human–machine team. The question of when to shift autonomy is taken up in Section IX.

II. Domain: Naval Command and Control

As our implementation domain concerns naval command and control, we begin our discussion with a brief introduction to this subject matter. Among other things, a combat management system (CMS) supports the team in the command center of a naval vessel with its tactical work. Basically, this means the continuous execution of the stages of information processing in the naval tactical domain and involves the building of a situational picture of the surroundings of the ship, an understanding of the situation (including an extrapolation into the future), and the possible undertaking of offensive and defensive actions. As already mentioned in the Introduction, this is known as the OODA loop. The loop is similar to the information processing model of Endsley [6] and of Parasuraman *et al.* [1]. The loop can be further subdivided into distinct tasks like correlation, classification, identification, threat assessment, and engagement. Correlation is the process whereby different sensor readings are integrated over time to generate a track. The term track denotes the representation of an external platform within the CMS, including its attributes and properties, rather than the trajectory of the platform. Classification is the process of determining the type of platform of a track (e.g., an F15 and F16 fighter aircraft, a U.K. type 45 air defense destroyer, or a U.S. Arleigh Burke class destroyer), and identification is the attempt to determine its identity or allegiance in terms of it being *friendly*, *neutral*, or *hostile* to the own ship. The threat assessment task assesses the danger a track (or a combination of tracks) represents to the own ship or other platforms it has to defend. At this stage, the information becomes more abstract as singular tracks are bunched together in larger aggregates like military formations and tactical patterns that need to be interpreted as a whole. The engagement task includes the decision to apply various levels of force to neutralize a threat and the execution of the decision.

Track attributes like height and speed need to be monitored continuously because these variables are input for more abstract functions like adherence to an air lane or formation in the identification process. Therefore, monitoring is also part of the duties of a command team.

All the tasks described earlier are currently handled in large part by the crew. Therefore, they must be replicated in algorithmic form in order to be able to automate the process and be made adaptive if we want to end up with an adaptive CMS. An adaptive CMS could provide naval crews with an answer to the looming risk of operator overload due to increasing information processing requirements and manning reduction initiatives. It should be clear from the outset, however, that there is a definite need to have different LoAs for these tasks. Apart from considerations of whether it is technically possible to fully automate all of them or sensible in terms of human-factor considerations, the question of responsibility immediately comes to the fore if we ponder automating the engagement process. There are less qualms with respect to the automation of track correlation or classification, although a bad classification or identification may lead to erroneous actions and, hence, to matters of responsibility as well. We will come back to this question when we discuss working agreements in Section VIII.

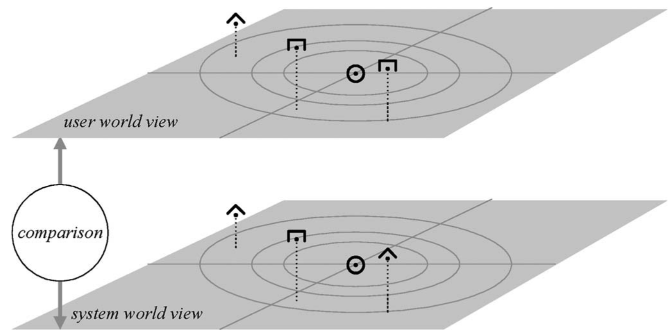

Fig. 1. User and system world views. A comparison of them could lead to interesting conclusions, e.g., a difference between the two views could lead to a user alert. Each view represents three tracks considered by the system's computational power or the user's intelligence. The circles represent different ranges, and the center represents the own ship.

III. System View and User View

Instead of letting a task be performed either by the human, the machine, or a combination of both, we start with a mechanism whereby both parties do their job concurrently. In this way, each party arrives at their own interpretation of the situation, building their respective world views at the same time (see Fig. 1). We thus assume that the system is capable of deriving its own interpretation of the tactical situation and making its own decisions. We will base our adaptive mechanism on the interaction between these two separate world views. One important aspect of this concept is the fact that the system *always* calculates its view and makes its decisions, independent of whether the user is dealing with the same problem or not.

The first step toward an implementation is the provision for "storage space" where the two parties can deposit the information pertaining to their individual view of the world. Thus, we provide two separate data spaces (which we call "system space" and "user space," respectively), where the results of their computational and cognitive efforts can be written to. Having these two distinct world views is akin to two people coming to different conclusions based on the same set of (nonconclusive) data. The system may be viewed as a team member with a limited but scrupulously objective view of the world. The user world view may be likened to the mental picture the user forms "in his head," but we intend here a more concrete tangible implementation.

As such, the user space serves two purposes. First, it is a method to create and maintain a common ground between

TABLE I
SUMMARY OF LoAs USING SIGNALING AND ACCESS TO USER SPACE AS DISCRIMINATORS

Level of Automation	Description	Copy from system space to user space	Signaling
MANUAL	No automation is available or allowed to assist the user. Necessary if automation is not technically or ergonomically possible.	Never, no system space.	no
ADVICE	The human keeps all responsibility. The system view is available for advice but the machine takes no initiative whatsoever ("pull"). If the human does not inspect the system space regularly, he or she may miss important information.	Manual copy; the human must actively adopt the system view	no
CONSENT	The human keeps all responsibility but the machine alerts the human to changes in the situation ("push").	Manual copy after the machine has advised the human.	yes
VETO	The human delegates the responsibility to the machine *unless overruled* (vetoed). Applicable in those cases where risks of wrong interpretation by the machine are large or unacceptable.	Automatic; the machine copies its view to user space unless vetoed by the human.	yes
SYSTEM	The human delegates all responsibility to the machine and there is no interaction between the machine and the human. Only acceptable in cases of low risk.	Automatic; data in system space are quietly copied to user space.	no

the system and the user [21]. When the system and user are engaged in joint activity (i.e., the system becomes a virtual team member), they should maintain a common ground in order to limit coordination breakdowns. Second, when multiple humans are engaged in the process, the user space can serve as a "blackboard" where a graphical picture of the tactical situation unfolds for the entire crew as different people add their information. In other words, it can be used to maintain a common ground between the various human players as well. In this capacity, it reflects the usage in current CMSs, where only user-controlled attributes (e.g., class and identity) are stored. The data in user space are leading in decision making further on in the C2 loop. In other words, a decision to engage a track will depend among other things on its identity *in user space*. The reason why we specifically refer to data in user space rather than the user's point of view will become clear when we start delegating authority to the system.

One advantage of these parallel world views is the increased transparency of the automation, both to the designer and the user. Another advantage is the separation of the domain processing needed to build the system view from the processing we will introduce to get adaptive behavior. A further advantage of the parallel approach is the fact that, in the application domain, we can push the limits of what is algorithmically and computationally possible, because the effects can always be moderated by keeping the world views apart. Finally, the dichotomy between user and system views makes it relatively easy to retroactively add the framework to existing systems. The fact that we need to maintain a "double bookkeeping" is a disadvantage, but we feel that the advantages outweigh this extra load. In addition, both computer speed and capacity continually increase, and the price we pay for this disadvantage shrinks at the same rate.

IV. AUTOMATION LEVELS

Starting from the system and user spaces, our LoAs more or less follow automatically. Because the system view is always available, *advice* is only a key press or mouse click away. If we grant the user access to the system space and provide insight into the reasoning and/or calculations behind the conclusions, the user always has a secondary opinion at his or her fingertips. This readily available opinion represents our lowest LoA (ADVICE). At the next higher LoA, the machine compares both views and signals any discrepancies to the human, thus alerting the user to possible gaps or errors in his situational picture. Of course, system space is still available for inspection. The signaling only occurs when the discrepancy occurs or when additional evidence is found for a deviating system view. This signaling functionality represents our second LoA (CONSENT). ADVICE thus denotes a passive mode, where the user must actively *pull* the opinion of the system from system space; CONSENT describes an active mode, where the system *pushes* its view to the user.

At the higher LoAs, we grant the machine more authority. At our highest LoA (SYSTEM), the machine takes over the responsibility of the human for certain tasks. At a lower LoA (VETO), the machine has the same responsibility, but alerts the human to its actions, thus allowing the latter to intervene. In architectural terms, the machine gets *write* access to user space in VETO and SYSTEM mode and thus is able to inject its own decisions into the human information processing loop, the only difference being that, in VETO mode, the system warns the user of its decision, whereas in SYSTEM mode, the system performs the same action in silence. We thus distinguish four separate LoAs: ADVICE, CONSENT, VETO, and SYSTEM. A fifth, manual, was added to denote the fact that no automation is present for whatever reason (technological, ergonomic, political, financial, etc.). Table I summarizes our automation levels.

We have based this frugal enumeration on the clear distinctions between them in terms of *authority* and *signaling*. Signaling happens when the system wants to inform the user of the fact that there is a discrepancy between the data in system space and user space or that a decision has been made by the system that the user may wish to revoke (VETO). Authority is defined in terms of whether the system has *write* access to user space (see Fig. 2).

The idea is that whoever (or whatever) enters data in user space has the final say with respect to the task associated with the data. For example, if the machine is allowed to change a track identity in user space, it has the authority for the identification process for that track, and if the machine is allowed to add an engagement to a track in user space, it has the authority for the ensuing engagement (i.e., the actual firing of the missiles). It is in this sense that the data in user space are "leading," whereas the user may not actually have that view.

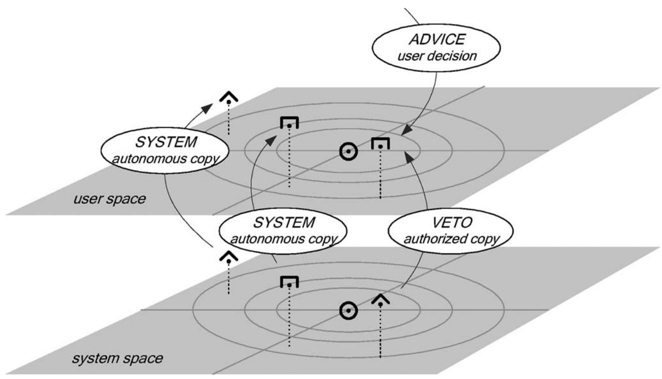

Fig. 2. Authority is defined as equivalent to the machine's write access to user space. Either the machine copies its view to user space (*autonomous copy*) or the human determines what is written in user space. In the latter case, it could be that the human accepts the view of the machine and copies that view (*authorized copy*) or that the user simply prefers his own view (*user decision*).

TABLE II
COMPARISON OF LOAs FROM THE LITERATURE AND OUR WORK

Level	Description	Our level
10	The machine decides everything, acts autonomously, ignoring the human	SYSTEM
9	Informs the human only if the machine decides to	not present
8	Informs the human only if asked to	not present
7	Executes automatically, then informs the human	VETO
6	Allows the human a restricted time to veto before automatic execution	definitely missing
5	Executes the suggestion if the human approves	not quite CONSENT
4	Suggests one alternative	
3	Narrows the selection down to a few	all ADVICE
2	The machine offers a complete set of decisions/action alternatives	
1	The computer offers no assistance	MANUAL

Other automation hierarchies are known (see, for instance, [5] and [22]). Although these hierarchies originally apply to the decision and action selection stages of the information processing loop, there is no problem in applying the hierarchies to the other stages as well. It is not hard to find analogues to our automation levels in them (see Table II derived from the study in [22]) nor would it be difficult to add more levels to our scheme, particularly upward from the point where the machine gets authority to act (levels 6–8). Our differentiation between passive and active machine modes is harder to find among the advisory levels (2–5) where the focus is rather on the multiplicity of the solutions or decisions. We feel that our differentiation between information push and pull adds a dimension to automation that is missing from the other lists.

In our approach, some of the lower levels (levels 2–4) are coalesced. In practice, we generally find multiple answers or solutions to a problem with a preferred or most probable one among them (for example, sensor data are generally so ambiguous that multiple classifications are possible, with one having the highest probability; multiple firing solutions are available with one having the highest kill probability). We usually present all options with the best one highlighted, but we appreciate the other possibilities.

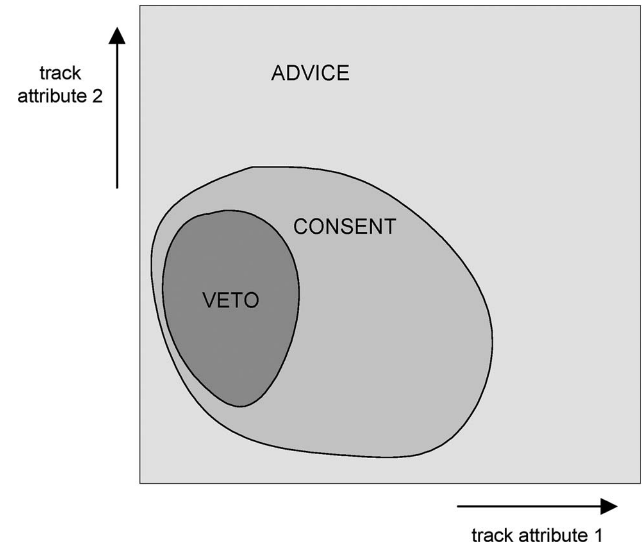

Fig. 3. Track sets related to a certain task defined using two track attributes (for example, identity and range or identity and class). Only VETO, CONSENT, and ADVICE are shown. The SYSTEM set is empty, and the ADVICE set effectively is what remains from the full track set after the other sets have been subtracted.

With system and user space and automation levels, we have enough building blocks to construct an adaptive system by allowing the machine access to the user view in a controlled gradual manner. This would involve switching between different LoAs. Before we do so, however, we want to introduce a more granular division of work between human and machine than one based on tasks.

V. OBJECT-ORIENTED WORK ALLOCATION

Similar to the paradigm shift from functional programming to object-oriented programming, we have found it fruitful to focus on *objects* rather than on tasks (see [47] for a comparable approach). In object-oriented design and programming, objects are abstractions of real-world things or entities that share characteristics and conform to similar sets of rules and policies [23], [24]. Certain domains yield objects that have a very tangible physical nature; other domains concern themselves with more abstract items. In the operation of an airport, objects like airplanes, runways, and air lanes quickly come to the fore, whereas communication systems will be concerned with objects such as data frames and acknowledgements. In this terminology, all objects have *attributes* that lay down the characteristics of the real-world entities that they represent, such as height, temperature, registration number, or location. Furthermore, objects (at least the more interesting ones) can be considered to have a *state* describing in overall terms the condition of the object. Although the state generally can be derived from the attributes of the object, it usually makes sense to add a statelike description to an object. For an air traffic controller, for example, an aircraft is either within or outside his or her air space, waiting for a landing slot, landing or taking off, taxiing, or being parked.

Once we have focused on objects, tasks return into the picture as *the processes related to* the objects. For example,

TABLE III
DIVISION OF WORK BETWEEN USER AND SYSTEM IN TERMS OF TRACK ATTRIBUTES FOR DIFFERENT TASKS

Task	Set Definition	Track Automation Level
Classification	Let the machine automatically classify all tracks with the possibility for the human to adjust or delimit the machine classification	all tracks : SYSTEM
Identification	Let the machine identify all supposed friendly and neutral tracks	friendly and neutral tracks : SYSTEM hostile tracks : CONSENT
Engagement	Let the machine handle all (hostile) missiles automatically by scheduling weapons against such tracks	hostile missile tracks : SYSTEM other tracks : CONSENT

tasks that can be associated with aircraft in the ATC domain are assignments to an air lane or runway, respectively, and continuous collision monitoring. One of the advantages of the use of objects is that it is much easier to pin down what exactly it is that a task is trying to do, namely, to create new objects, to assign values to attributes, to establish (or remove) associations between objects, and so on. A task like "situation assessment" is hard to define, but the sharply outlined purpose of the identification task is to fill in the identity attribute of a track object. The focus on objects does not mean that tasks disappear; it is only that the emphasis is on objects first. In air traffic control, most tasks involve collecting, processing, and inspecting data about aircraft tracks and updating or generating other information related to the same objects.

In the naval domain, the prime objects of interest to the crew are the platforms present in the tactically relevant surroundings of the own ship. Because these platforms are represented by tracks in the CMS, the latter form the actual objects of attention. We have already discussed some of the tasks that are associated with tracks in terms of combat management. They include classification, identification, threat assessment, and engagement, supplemented by behavioral monitoring (often not explicitly mentioned, but executed nevertheless). Aside from tracks, which represent very tangible objects in the ship's surroundings, more abstract objects like formations and task groups, mission objectives, and tactical patterns also play a role in military command and control. It is at these levels of abstraction that warfare officers switch from perception and interpretation of the current situation to predictions of the future situation. In our work on adaptive automation, however, we have focused on tracks as objects for which the information processing can be automated to a reasonable extent.

VI. ASSIGNING AN LoA TO OBJECTS

Aside from the sharper definition of tasks forced upon us by the use of objects, the second advantage of the use of objects is that they allow for a more granular work division between human and machine than a task-oriented approach.

If we consider a task like identification, this task by definition is performed for each object (track) in turn. We can tune the human workload for this task by increasing or decreasing the LoA for each individual object for which the task is being performed, thus parceling out the objects between the human and the machine. The easiest way to achieve this is by means of the object attributes, i.e., by letting the attribute values of the objects determine what LoA is assigned to each object. Put in another way, the object attributes define *sets of objects* where each set has a different LoA

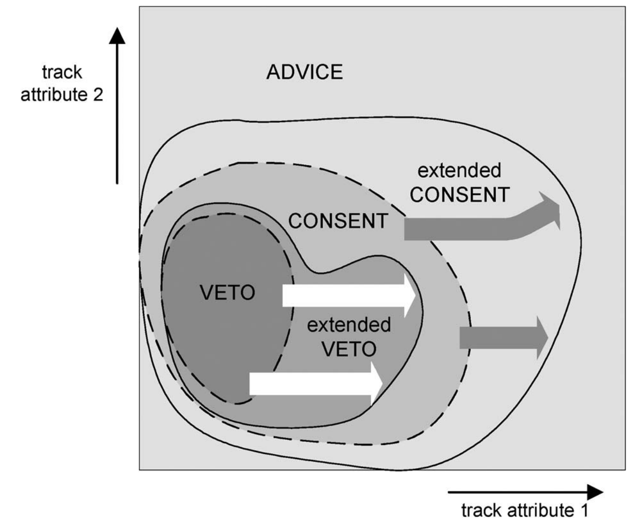

Fig. 4. Adaptable track sets related to a certain task defined using two track attributes. By adjusting the set boundaries, adaptive behavior is implemented, thereby adjusting the workload of the human.

(see Fig. 3). In order to clarify this idea, we have defined track sets for tasks that are to be shared between human and machine (see Table III). The table shows that it is fairly easy to define which tracks will be handled by the user and which by the system. Because the objects and object attributes themselves are part and parcel of the domain, user-friendly domain terminology comes naturally to these work allocations. In the case of identification, for example, the determining attribute is the identity; in the case of engagement, the determining attributes are identity and class (which must be hostile and missile, respectively). In both of these cases, the relevant attributes are ordinal types, but continuous attributes like range or height could be used as well. An additional advantage is the fact that the domain-related attributes allow the user to keep control over those objects he or she deems the most important (e.g., the suspect and hostile tracks).

It should be clear that during an actual operation, some tracks should be brought to the attention of the human (should be signaled). These include tracks in CONSENT mode where the machine has detected a discrepancy between the system view and the user view and also tracks in VETO mode where the machine has already taken steps to correct the situation but has the duty to inform the human of this fact. An attention mechanism therefore has to be part of the adaptive system [25], [26].

VII. ADAPTIVE AUTOMATION

All that remains in order to be able to shift the workload between the human and the machine is to be able to *adjust* the boundaries of the sets (see Fig. 4). When the boundaries of the

TABLE IV
DIVISION OF WORK BETWEEN USER AND SYSTEM FOR DIFFERENT TASKS AND FOR DIFFERENT WORKLOADS

Task	Workload	Set Definition	Track Automation Level
Classification	LOW	Limit the system to giving advice with respect to classification	all tracks : ADVICE
	HIGH	Let the system automatically classify all tracks with the possibility for the user to adjust or delimit the system classification	all tracks : SYSTEM
Identification	LOW	Let the user identify all tracks; limit the system to warning the user when something seems amiss with a track	al tracks : CONSENT
	HIGH	Let the system identify all supposed friendly and neutral tracks	friendly and neutral tracks : SYSTEM hostile tracks : CONSENT
Engagement	LOW	Limit the system to giving advice with respect to engagement	all tracks : CONSENT
	HIGH	Let the system handle all hostile missiles automatically by scheduling weapons against such tracks	hostile missile tracks : SYSTEM other tracks : CONSENT

machine-controlled sets are stretched outward, thus including more tracks, the workload of the user will be reduced as fewer tracks will be his or her immediate responsibility. Even if the responsibility for some of the tracks remains the user's, signaling important changes with regard to these tracks will relieve their workload as less monitoring is required. When the boundaries of the track sets are contracted, on the other hand, the workload on users will increase, because more tracks become their responsibility. The boundaries are changed by changing the attribute values that define the sets. For example, by reducing the range at which tracks are identified manually, the set of tracks handled by the machine (either by CONSENT, by VETO, or fully autonomous) is increased, and the workload of the user is reduced in proportion.

We thus get adaptive automation by adjustment of the automation levels of the objects, which is accomplished by adjusting the boundary values of the relevant object attributes for the sets concerned. In this way, we effectively transfer objects from the human to the machine or vice versa. In Table IV, two different configurations are shown: one for a low workload and another for a higher workload. The human can set these configurations offline using *working agreements*.

VIII. WORKING AGREEMENTS

The moment that high-workload conditions are encountered, the adaptive system will increase its authority and assume responsibility for some of the work that was previously the responsibility of the human. At such a time, however, the adaptive system should not negotiate this shift with the human. If the machine would start a discourse with the human, an extra task would be initiated that would further decrease the latter's performance. It is for this reason that working agreements are introduced.

The general idea behind our concept of working agreements is that, as the human and the machine work together to achieve common goals, both should act as team members. In the field of human teamwork, several people have elaborated the concept of *working agreements* (see, for instance, [27] and [28]) that are a means of balancing workload among team members based on previously agreed upon arrangements. These agreements aid team members in providing each other with information in time, jointly solving problems, and assisting each other during periods of heavy workloads.

In our framework, we regard the machine as a virtual team member [20], [21] and the different settings of the set boundaries as working agreements. Table IV shows working agreements for three tasks during low- and high-workload situations. Based on predefined tasks and object attributes, the human instructs the machine prior to a mission on the extent to which authority may be increased. Furthermore, we propose that the human evaluates these working agreements during debriefing sessions following work shifts. Based on the human's most recent experience during the debriefing sessions, the human can improve his or her dealings with the machine by adjusting incomplete or cumbersome agreements.

The role of the working agreements is larger than merely regulating the workload between human and machine. They can reflect the opinions of the system designers and users on the proficiency of the software and be used to establish trust in the workings of the machine by starting at fairly low LoAs. They will also mirror thoughts on how much authority is ultimately delegated to the machine under the political and strategic constraints of the mission. Peace-keeping operations will likely involve lower maximum LoAs than a full-out war will allow. The considerations that are taken into account when drafting these working agreements are the same as the secondary evaluation criteria brought to the fore by Parasuraman *et al.* [1] for the LoA of information processing: reliability and the costs of actions. In the military domain, there is a continuous balancing act between the need to further automate processes that require fast reaction times or complicated algorithms (e.g., missile defense) and the need to keep the responsibility for the consequences of these processes firmly in human hands. The working agreements advocated by us allow this to be realized for each distinct task. The primary evaluation criteria described in [1] (mental workload, situational awareness, and avoidance of complacency) should be met—to some degree at least—by the workings of the adaptive system itself.

IX. TRIGGERING ADAPTATION

One of the challenging factors in the successful development of adaptive automation is the question of *when* changes in LoA must be effectuated. "Workload" generally is the key concept to invoke such a change of authority. Most researchers, however, have come to the conclusion that "*workload is a multi-dimensional multifaceted concept that is difficult to define. It is generally agreed that attempts to measure workload relying on a single representative measure are unlikely to be of use*" [29]. The definition of workload as "*an intervening variable similar to attention that modulates or indexes the tuning between the demands of the environment and the capacity of the operator*" [30] seems to capture the two main aspects of workload, i.e., the *capacity* of humans and the *task demands* made on them. The workload increases when the capacity decreases or the task demands increase. Both the capacity and task demands are not fixed entities, and both are affected by many factors. Skill and training are two factors that increase capacity, for example, whereas capacity decreases when humans become fatigued or have to work under extreme working conditions for a prolonged period.

If measuring workload directly is not a feasible way to trigger the adaptive automation mechanism, other ways must be found. Wilson and Russell [14] define five strategies based on a division by Parasuraman *et al.* [10]. They state that triggers may be based on critical events, operator performance, operator physiology, models of operator cognition, and hybrid models that combine the other four techniques.

The occurrence of *critical events* can be used to change to a new LoA. Critical events are defined as incidents that could endanger the goals of the mission. Scerbo [11] describes a model where the system continuously monitors the situation for the appearance of critical events, and the occurrence of such an event triggers the reallocation of tasks. Inagaki has published a number of studies [31], [32] where a probabilistic model was used to decide who should have authority in the case of a critical event.

A decline in *operator performance* is widely regarded as a potential trigger. Such an approach measures the performance of the human over time and regards the degradation of the performance as an indication of a high workload. Many experimental studies derive operator performance from the measured performance of a secondary task [8], [16], [17], [33]. Although this approach works well in laboratory settings, the addition of an artificial secondary task in a real-world setting is clearly unfeasible.

Physiological data from the human are employed in various studies [14], [34]–[37]. The capability of human beings to adapt to variable conditions, however, may distort accurate measurements [38]. There are two reasons why physiological measures are difficult to use in isolation. First of all, the human body responds to an increased workload in a reactive way. Physiological measurements therefore provide the system with a delayed workload state of the operator instead of the desired real-time measure. Second, it is possible that physiological data signal a high-workload level but not necessarily a poor performance. This is the case when operators put in extra effort to compensate for increases in task demand. At least several measurements (physiological or otherwise) are required to get rid of ambiguities.

The fourth approach uses *models of operator cognition*. These models are approximations of human cognitive processes for the purpose of prediction or comprehension of operator state and workload. The winCrew tool [39], for example, implements the multiple-resource theory [40] to evaluate function allocation strategies by quantifying the moment-to-moment workload values. Alternatively, the human's interactions with the machine can be monitored and evaluated against a model to determine when to change LoAs. In a similar approach, Geddes [41] and Rouse *et al.* [42] base adaptive automation on the human's intentions as predicted from patterns of activity. Alternatively, previous work [43] suggests a model of operator cognition based on Neerincx's [44] three independent cognitive task load factors. These factors can be derived from environmental characteristics and utilize a combined vector as a predictive measure.

Last, most of the aforementioned techniques could be applied together, and in combination, they could resolve ambiguities better than a single one would do.

The *perceived workload or operator performance* might serve as yet another trigger for a change in automation, although the system would more properly be called adaptable in such a case. The actual request to adapt could come either from the overloaded human in person or from another member of the crew (e.g., a supervisor). Against the first option counts the fact that he or she is already overloaded and any such actions would likely be neglected. The second option therefore seems more feasible, but would probably involve independent measurements of workload to support the supervisor's view, leading to a combination of this method and other methods.

We propose that an adaptive allocation scheme be based on operator performance in combination with a model of operator cognition. One of the advantages of the object-oriented framework described in the previous sections is that it offers a number of hooks for such a combined approach. The first hook is the difference between user space and system space. Although differences will occur as the human and machine do not necessarily agree in their world views, an increasing skew between the two views is a reasonable indication that the human is struggling with his or her workload. This is reflected, in turn, in an increasing number of outstanding warnings and an increasing response time to such warnings. One disadvantage of this approach is the sparseness of data as the number of significant events per time unit can be rather small. The second hook is based on the numbers of objects and is utilized for predicting the workload of the human. The total number of tracks provides an indication of the *volume* of information processing, whereas the number of unknown, suspect, and hostile tracks is an indication of the *complexity* of the situation. Of these, the suspect and unknown tracks require the most cognitive resources due to their ambiguous nature. The combination of these factors results in an estimate of the perceived workload. In case the human is forced to switch between different tasks, the task load model [44] addresses the demands of such attention switching by adding this factor to the workload estimate.

Volume and complexity (indicators of cognitive workload) can also be used to pedal down the automation automatically as things quiet down. Letting the user initiate the withdrawal of the software is feasible as well. However, there is a distinct possibility that when things slow down, the human will let the machine remain in charge, thereby falling into the trap of user complacency and loss of situational awareness that adaptive automation was in part designed to avoid.

X. Conclusion

The framework for adaptive automation we have described aims at maintaining an operator's workload at a manageable level. This is achieved by adjusting the LoA for the tasks the human must perform or, more specifically, by shuttling the objects these tasks are concerned with between sets of different LoAs. Which objects are shuttled to which sets depends on the attributes of the objects, allowing fine-grained control over the transferred work that can be expressed in terms of the domain itself (for example, all hostile tracks or only tracks with high velocities at lower altitudes). Depending on the LoA of an object, a task is under control either of the human or the machine, while the latter may additionally signal discrepancies or gaps in the human's view. The boundaries of the LoAs are agreed to in advance between human and machine, utilizing the concept of working agreements, in order to minimize surprises and to ensure the fact that the system does not take on too much responsibility.

This approach allows a gradual delegation of responsibility to the machine while keeping the human in firm control of those tasks and objects that are regarded important. The human can delegate complete autonomy to the machine or express a wish of being informed about decisions of the machine, modes of automation that more or less correspond to automation levels 7 and 10, respectively, in the automation hierarchy of Wickens *et al.* [22]. Alternatively, the human can keep control over critical items while being optionally informed about potential problems, modes that roughly correspond to levels 2–5 of the hierarchy. For our taxonomy, we make a sharper distinction between silent and active modes than Wickens *et al.* do. Although the overall picture is the same, we created this sharp distinction from the user-centered perspective in that it disambiguates on how the human wants to cooperate with the machine. The LoAs will be maximized either by responsibility (reflecting the responsibility the users or higher authorities are willing to delegate to the machine) or the comprehensiveness and quality of the software.

Responsibility for the adaptation mechanism itself is delegated according to the same hierarchy; while some researchers keep the human firmly in command [45], others are more inclined to let the machine balance the workload [14], [17]. Owing to the fine-grained working agreements made possible by the object-oriented approach, we feel justified in advocating a highly autonomous ("SYSTEM mode") adaptation process. Because the circumstances under which the system adjusts itself and the limitations of these adjustments are proscribed in advance, the human knows what to expect. When the human starts being overwhelmed by the situation, we feel that the machine should step in without forcing the human to think about a better distribution of work or, worse, start a negotiation with the machine with respect to the amount of adjustment. Such an approach stands or falls with a good estimate of the workload, however, and this remains a topic for further research for our approach and similar approaches that depend on workload estimates.

An additional advantage with our form of adaptive automation and the underlying working agreements is the fact that *trust* in the machine can be built by starting the automation at relatively low levels, with mainly advice and warnings coming from the machine. When humans start to see the benefits (and limitations) of the machine, the LoAs can be increased to the point where the operators are comfortable with the level of support at each workload level.

What makes all this possible is the fact that the machine continuously updates its own view of the situation (in "system space") and is only restricted in the *expression* of this view, be it by having an advice ready, signaling, or taking autonomous action. The major assumption is that the machine is indeed capable of calculating its own view of the situation. If this is not the case, the LoA for the corresponding tasks must be set to low (ADVICE or CONSENT) or automation must be considered absent (MANUAL). Of course, this applies to adaptive or adaptable automation in general: One cannot have adaptive automation without automation (the underlying algorithms) to start with.

Upon reflection, an object-oriented approach toward adaptive system behavior seems quite natural. The notion that the amount of work relates in some way to the number of objects and that work can be divided by distribution of objects is an intuitive one. The approach has the additional advantage that it tends to closely match the domain terminology because the objects form the natural core of the domain. The idea of the machine continuously updating its own view of the situation independently from the human perhaps seems more farfetched. Ordinarily, updates would take place only in response to a request or because the human has delegated some of the work to the machine. In this paper, we have taken both ideas and pushed them to their logical limits, leading to a comprehensible, holistic, and consistent framework for adaptive automation.

Recently, we conducted a first quantitative evaluation of the adaptive framework with eight officers from the RNLN to investigate some of the claims with respect to transparency and usability [46]. The participants were four warfare officers and four warfare assistants from the RNLN each with extensive operational experience on naval vessels. The participants were given the primary mission goal to build a recognized maritime picture of the surroundings of the ship (i.e., to detect and identify all platforms present), to investigate suspicious entities, and to defend the ship against threats. The participants were offered four scenarios that were developed in close cooperation with tactical experts from the RNLN. Two of the four scenarios centered around more or less traditional air and surface warfare in a high-tension peace-enforcing situation, while the other two dealt with an asymmetric threat against a civilian smuggling background. The participants were offered two scenarios in the adaptive automation mode, while the other two scenarios were run in a fixed automation mode. The distribution of automation mode, officer rank, and scenario was balanced over all subjects using a Latin square design. A number of variables were measured, among which the subjective workload, a number of performance measures, and a number of expert ratings. Space does not permit detailed reporting here, but the results from the evaluation look promising as they clearly demonstrate a positive performance effect when the system adaptively aids the operator. It shows, for instance, in improved *reaction times* in general and reaction times to hostile tracks in particular. The numbers of *signaled tracks* and *pending tracks* (tracks waiting for identification) per time unit are also significantly lower with adaptive than with fixed automation. As the latter effect can be partly ascribed to the autonomous actions of the system, it is best interpreted as *a performance increase of the human–machine combination*, but it still corresponds to a lower workload of the human because only problematic tracks remain and he or she can be confident that the other unobserved tracks do not pose a problem.

To our surprise, the results did not reveal an effect of the automation mode on the subjective workload. The operators performed better but did not rate their workload lower, neither did neutral observers. Our tentative explanation at this moment is that the operators reallocated a part of their cognitive resources to a better understanding of the situation. It appears that in the fixed-automation runs, the subjects took more risk by willfully ignoring contacts at larger ranges (indicated by the number of pending contacts), and they communicated less and generally found little time to "lean back" to try interpret the situation. The adaptive mode took away some of the less important work, thereby resolving cognitive limitations.

Various studies have pointed to potential problems with adaptive systems due to incorrect or untimely behavior. Our study shows no decrease in situational awareness, leading to the conclusion that the type of aid and the timing of the aid were applied appropriately. To some subjects, it seemed that the system "read their mind" when taking over work, although other participants felt that there was no clear correlation between workload and time of adaptation. Obviously, the triggering methods and the feedback need still be improved for operational use, or the human must get more involved after all. Because the latter involvement would merely mean determining the *time* of adaptation, the amount of extra work would be relatively small and therefore acceptable.

We should stress that, although all data point to an improvement in the performance of the subjects, it is difficult to ascribe this to the *adaptive* automation rather than the mere fact that the automation was *higher*—even for limited time periods—and thus was helping the subjects more. What seems clear, however, is the fact that adaptive automation does not impair people's abilities to perform their tasks in a complex environment. With respect to the question posed by some subjects, i.e., why not simply use more automation all the time, there is the counterargument that increased automation decreases the situational awareness of operators so that their reactions when things do start to get out of hand are slow, hesitating, or outright wrong. This argument strongly argues for less automation when the crew is capable of doing the job themselves. We foresee, however, that the discussion about the lowest LoA will continue within our focus group until it is shown without doubt that higher automation levels during periods of slack do impair people's effectiveness.

Our work shows that it is possible to define an adaptive framework using the concept of separate system and user spaces and an object-oriented approach on top of the model of Parasuraman *et al.* [1]. Aside from its use in a naval command and control system, the framework looks applicable across a wide range of complex systems, both military and civilian, because the inherent assumptions about the problem domain are limited in scope. Obviously, we must be able to automate the domain to some extent, we must be able to model it in objects (preferably large numbers of similar ones), and we must have a user view represented in the system. The need for a user view may seem to limit the generalizability of our approach as some domains do not have such a view represented within the machine. Whether this lack means a true limitation to our method or signals a shortcoming in the current approach in human machine ensembles remains open for discussion. However, Klein *et al.* [21] argue that maintaining a common ground is a key requirement of joint activity of human–machine ensembles. A user space is one method of achieving this.

The framework therefore presents other researchers and developers with a fine-grained architecture that they can use in their own work to get adaptive automation up and running in a naturalistic setting. It enables a granular way of allocating work between humans and machines while speaking in terms that match the operational knowledge of the user. Moreover, it gives a new perspective at the LoAs that takes into account passive and active modes of the system.

Acknowledgment

The authors would like to thank the following colleagues from TNO Defence, Security and Safety who collaborated in the design and implementation for their continuous support and reflective ideas: J. Lindenberg, B. Bierman, L. Prins, R. van der Meer, and K. Houttuin. The authors would also like to thank J. M. Schraagen and W. Treurniet for their useful comments and the officers of the RNLN that participated in the evaluation and following discussions for their cooperation. Finally, the authors would like to thank the reviewers for their insightful remarks that helped to improve this paper.

References

[1] R. Parasuraman, T. B. Sheridan, and C. D. Wickens, "A model for types and levels of human interaction with automation," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 30, no. 3, pp. 286–297, May 2000.

[2] M. R. Endsley, B. Bolte, and D. G. Jones, *Designing for Situation Awareness*. New York: Taylor & Francis, 2003.

[3] T. E. de Greef, K. van Dongen, M. Grootjen, and J. Lindenberg, "Augmenting cognition: Reviewing the symbiotic relation between man and machine," in *Foundations of Augmented Cognition*, vol. 4565, *Lecture Notes in Computer Science*, D. D. Schmorrow and L. M. Reeves, Eds. Berlin, Germany: Springer-Verlag, 2007, pp. 439–448.

[4] P. M. Fitts, *Human Engineering for an Effective Air Navigation and Traffic Control System*. Washington, DC: Nat. Res. Council, 1951.

[5] T. B. Sheridan and W. L. Verplank, "Human and computer control of undersea teleoperators," Man-Machine Lab., MIT, Cambridge, MA, 1978.

[6] M. R. Endsley, "The application of human factors to the development of expert systems for advanced cockpits," in *Proc. 31st Annu. Meeting Human Factors Soc.*, 1987, pp. 1388–1392.

[7] R. Coram, *Boyd: The Fighter Pilot Who Changed the Art of War*. New York: Little, Brown & Company, 2002.

[8] M. P. Clamann, M. C. Wright, and D. B. Kaber, "Comparison of performance effects of adaptive automation applied to various stages

of human–machine system information processing," in *Proc. 46th Annu. Meeting Human Factors Ergonom. Soc.*, Baltimore, MD, 2002, pp. 342–346.

[9] M. R. Endsley and D. B. Kaber, "Level of automation effects on performance, situation awareness and workload in a dynamic control task," *Ergonomics*, vol. 42, no. 3, pp. 462–492, Mar. 1999.

[10] R. Parasuraman, M. Mouloua, and R. Molloy, "Effects of adaptive task allocation on monitoring of automated systems," *Human Factors*, vol. 38, no. 4, pp. 665–679, Dec. 1996.

[11] M. Scerbo, "Theoretical perspectives on adaptive automation," in *Automation and Human Performance: Theory and Applications*, R. Parasuraman and M. Mouloua, Eds. Mahwah, NJ: Lawrence Erlbaum, 1996, pp. 37–63.

[12] W. B. Rouse, "Adaptive aiding for human/computer control," *Human Factors*, vol. 30, no. 4, pp. 431–443, Aug. 1988.

[13] C. D. Wickens and J. G. Hollands, *Engineering Psychology and Human Performance*. Upper Saddle River, NJ: Prentice-Hall, 2000.

[14] G. F. Wilson and C. A. Russell, "Performance enhancement in an uninhabited air vehicle task using psychophysiologically determined adaptive aiding," *Human Factors*, vol. 49, no. 6, pp. 1005–1018, Dec. 2007.

[15] B. Hilburn, P. G. Jorna, E. A. Byrne, and R. Parasuraman, "The effect of adaptive air traffic control (ATC) decision aiding on controller mental workload," in *Human-Automation Interaction: Research and Practice*, M. Mouloua, J. Koonce, and V. D. Hopkin, Eds. Mahwah, NJ: Lawrence Erlbaum, 1997, pp. 84–91.

[16] D. B. Kaber, C. M. Perry, N. Segall, C. K. McClernon, and L. J. Prinzel, III, "Situation awareness implications of adaptive automation for information processing in an air traffic control-related task," *Int. J. Ind. Ergonom.*, vol. 36, no. 5, pp. 447–462, May 2006.

[17] D. B. Kaber and J. M. Riley, "Adaptive automation of a dynamic control task based on secondary task workload measurement," *Int. J. Cogn. Ergonom.*, vol. 3, no. 3, pp. 169–187, Jul. 1999.

[18] G. E. Krasner and S. T. Pope, "A cookbook for using the model-view-controller user interface paradigm in Smalltalk-80," *J. Object-Oriented Program.*, vol. 1, no. 3, pp. 26–49, Aug./Sep. 1988.

[19] F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad, and M. Stal, *Pattern-Oriented Software Architecture: A System of Patterns*. West Sussex, U.K.: Wiley, 1996.

[20] K. Christoffersen and D. D. Woods, "How to make automated systems team players," *Adv. Human Perform. Cogn. Eng. Res.*, vol. 2, pp. 1–12, 2002.

[21] G. Klein, D. D. Woods, J. M. Bradshaw, R. R. Hoffman, and P. J. Feltovich, "Ten challenges for marking automatio a 'team player' joint human-agent activity," *IEEE Intell. Syst.*, vol. 19, no. 6, pp. 91–95, Nov./Dec. 2004.

[22] C. D. Wickens, A. S. Mavor, R. Parasuraman, and J. P. McGee, *The Future of Air Traffic Control: Human Operators and Automation*. Washington, DC: Nat. Acad. Press, 1998.

[23] J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy, and W. Lorensen, *Object-Oriented Modeling and Design*. Upper Saddle River, NJ: Prentice–Hall, 1991.

[24] S. Shlaer and S. J. Mellor, *Object Lifecycles: Modeling the World in States*. Upper Saddle River, NJ: Yourdon Press, 1992.

[25] H. F. R. Arciszewski, T. E. de Greef, and J. H. van Delft, *An Attention Allocation System Using an Object-Oriented Task Model*. Soesterberg, The Netherlands: TNO Defence, Security Safety, 2005.

[26] W. Nugent and R. Obermayer, *Human Computer Interaction Design Guidelines for an Alert Warning and Attention Allocation System I: Literature Review*. San Diego, CA: Space Naval Warfare Syst. Center, 2000.

[27] J. A. Cannon-Bowers and E. Salas, "Team performance and training in complex environments: Recent findings from applied research," *Curr. Dir. Psychol. Sci.*, vol. 7, no. 3, pp. 83–87, Jun. 1998.

[28] P. C. Rasker and E. W. Willeboordse, "Werkafspraken in de commandocentrale [Work arrangements in the Combat Information Centre]," TNO: Human Factors, Soesterberg, The Netherlands, 2001.

[29] D. Gopher and E. Donchin, "Workload—An examination of the concept," in *Handbook of Perception and Human Performance*, vol. 2, K. Boff and J. Kaufmann, Eds. New York: Wiley-Interscience, 1986, pp. 4141–4949.

[30] B. H. Kantowitz, "Mental workload," in *Human Factors Psychology*, P. A. Hancock, Ed. New York: North Holland, 1987, pp. 81–121.

[31] T. Inagaki, "Situation-adaptive autonomy: Dynamic trading of authority between human and automation," in *Proc. XIVth Triennial Congr. Int. Ergonom. Assoc., 44th Annu. Meeting Human Factors Ergon. Assoc. 'Ergonomics for the New Millennium'*, San Diego, CA, 2000, pp. 13–16.

[32] T. Inagaki, "Situation-adaptive autonomy for time-critical takeoff decisions," *Int. J. Model. Simul.*, vol. 20, no. 2, pp. 175–180, 2000.

[33] D. B. Kaber, M. C. Wright, L. J. Prinzel, and M. P. Clamann, "Adaptive automation of human–machine system information-processing functions," *Human Factors*, vol. 47, no. 4, pp. 730–741, 2005.

[34] N. R. Bailey, M. W. Scerbo, F. G. Freeman, P. J. Mikulka, and L. A. Scott, "Comparison of a brain-based adaptive system and a manual adaptable system for invoking automation," *Human Factors*, vol. 48, no. 4, pp. 693–709, 2006.

[35] E. A. Byrne and R. Parasuraman, "Psychophysiology and adaptive automation," *Biol. Psychol.*, vol. 42, no. 3, pp. 249–268, Feb. 1996.

[36] J. A. Veltman and A. W. K. Gaillard, "Physiological workload reactions to increasing levels of task difficulty," *Ergonomics*, vol. 41, no. 5, pp. 656–669, May 1998.

[37] L. J. Prinzel, F. G. Freeman, M. W. Scerbo, P. J. Mikulka, and A. T. Pope, "A closed-loop system for examining psychophysiological measures for adaptive task allocation," *Int. J. Aviat. Psychol.*, vol. 10, no. 4, pp. 393–410, Jan. 2000.

[38] J. A. Veltman and C. Jansen, "The adaptive operator," in *Proc. Human Perform., Situation Awareness Autom. Technol. Conf.*, Daytona Beach, FL, 2004.

[39] R. D. Archer and J. F. Lockett, "WinCrew—A tool for analyzing performance, mental workload and functional allocation among operators," in *Proc. 1st Int. Conf. ALLFN*, Galway, Ireland, 1997, pp. 157–165.

[40] C. D. Wickens, "Processing resources in attention," in *Varieties of Attention*, R. Parasuraman and D. R. Davies, Eds. Orlando, FL: Academic, 1984, pp. 63–101.

[41] N. D. Geddes, "Intent inferencing using scripts and plans," in *Proc. 1st Annu. Aerosp. Appl. Artif. Intell. Conf.*, Dayton, OH, 1985, pp. 160–172.

[42] W. B. Rouse, N. D. Geddes, and R. E. Curry, "Architecture for interface: Outline of an approach to supporting operators of complex systems," *Human-Comput. Interact.*, vol. 3, no. 2, pp. 87–122, 1987.

[43] T. E. de Greef and H. F. R. Arciszewski, "A closed-loop adaptive system for command and control," in *Proc. 3rd Int. Conf. FAC, Held as Part of HCI International*, Beijing, China, 2007, pp. 276–285.

[44] M. A. Neerincx, "Cognitive task load design: Model, methods and examples," in *Handbook of Cognitive Task Design*, E. Hollnagel, Ed. Mahwah, NJ: Lawrence Erlbaum Associates, 2003, pp. 283–305.

[45] C. A. Miller and R. Parasuraman, "Designing a flexible interaction between humans and automation: Delegation interfaces for supervisory control," *Human Factors*, vol. 49, no. 1, pp. 57–75, Feb. 2007

[46] T. E. de Greef, H. F. R. Arciszewski, J. Lindenberg, and J. H. van Delft, *Adaptive Automation Evaluated*. Soesterberg, The Netherlands: TNO Defence, Security Safety, 2007.

[47] F. Bolderheij, "Mission-driven sensor management: Analysis, design, implementation and simulation," M.S. thesis, Delft Univ. of Technol., Delft: The Netherlands, 2007.

Henryk F. R. Arciszewski received the M.Sc. and Ph.D. degrees in physics from the University of Utrecht, Utrecht, The Netherlands.

In 1984, he started his career with the Netherlands Organization for Applied Scientific Research (TNO), Defence, Security and Safety, Information and Operations, The Hague, The Netherlands, where he has been involved with work for the Royal Netherlands Army and Navy ever since. His research interests are in the areas of automation of combat management systems and decision support systems for the army and navy. He has been involved with the research and development of a decision aid for the deployment of electronic warfare countermeasures for the Royal Netherlands Navy that led to an operational system in the fleet in 2003. Since 2000, he has been collaborating with TNO Human Factors on new ways of human–machine collaboration and human–machine interfacing and the adaptive behavior of C2 and decision support systems. He has published a number of papers on these aspects.

Tjerk E. de Greef received the B.S. degree in electrical engineering from HAN University, Arnhem, The Netherlands, in 1999 and the M.Sc. degree in information science from the University of Utrecht, Utrecht, The Netherlands, in 2003.

After receiving his M.Sc. degree, he was offered a position at the Netherlands Organization for Applied Scientific Research (TNO), Defence, Security and Safety, Human Factors, Soesterberg, The Netherlands, where he was mainly involved in human-factor-related research for the Royal Netherlands Navy, in the area of adaptive automation in combat management systems. During this four-year research project, he published a number of papers. Recently, he accepted a Ph.D. position at the Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, Delft, The Netherlands, in the area of collaboration at a distance for urban search and rescue missions.

Jan H. van Delft received the M.Sc. degree in experimental psychology and ergonomics from the University of Utrecht, Utrecht, The Netherlands, in 1981.

After his study, he started his career with the Netherlands Organization for Applied Scientific Research (TNO), Institute for Perception, later renamed to TNO Human Factors and now part of the TNO Defence, Security and Safety. As a Senior Research Scientist, Project Manager, and Program Manager, he was involved in many human-factor-related projects for the Royal Netherlands Navy and Army. He has been a member of several NATO research study groups. In 2007, he was offered a position at the ForceVision Department, Centre for Automation of Mission Critical Systems, part of the Netherlands Defence Material Organization and supplier of combat management systems for the Royal Netherlands Navy, Den Helder, The Netherlands, where he is involved in the development of intelligent user-cooperation components. His main research interests include decision support and human–computer interface development for combat management and command information systems.