Adaptive Automation using an object-orientated task model

Tjerk de Greef
TNO Defense, Security and Safety
Tjerk.deGreef@tno.nl

Henryk Arciszewski
TNO Defense, Security and Safety
Henryk.Arciszewski@tno.nl

Jan van Delft
TNO Defense, Security and Safety
Jan.vanDelft@tno.nl

ABSTRACT

On naval ships the crew is assisted by a combat management system (CMS) in their assessment of the tactical situation around the ship and their subsequent actions. Today's information-rich littoral environments force operators to divide attention between different information items. The risk of overload is imminent and the concept of *adaptive automation* can aid the operator. The first concept refers to the dynamic division of work between operator and system. We propose an object-oriented method to realize the concepts of *adaptive automation* in a CMS. The object-oriented approach allows a sufficiently fine-grained division of work between system and operator that the latter's workload is reduced while keeping his situational awareness intact.

Keywords

Adaptive Automation, Human Factors, Combat Management System.

INTRODUCTION

On naval ships the crew is assisted by a combat management system (CMS) in their assessment of the tactical situation around the ship and their subsequent actions. A continuous technology push can lead to innovative but at the same time complex systems. Many information systems, like a CMS, have evolved this way through time to become more complex and autonomous. Technological development enabled the crew to work more efficiently or effectively (or both) using such a system. In such information-rich environments, however, a competition for the users' attention is going on between different information items, possibly 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 proven repeatedly, moreover, that aiding the crew by as much automation as possible does not necessarily lead to a better performance (Endsley, Bolté & Jones, 2003; Parasuraman, Mouloua & Molloy, 1996). It is therefore a question of continuous research where to draw the line between ongoing automation and a truly improved man-machine combination.

ADAPTIVE AUTOMATION

If a user is starting to get overwhelmed by the situation, a CMS that is capable of autonomous decision making could intervene and reallocate part of the work to itself so that the workload of the user is lessened and he or she is again up to the task. The CMS should aim to take over those jobs that are less critical in terms of severity or responsibility or those are more repetitive and monotonous (provided it can handle them, of course) and the attention of the user should be guided to more relevant, high priority tasks for optimal task performance. An information processing system that fulfills this requirement will aid the user in doing his job with maximum effectiveness. Thanks to this support the user will experience a lowered cognitive workload and the much feared information overload will hopefully be avoided.

This automation approach is called *adaptive automation* and currently receives a lot of attention in the academic community. Adaptive automation takes as its starting point that the division of labor between man and machine should not be static but dynamic in nature. It is based on the idea of supporting the crew only at those moments in time where their performance is in jeopardy. W. B. Rouse introduced *adaptive aiding* in 1988 (Rouse, 1998) as a first description of adaptive automation. He 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.* Today's literature uses the term adaptive automation when control is dynamically shifted between the operator and the system. A so-called adaptive system is thought to represent a better solution to the problem of function allocation than the static one currently in use. Whether one uses the terms adaptive automation, dynamic task allocation, dynamic function allocation or adaptive aiding, it all reflects the dynamic reallocation of work in order to improve operator performance or to prevent performance degradation.

This paper focuses on the utilization of adaptive automation in the command and control domain. The research goal is to aid the functionary performing its tasks at those moments in time the functionary feels overwhelmed by the situation. Adaptive automation helps thereby to improve the overall performance. A number of interesting research questions are to be answered. First of all, the question of what exactly is to be shifted between man and machine. Although it is easy to say ´tasks´, a task is harder to share between operator and system than apparent at first. Second, the determination of the boundaries of control and the third question concerns the point in time when to shift autonomy are subjects of research.

We believe that an object oriented approach, that leads to set orientations, in combination with working agreements adequately will address dissolve the first two research questions. The third research item is captured in the development of a performance model that can be used to trigger adaptive automation.

FROM TASKS TO OBJECTS

People often refer to the work the crew of a naval ship has to deal with as the 'tasks' they must perform. In a command centre of a ship such tasks are generally assigned to specific operators. For instance, one or more crew members are responsible for the building or compilation of the tactical air or surface picture and hence it can be said that their task is 'air' or 'surface picture compilation'. If we want to prioritize or reallocate work, however, such a task-oriented approach soon runs into problems. Either because tasks are defined in a way that is too broad for our purposes or because we lose ourselves in a hierarchical forest of interrelated subtasks and their sub-subtasks when trying to describe what each member of the team must do. Finally, tasks are hard to parcel out between operators and system, if we want to dynamically allocate them to different resources.

In a paradigm shift not unlike that from functional programming to object-oriented programming, we stop focusing on tasks but rather start thinking in terms of *objects* of interest to the crew of a naval vessel. 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 (Shlaer and Mellor, 1992; Rumbaugh et al, 1991). 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, acknowledgements, and so on. 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* (also known as a state vector) describing in overall terms the condition the entity is in. Although the state generally can be derived from the attributes of the object, it usually makes sense to add a state-like description to an object. For an air controller, an aircraft would be within or outside his 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 'things to do' with these objects. For example, tasks that can easily be associated with aircraft in the air traffic control domain are assignment to an air lane or runway and continuous collision monitoring. The advantage of objects is that it is much easier to pin down what exactly it is that a task is trying to do, namely assign values to attributes, create new objects, establish (or remove) associations (relations) between objects, and so on. The focus on objects therefore does not mean that the concept of tasks suddenly disappears; it is only that the primary emphasis is on objects first. Furthermore, *tasks are explicitly linked to objects* (compare Figure 1). In air traffic control, tasks thus involve collecting, processing and inspecting data about aircraft tracks and updating or generating other information related to the same objects.

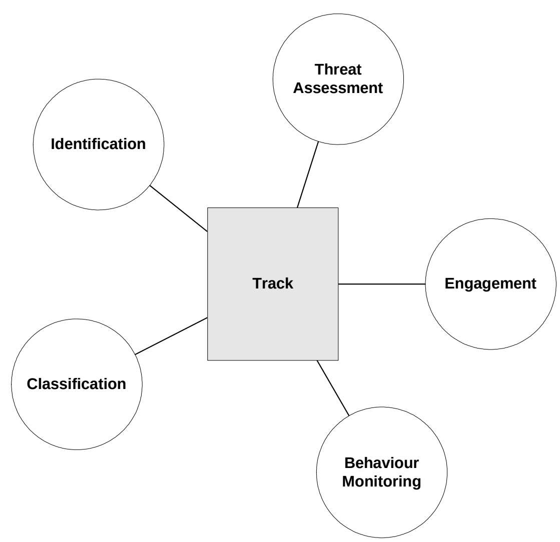

Figure 1 - Tasks (CMS processes) related to tracks.

Returning to the naval domain, one particular class of objects that seems to be a prime candidate of interest to the crew is that of *the platforms present in the tactically relevant surroundings of the own ship*. It is usually reasonable clear where tactical relevancy begins and ends. Such relevancy has to do with sensor and weapon ranges, possible platform velocities and so on. Because all external platforms are represented by tracks in the Combat Management System (CMS), an alternative definition of this particular class of interesting objects is that of *the tracks present in the ship's tactical database*. Tracks are not the only objects that are of interest in the area of C2. More abstract things like military formations and tactics, underlying strategies, etc., can also play a role. Tracks, however, seem to be the objects that are most central to the work in the command centre. The tasks that we associate with tracks in terms of combat management include classification, identification, threat assessment and engagement, supplemented by behavioral monitoring (see figure 1). Because tracks are the primary focus of attention, *adaptive automation should concern itself with tracks first, with tasks taking second place*.

SHARING THE WORKLOAD

A first stab at the notion of workload sharing would be to delegate certain tracks (of a certain type, identity, or something similar) to the system, let the user handle a number of other (uncertain, threatening) tracks and let user and system look together at the rest (compare figure 2). For example, let the system handle all neutral

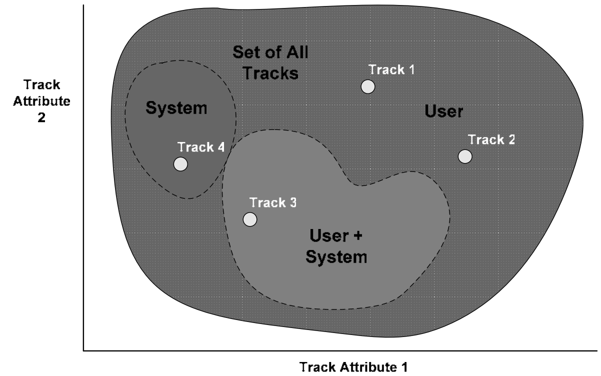

Figure 2 - A first, rough division of tracks between system and user.

tracks, the user all hostile tracks and let the remainder be handled by the system/user combination.

LEVELS OF AUTOMATION

The previous paragraph states that adaptive automation should not be centered on tasks but rather on objects and the processing associated with each (see figure 3Figure 3).

The concept of sets of tracks that are either under the supervision of the user or the system or both is extended to *five* sets (table 1 and figure 3), each with a distinct level of automation. Tracks can move from one set to another if their attributes change (causing them to conform to another set's constraints) or if the set conditions themselves are changed (see figure 5). One or more sets can be empty. In this way, workload can shift from user to system and vice versa, assuming that

Level of Automation	Description	Comment	Signaling
MANUAL	No automation is available or allowed to assist the user.	Necessary if automation is not technically or ergonomically possible.	no
ADVICE	The user keeps all responsibility. The system view is available for advice but the system takes no initiative whatsoever (to alert the user).	Weak form of automation ("pull" only); if the user does not inspect the system space regularly, he may miss important data.	no
CONSENT	The user keeps all responsibility but the system alerts the user to changes in the tactical situation.	Stronger form of automation ("push"); the system advises a copy from system to user space with copying taking place only after consent by user.	yes
VETO	The user delegates the responsibility to the system *unless overruled* (vetoed) by user.	Applicable in those 'fully automated' cases where risks of wrong interpretation by the system are large or unacceptable.	yes
SYSTEM	The user delegates all responsibility to the system and there is no interaction between the system and the operator.	Only acceptable in cases of low risk, e.g., with neutral contacts.	no

Table 1 - Levels of automation.

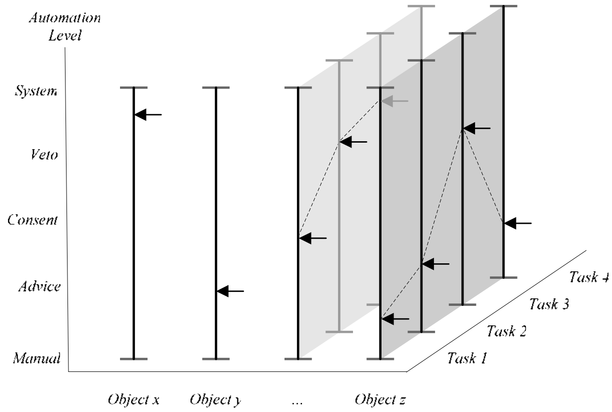

Figure 3 - Instead of applying the concept of adaptive automation to tasks it is applied to objects and object-related tasks at the same time. The arrows per object (track) indicate the level of system control for each task (the top level indicates a high level of system autonomy).

the higher levels of automation imply a lower workload for the user. The sets do not intersect (by definition) and the union of the sets is the entire track set.

We have opted for five automation levels because we are able to make a clear distinction between these levels in terms of *responsibility* and *signaling* (see table 1). Responsibility is defined in terms of whom has the final say with respect to the task associated with the data. More specifically, if the system is allowed to change a track identity, it has responsibility for the identification process, and if the system is allowed to add an engagement to a track, it has responsibility for the ensuing engagement (i.e., the actual firing of missiles).

Signaling happens when the system wants to inform the user of the fact that there is a discrepancy between the user's view of the object and the system's view of the object, or when a decision has been made *by the system* that can still be revoked (vetoed) by the user.

TASK REALLOCATION

In order to be able to shift the workload between user and system, the boundaries of the automation sets are made *adjustable* (figure 4) If the boundaries of the system-controlled sets stretch outward (in this way including more tracks), the workload on the user is automatically reduced, because less tracks are his immediate responsibility. The boundaries are changed by changing the attribute values that define the sets themselves. For example, by reducing the range at which tracks are identified manually, the set of tracks handled by the system (either by consent, by veto or fully autonomous) is increased and the workload of the user is reduced in proportion. Because tracks usually 'move' in their state space, there are two ways in which a track comes under a higher level of system control. Either the track attributes change in such a way that the track starts

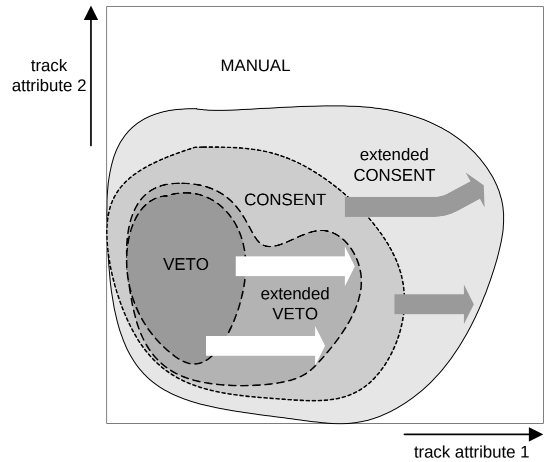

Figure 4 - Adaptable track sets related to a certain task. In this example, the track sets are defined using two attributes (for example, ID and range) and only VETO and CONSENT are shown. The SYSTEM and ADVICE sets are empty and the MANUAL set effectively is what remains from the full track set after the other sets have been subtracted.

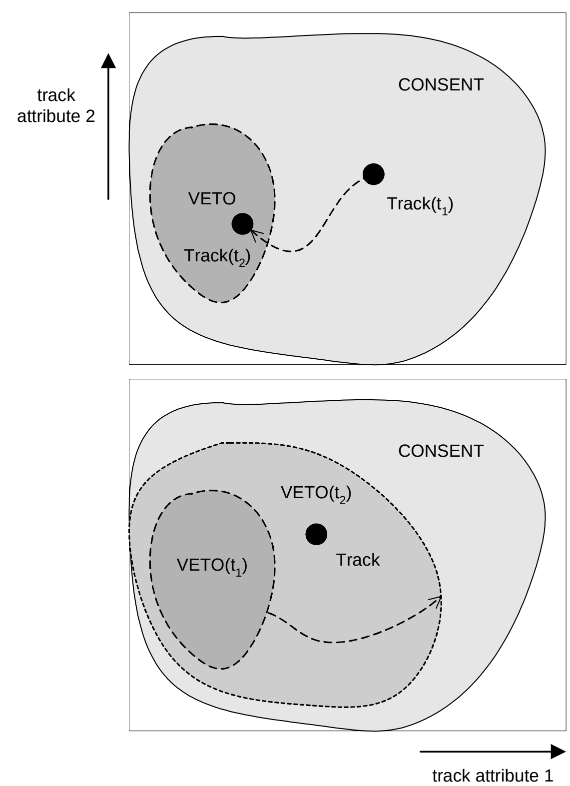

Figure 5.- Adaptable track sets related to a certain task. Promotion of a track to another level of automation is due to a change in the track's attributes (above) or a change in the set definition (bottom).

adhering to another set's conditions, or the conditions of a set change in such a way that a track suddenly adheres to these new conditions (see Figure 5). It is the latter possibility of change that makes the system adaptable. The former change is a result from a predefined, static division of work. Track attributes themselves may change gradually (for example, range) or in discrete steps (for example, identity).

WORKING AGREEMENTS

In the vein of teamwork, several people have elaborated the concept of *working agreements* (see, for instance, Cannon-Bowers & Salas, 1998, Rasker & Willeboordse, 2001). Working agreements should not be regarded as procedures but rather as a method to make *implicit* assumptions *explicit*. Working agreements aid team members in providing each other with information in time, in jointly solving problems, and in assisting each other during periods of large workloads. Navy officials emphasize that working agreements can reduce communication during work shifts. Cannon-Bowers & Salas (1998) used the term *implicit coordination* to describe the same desired behavior. They showed that communication is liable to deteriorate due to the effects of stress. On the other hand, communication is crucial when a team tries to achieve a common objective, and they therefore introduce several training methods to achieve implicit coordination. Working agreements depend on the experience and competence of the operators and are therefore specific to each team and to every situation. In a working agreement, two or more team members agree beforehand on the allocation of tasks, methods of communication and cooperation. These elements are all important for good team work (Rasker & Willeboordse, 2001; Cannon-Bowers & Salas 1998). In the framework of this paper we can view the operator and the system as a team and the arrangement of a *configuration as* a *working agreement*. A configuration determines, based on attribute values, the object-set definitions for each automation level (see figure 6). When the working conditions are normal, the operator works in the *minimal* configuration but when things get busy, the subsequent configuration is applied. The object oriented approach facilitates the usage of tangible objects and attributes that fit the way of thinking of the operator thereby avoiding cognitive reorientation. Attributes like range, identity, and velocity correspond to the way of thinking.

The operator and the system have a common goal and they attempt to cooperate to achieve this goal. The ideas of Rasker & Willeboordse (2001) thus seem to apply to human-machine cooperation as well and in order to reduce the risk of wrong implicit assumptions the operator and the machine should make such agreements explicit.

We furthermore propose a working method where the operator evaluates these working agreements during a debriefing session following a work shift (see Figure 7). During the debriefing session incomplete or cumbersome agreements can be adjusted based on the operator's most recent experience working with the system. These debriefing sessions are used by the operator to improve the working agreements with the system through several iterations. Specifically, the operator makes working agreements with the system with regard to the adaptive automation. Based on predefined tasks and object attributes the operator will instruct the system when and how to increase authority.

WHEN TO TRIGGER

The basis for the adaptive automation argument is the inherent trade-off between the workload and the user's decreased situational awareness that results as the level of automation is increased.

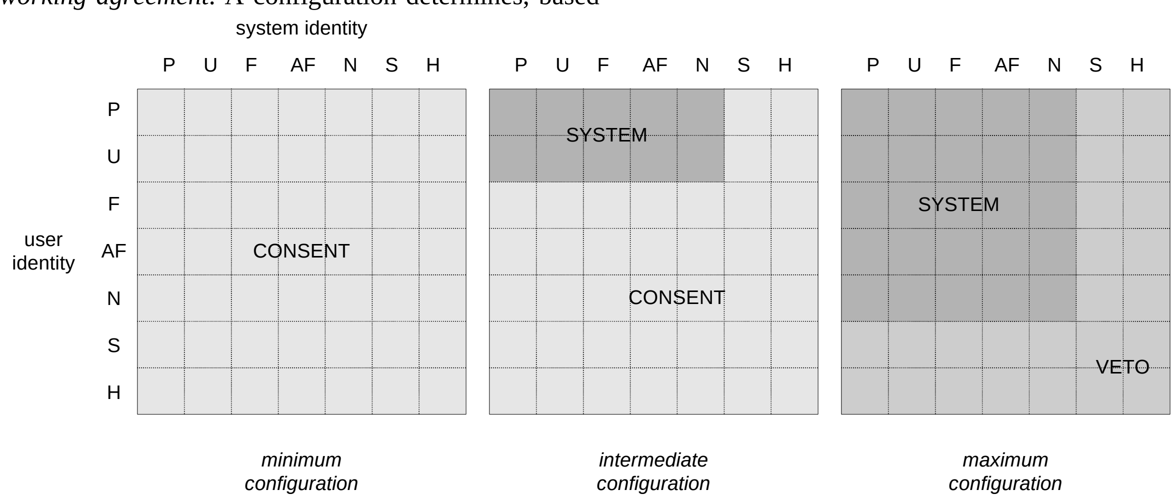

Figure 6 - The correspondence between working agreements and min-max settings for the automation level. F = FRIENDLY, AF = ASSUMED FRIENDLY, N = NEUTRAL, S = SUSPECT and H = HOSTILE.

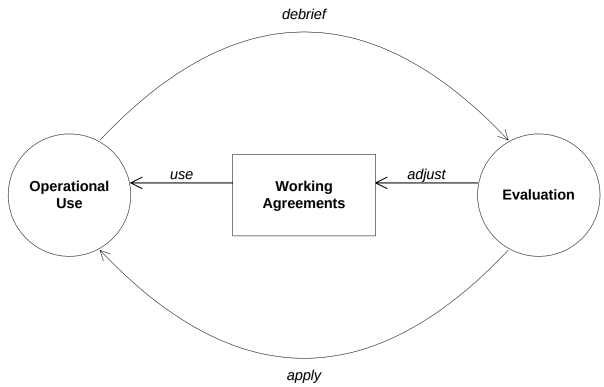

Figure 7 - The operator should work with the system under several working agreements. During debriefing sessions the agreements can get evaluated.

Assuming that automation should be kept at lower levels (to preserve situational awareness) unless high workload precludes effective human performance, adaptive automation will optimize the contribution of both human and machine in an environment where the workload is varying (Wickens & Hollands, 2000). One of the challenging factors in the development of a successful adaptive automation concept concerns the question of *when* changes in level of automation must be effectuated. 'Workload' generally is the key concept to invoke such a change of authority but this concept is used in the broadest sense only.

Workload

Most researchers have come to the conclusion that "*workload is a multidimensional, 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.*" (Gopher & Donchin, 1986). Mental workload can be defined 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*" (Kantowitz, 1987). This definition highlights the two main features of workload, which are the *capacity* of operators 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 are affected by many factors. Factors that increase the capacity are for example skills and education. Factors that reduce the capacity are stressors. For example the capacity decreases (and hence the workload increases) when the operator becomes fatigued or has to work under conditions of high vibrations, at lot of noise or high or low temperatures.

The measurement of workload is again much debated and we would like to elucidate the relationship between workload, task demands and performance. Figure 8 shows the relationship between these three variables.

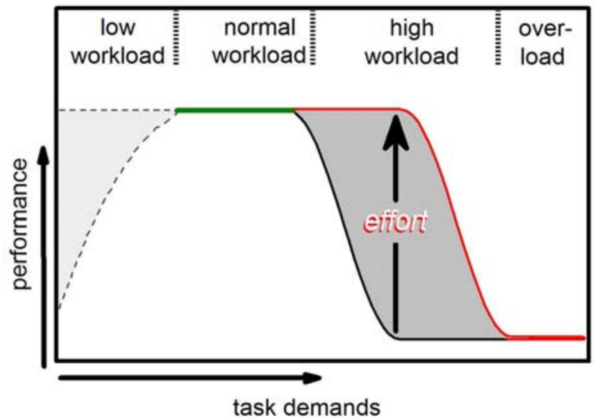

Figure 8 - The relation between task demands, performance and workload (from Veltman & Jansen, 2006).

It shows that an operator can experience different levels of workload dependent on the demands of a task. It also shows that the performance does not necessarily decline as the operator experiences a high workload. It makes sense that one has to work harder (increased workload) when task demands rise and that the performance does not have to decline immediately. Human beings have survived in changing circumstances by being adaptable systems. We can cope with changing conditions without decreasing our performance or getting into an overloaded state by putting more energy into it. The difference between maintaining the level of performance and an increased workload is referred to as the *effort* of the operator. The problem is, of course, that we cannot sustain this effort for a prolonged time.

If measuring workload directly does not look like the best way to trigger the adaptive automation mechanisms other ways must be found. There seem to be three possible methods to initiate adaptive automation: measurements of operator performance, physiological measurements and mission indicators. The *operator's own performance* could be used as a trigger by monitoring the operator's interactions with the system interface and by evaluating against a model to determine when to change levels of automation (see for example Rouse, Geddes & Curry, 1988). Sidestepping the discussion on how to create a performance model, this candidate should measure the performance over time and spot any performance degradation to indicate a decline of operator performance. Second, *physiological data* from the operator are used in various studies as well (Pope, Comstock, Bartolome, Bogart & Burdette, 1994; Byrne & Parasuraman, 1996; Veltman & Jansen, 2006). There are two reasons why physiological measures are difficult to use in isolation, however. 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 only. Second, it is quite possible that some physiological data contradict the current operator state. Various experiments have shown that whereas one measurement indicated an increase in the workload, another measurement contradicted the increase. One therefore needs at least several measurements (physiological or otherwise) to get rid of this ambiguity. Another candidate to vary the levels of automation is to use the flow of the mission itself. Here, the occurrence of *critical events* is used to change to a new level of automation. Critical events are defined as incidents that could endanger the goals of the mission. Scerbo (1996) 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.

Although all three techniques to adapt authority have proven successful in experimental environments and have their pros and cons at first we use performance modelling to invoke adaptive automation for two reasons. First, it is relatively easy and not invasive to measure performance. Second, operator effort is taken into account automatically. This does not mean that the two other techniques cannot be used. We think the techniques can be applied together and in combination they can assist resolving any ambiguity. For example, Ferrez & Millan (2005) use physiological (EEG) data to detect errors in man-machine interaction and one can imagine that similar information can aid a performance model. Physiological data can be used in another interesting way. When one expects an increase in physiological indicators due to increased task demands and this increase is not reflected in the physiological data, one could draw the alternative conclusion with respect to the state of the operator (i.e., he could be presumed to be still in command of the situation).

In view of the above considerations, we propose that an adaptive allocation scheme be based on performance modelling. To this end the system should monitor the responsiveness of the operator and over time build a model of the operator based on the average response times to system alerts. To this purpose the operator performance model measures the time between the generation of a tactical signal and the operator acknowledgement. Using on-line statistical analysis, the average and the standard deviation can be calculated and later deviations can be detected. In a situation where an increasing number of objects (tracks) becomes the responsibility of the operator, he or she will experience an increased workload but with an increased *effort* (cf. Figure 8) the officer will be able to manage the extra objects. As indicated in this paper, an operator manages such an increase in workload only a certain amount of time. If the situation remains 'busy' for a prolonged time, the operator will find it more and more difficult to exert the necessary effort, a situation that will eventually result in a loss of performance. Once this decline in the operator performance is detected, the system will start to shift objects to its own sphere of responsibility. The operator has agreed beforehand (using the working agreements) which objects will be managed henceforth by the system. After the shift, less objects need to be handled by the operator, in turn resulting in a better performance.

CONCLUSION

This study uses an object-oriented approach to implement adaptive automation in the CMS. This approach facilitates a cleaner division between the workload of an operator and the system. It is centered on the *tracks* that are the objects of primary interest in a CMS and the processing associated with each of them, rather than on tasks. The tracks are parceled out per process among five sets, each of which has a distinct level of automation. The five levels of automation we recognize run from MANUAL via ADVICE, CONSENT and VETO to SYSTEM. Each of these automation levels is clearly distinguishable by the level of autonomy of the system and the desired signaling by the system. As the level of automation increases, the authority of the system to make decisions increases as well. At the lower levels (ADVICE, CONSENT), the operator decides, after possible inspection of the system's advice. At the higher levels (VETO, SYSTEM), the system will autonomously act, with a possible retraction of the decision by the operator.

Tracks can move from one set to another if their attributes change (causing them to conform to the constraints of another set) or if the set conditions themselves are changed. In this way, by moving tracks to sets with another level of automation, work is shifted from the operator to the system and vice versa, assuming of course that the higher levels of automation imply a lower workload for the operator. For example, by reducing the range at which tracks are to be identified manually, the set of tracks handled by the system is increased and the workload of the operator is reduced in proportion. Hence, by making the set conditions adjustable, adaptive behavior is introduced.

The adjustment of the set boundaries is triggered by a decrease of the operator performance over time *as* *perceived by the system*. Although the operator is notified of the adjustment, no further operator interaction is instigated, because such an interaction would be counterproductive, momentarily adding to the workload of the user and causing a further decline in operator performance. The perception of the operator's increased workload is in our opinion best based on models of the operator's performance, because such a model facilitates pro-active behavior. Not much deliberation between operator and system is necessary anyway because the adjustments are delineated in advance in the form of a working agreement between operator and system. In these working agreements the operator stipulates to what extent and under what circumstances the system is allowed to change its levels of automation.

ACKNOWLEDGMENTS

This research was partly funded by the Royal Netherlands Navy under program number V206.

REFERENCES

Byrne, E. A. & Parasuraman, R. (1996). Psychophysiology and adaptive automation. *Biological Psychology, 42*, 249-268.

Cannon-Bowers, J. A & Salas, E. (1998). *Making decisions under stress*. Washington, DC: American Psychological Association.

Endsley, M., Bolté, B, & Jones, D. (2003). *Designing for situation awareness: an approach to user-centered design*. London: Taylor and Francis.

Ferrez, P. W. & Millan, J. d. R. (2005). You are Wrong! --- Automatic Detection of Interaction Errors from Brain Waves. *Proceedings of the 19th Interational Joint Conference on Artificial Intelligence*.

Gopher, D. & Donchin, E. (1986). Workload -an examination of the concept. In K.R. Boff, L. Kaufman & J.P. Thomas (Eds.), *Handbook of perception and human performance. Volume II, cognitive processes and performance*. (pp 41/1-41/49). New York: Wiley.

Kantowitz, B.H. (1987). Mental workload. In P.A. Hancock (Ed.), *Human Factors Psychology*. (pp 81-121). Amsterdam: North-Holland.

Parasuraman, R., Mouloua, M., & Molloy, R. (1996). Effects of adaptive task allocation on monitoring of automated systems. *Human factors, 38,* 665-679.

Pope, A. T., Comstick, R. J., Bartolome, D. S., Bogart, E. H., & Burdette, D. W. (1994). Biocybernetic system validates index of operator engagement in automated task. In M. Mouloua & R. Parasuraman (Eds.), *Human performance in automated systems: Current research and trends* (pp. 300-306). Hillsdale, NJ: Lawrence Erlbaum.

Rasker, P. C. & Willeboordse, E. J. (2001). *Werkafspraken in de commandocentrale* (Rep. No. TM-01-A002). Soesterberg: TNO Technische Menskunde.

Rouse, W. B., Geddes, N. D., & Curry, R. E. (1987-1988). An architecture for intelligent interfaces: Outline of an approach to supporting operators of complex systems. *Human Computer Interaction, 3,* 87-122.

Rouse, W. B. (1988). Adaptive aiding for Human/Computer Control. *Human factors, 30,* 431-443.

Rumbaugh, J. R., Blaha, M. R., Lorensen, W., Eddy, F., & Premerlani, W. (1991). *Object-Oriented Modeling and Design*. Prentice Hall Inc.

Scerbo, M. (1996). Theoretical perspectives on adaptive automation. In R. Parasuraman & M. Mouloua (Eds.), *Automation and human performance: theory and applications* (pp. 37-63). New Jersey: Lawrence Erlbaum Assiciated, Publishes Mahwah.

Shlaer, S. & Mellor, S. J. (1992). *Object lifecycles: modeling the world in states*. Englewood Cliffs, NJ: Yourdon Press.

Veltman, J.A., Jansen, C. (2004). The adaptive operator. *Proceedings of the Human Performance, Situation Awareness and Automatoin Technology Conference 2004*.

Wickens, C & Hollands, J. (2000). *Engineering psychology and human performance*. (Third ed.) Upper Saddle River, New Jersey: Prentice-Hall Inc.