Grootjen, M; Neerincx, M.A., "Operator load management during task execution in process control", *Human Factors Impact on Ship Design*, Genoa, 2005.

# Operator load management during task execution in process control

M. Grootjen. (Marc@Grootjen.nl) Royal Netherlands Navy/TNO Human Factors, Kampweg 5, P.O. Box 23, 3769 ZG Soesterberg, the Netherlands.

M. A. Neerincx. TNO Human Factors/Technical University of Delft, Kampweg 5, P.O. Box 23, 3769 ZG Soesterberg, the Netherlands

Keywords: *mental load, task analysis, human-computer interaction, cognitive engineering, dynamic allocation, adaptive interface*

The ever increasing automation and manning reductions in process control result in enormous fluctuations of the operators cognitive task load. A possible solution in keeping operator load and performance at an adequate level over time, is adaptive support. Our paper combines recently developed support functions into a new adaptive support framework. It shows which information of operator, task, system en context models can be used as input for the dynamic task allocator, which generates an optimal work plan. This work plan is carried out with use of dynamic task allocation and an adaptive interface. Finally, based on design rules and an evaluated interface concept, an adaptive interface prototype is presented and discussed.

# 1. INTRODUCTION

In the last 2 decades, major changes in information technology have taken place. In process control, the ongoing automation and the application of new technologies caused a radical change in the position of the operator. However, the development of new support systems couldn't keep up with rest of the changes in information technology. This section first summarizes the problems in process control and then points out the focus of the current research how to deal with these problems.

## 1.1 Problems

Analyzing the problems in process control during multiple seminars and projects, we identified some main topics:

- Information type. The type of information an operator has to handle changed. For example from analog to digital, and from sensor value to information on a 'higher level'. Especially when different types of information are used together, problems can arise.
- Information volume. The number of used sensors, and the accuracy of the sensors, has increased enormously in the last decades. Because of this, the information volume offered to the user has grown exponentially.
- Information volume fluctuation. Nowadays, highly dynamic systems cause extreme fluctuations in the volume of information produced.
- Task integration. The integration of tasks from different domains (e.g. navigation and propulsion) increases, demanding more flexibility of human operator and system.
- Increasing autonomy. Autonomous systems may cause 'out of the loop' problems for an operator.
- Increasing complexity. Systems and their dependencies become more and more complex. Consequently the problems that the operator has to solve increase in complexity.
- Low personnel costs. The pressure to work with the lowest possible costs and the highest possible efficiency of the overall man-machine system after implementation asks for an optimal distribution of personnel.
- Low training costs. The minimization of the training costs asks for intuitive systems.
- Increasing legislative constraints. The increasing number of laws (e.g. to protect the environment & employees [2]), in combination with the pressure to maintain maximum operational capability, put extra constraints on the system design.

## 1.2 Research focus

To deal with the mentioned problems, we think the operator needs specifically personalized support which can differ in time [1]. A strong need for a new, innovative framework, which can be used to design adaptive user friendly systems from the operators point of view, urges. Such a system should accommodates the operator with the right task support at the right time. This paper is part of a Ph.D. research, called “Adaptive Interfaces for Operational Support”, which has as main goal the development and testing of such a framework in the maritime domain.

Recently, a method for the design of such user friendly systems has been developed [3, 4]. The method results in interface concepts which are of great value and increase efficiency and effectiveness. The method aims at an optimal Cognitive Task Load (CTL) for the operators at all time, but, until now, it was evaluated for static function allocation and interface design. Static function allocation refers to the process of distributing tasks and functions at design time, by analyzing both system and operator competences. However, CTL is a dynamic variable which varies enormously over time. If we want to keep the CTL at an optimum at all time, we need a real time tool to influence CTL, in other words, we want to develop a new adaptive support application, which uses *dynamic task allocation*. With dynamic task allocation a task, or part of a task, can be transferred to another actor (i.e. automated component or operator). Obviously, this has consequences for the interface, which is dynamically altered as well.

This paper first describes an initial high level framework for adaptive support and the role of dynamic task allocation. Section 3 shows possible consequences for the interface. Finally section 4 will present the conclusion.

## 2. FRAMEWORK FOR ADAPTIVE SUPPORT

Our goal, to develop a user friendly, efficient and effective system which adapts to the situation and the operator, accommodating the user with the right task support at the right time, evokes an important question: How and when should support be adjusted? To answer this question, a lot of information is needed. Furthermore, we need a mechanism that transforms this information into a plan which describes who or what performs each action. We define this plan here as the *work plan*, the mechanism that generates this work plan the *dynamic task allocator* or shortly *allocator*, the sources of information are *models* and the mechanisms that produce the models by filtering and collecting information from the environment are *modeling components*.

First step in the design of the our framework is the identification of the models we need. Literature shows a large variety of models, for example Schneider-Hufschmidt et. al [3] describe a structural model of the interface composed of about 13 elementary models which focus on the various functional aspects relevant to adaptive interaction. Of course, if the amount of models is increased, the information will be more precise and approach the real situation more and more. However the complexity of the system will also increase enormously. If we want to keep the system controllable, we should minimize the number of models. Based on our goal, we need at least the following models:

1. task model

2. operator model (or user model)
3. system model
4. context model (or domain model)

Figure 1 shows our framework for adaptive support. Modeling components use information from the environment to produce the four models. The models are used by the allocator, which generates an effective and efficient work plan. Models and allocator are explained below. The work plan tells when and how to adapt the system, it describes the dynamic task allocation between the actors.

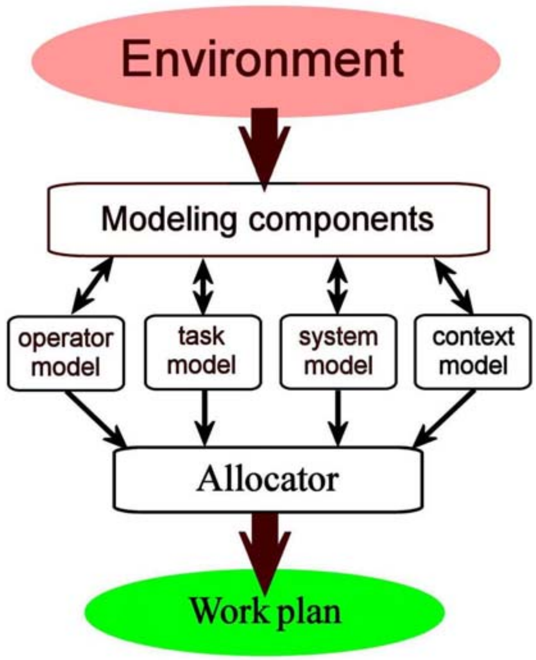


Figure 1: Initial framework for dynamic task allocation.

Operator model. An essential source of information for the allocator is the operator model. In order to adapt the system to the characteristics of the operator, the operator model can contain a large variety of information:

- Real-time CTL measures (TO, TSS and LIP, see allocator)
- Profile parameters; routine on the job [8], spatial abilities, memory span [1], education & characteristic knowledge, preferences, generic role in environment.
- Physiological measures; heart rate & variability, respiratory frequency & amplitude, eye blink frequency & duration, thermacam to measure face warmth.
- Performance; reaction time, time pressure, correct handling.
- Subjective effort; workload watch, continuous memory task
- Behavioral variables; Eye tracker, gaze tracker.

Task model. The task model should contain information about the possible tasks for an operator. It should be noted that this information represents task demands that affect human operator performance and effort (i.e. it is not a definition of the operator cognitive state). We see the task model as some kind of static representation expressing all possible tasks that can be executed. The *effects* these tasks have on the operator belong to the operator model. The CTL method was developed to analyze the CTL for specific, concrete task contexts. With use of this method and a description format, action sequences can be created and assessed. So, when a task description is available, the accompanying CTL can be calculated. For more explanation about CTL see allocator.

System model. In order to support a user, a system needs to have information about itself available. Consequently, the system model contains technical information about the different system components (e.g. layout, software applications and dependencies). Special cases of system models are for example application models and dialogue (interaction) models. The application model consists of information how the application can be adapted, or can adapt itself. The dialogue model is very important in adaptive support [9]. It describes the dialogue between the system and the user, the interactive behavior of an user interface: how information has to be shown to the user in an understandable way. The model must contain rules describing what changes can be made to the system.

Context model. The context model (or domain model) contains relatively static, high level information of the operational unit and its environment (e.g. the readiness state of a combat unit or the production quota of a chemical plant).

Allocator. Neerincx [4] developed a CTL model which seem very useful for the allocator. The CTL model (Figure 2) distinguishes three load factors that have a substantial effect on task performance and mental effort. The first load factor is the classical factor percentage time occupied (%TO). The second load factor is the level of information processing (LIP) (cf. the skill-rule-knowledge framework of Rasmussen [5]). To address the demands of attention shifts, the cognitive load model distinguishes task-set switching (TSS) as a third load factor.

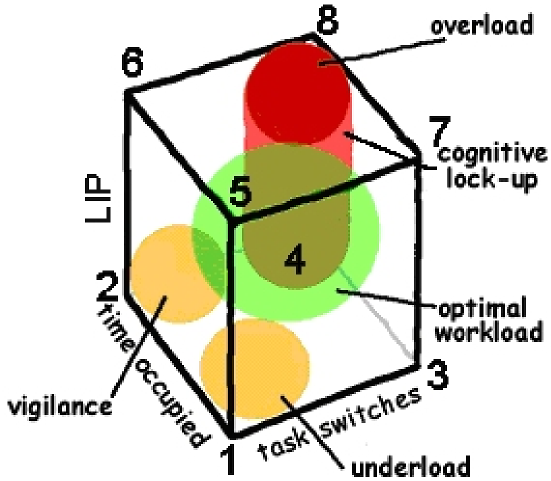


Figure 2: Dimensions of the CTL model ([4], see text for explanation).

Figure 2 presents a 3-dimensional 'load' space in which human activities can be projected with regions indicating the cognitive demands that the activity imposes on the operator. In the middle area, CTL matches the operator's mental capacity. At angular point 8 CTL is high and an overload situation occurs. Angular point 1 represents the area in which CTL is not optimal due to underload. When TO is high, and LIP and TSS are low, vigilance problems can appear (angular point 2) [6] When TO and TSS are high, lock-up can appear [7].
With the information of the different models, the allocator can generate 'CTL space' and determine the current and future working point for a specific operator. With this information an work plan can be generated.

## 3. THE ADAPTIVE INTERFACE

Oppermann [10] states that the user interface is the part of a system responsible for getting input form the user and for presenting system output to the user. A system that adapts either of these functions to the user's task or to the characteristics or preferences of the user, is an adaptive interface. This section shows how an interface in naval ship control dynamically changes due to application of the adaptation strategy of section 2.
After validation of the CTL model, [11] developed four support functions and an interface that influence CTL specifically on the three load factors (Table 1).

| Support function | Supports on | Support concept |
|---|---|---|
| Information handler | TO | Combining and structuring information |
| Rule Provider | LIP | Providing normative procedures |
| Diagnosis Guide | | Guidance of diagnostic processes |
| Task Scheduler | TSS | Providing an overall work plan |

Table 1: Four support functions, the accompanying load factor and their support concepts.

For each concept, several instances can be distinguished (see also Figure 3):

1. Combining and structuring information
   - Ordering alarms in categories
   - Process-based (automatic) presentation of required interface component
   - Hyperlinks within and between the components
2. Providing normative procedures
   - Context specific procedural information
   - Spatial advice (graphical presentation of ‘routing rules’)
3. Guidance of diagnostic processes
   - Help with diagnostic process
4. Providing an overall work plan
   - Task overview
   - Check mark ability (process state)
   - Prioritising alarms

Until now, these concepts and instances were extensively tested [12, 13] without the application of the dynamic support framework of section 2. They served as basis for the current research. Following the design principles for adaptive interfaces [14], a new adaptive support interface was developed shown in Figure 3. In addition to the support functions of Table 1, the adaptive user interface:

- Shows only the groups with active alarms (‘fire’, ‘navigation’, ‘sitrep’ (= situational report)). ‘Empty’ groups disappear from the interface.
- Contains only the buttons that are relevant for the alarms the operator handles (in this case ‘fire fighting’, ‘navigation’ and ‘communication’). All information is always available under the ‘database button’ on the right.
- Provides the operator with the possibility to redirect the alarm to another operator, or to accept the alarm and choose a certain level of task allocation between the operator, computer and/or other operators. This level is represented by the icon behind each alarm

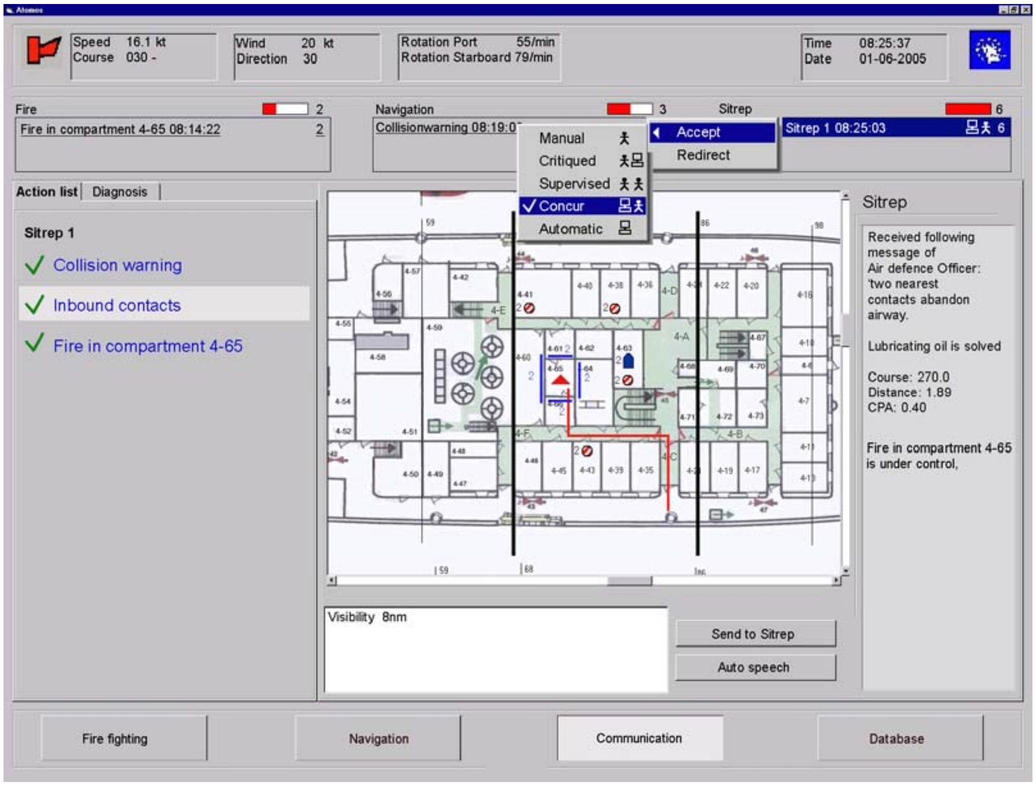


Figure 3: The adaptive interface: a new alarm (group) appears. The operator accepts the alarm in concur mode.

In Figure 3 a new alarm in a new group appears: sitrep 1 in the sitrep group. The operator selects the alarm, and decides to accept it. He can choose 5 support modes (see also [15]):

1. Manual. In this mode the operator gets an alarm that it's time to give a sitrep. When he selects the alarm a procedure is given, presented in the procedure presentation area. Clicking on a procedure step shows the accompanying information in the middle of the interface. From here information can be selected and put into the sitrep. No further advise is given.
2. Critiqued. See manual mode, but now the system criticizes the operator in case of faults. The operator still has to collect the information himself.
3. Supervised. The tasks are executed as in the manual mode, but a human supervisor checks the operator.
4. Concur (shown in Figure 3), See critiqued, and in this mode all information will be selected by the system. The operator is able to change the information, and can choose to do the sitrep himself, or let the system do it (button: autospeech). In the concur mode the procedure steps are already checked and performed by the system. However, it is still possible to return to these procedure steps and change the information in the sitrep.
5. Automatic. See concur, only in this highest automation mode, returning to the procedure steps is not possible anymore. The operator has to press

the 'autospeech' button after which the system does the sitrep. If the operator waits too long with this command, the system will automatically do the sitrep.

## 4. CONCLUSION

The increase in automation in process control emerged a strong need for dynamic adaptation tools. To generate a work plan which describes the dynamic task allocation between operators and automation, this paper presents a high level framework for adaptive support. In this framework, four information models have to be implemented: the operator, task, system and context model. With the information of this models, the allocator generates a work plan which aims at an optimal CTL and performance. Individual parts of this dynamic function allocation network were tested in [4] (CTL model and method), [12] (interface support), [15] (dynamic task allocation), [14, 16] (operator model). Findings are combined and used in the development of an adaptive user interface. Experiments on trust, situational awareness and performance with this interface will be conducted at the end of 2005.

In the (theoretical) consideration of this paper, at least two important points should be taken into account:

Adaptation versus Self-Adaptation. Who should make the decision for dynamic task allocation is an important question we didn't address in this paper. The decision can be made by automation, the operator or by a mix of both. Even another operator (e.g. a superior) can be involved and make the decision. Schneider-Hufschmidt et. al [3] describes 4 different tasks that have to be performed in the adaptation process from the user's point of view. Each tasks can be performed by the system, or by a user, so 8 different configurations arise. The configuration where the computer takes all actions is defined as 'self-adaptation', the one where the operator takes all actions is called 'adaptation'. van der Kruit [15] defines only two categories, 'adaptable' and 'adaptive' automation. The support example of Section 3 could be identified as adaptable (according to [15]), or as computer-aided adaptation (according [3]). Other possibilities are currently researched.

Modeling components. Which information of the models will be used by the modeling components has to be determined separately for every domain. For the command and control domain of navy ships, experiments were conducted by Neerincx et. al [14] and Veltman & Jansen [16]. For example eye blink frequency appeared to be an important variable for the operator model.

## REFERENCES


1. Neerincx, M.A., Lindenberg, J., and Grootjen, M.: “Accessibility on the Job: Cognitive Capacity Driven Personalization”, *Proceedings HCII*, (2005).
2. Teltzrow, M. and Kobsa, A.: “Impact of User Privacy Preferences on Personalized Systems - a Comparative Study”, Dordrecht, (2004).
3. Schneider-Hufschmidt, M., Kühme, T., and Malinowski, U.: “Adaptive User Interfaces - Principles and Practice”, Elsevier, Amsterdam, (1993).
4. Neerincx, M.A.: “Cognitive task load design: model, methods and examples”, *Handbook of Cognitive Task Design*, Hollnagel, E. (eds), Mahwah, NJ., (2003).
5. Rasmussen, J.: “Information processing and human-machine interaction: An approach to cognitive engineering”, Elsevier, Amsterdam, (1986).
6. Parasuraman, R.: “Vigilance, monitoring, and search”, *Handbook of perception and human performance, volume 2: cognitive processes and performance*, New York, (1986).
7. Boehne, D.M. and Paese, P.W.: “Deciding whether to complete or terminate an unfinished project: A strong test of the project completion hypothesis”, *Organizational Behavior and Human Decision Processes 2, 178-194*, (2000).
8. Benyon,D. and Murray, D.: “Experience with adaptive interfaces”, *The Computer Journal,* (1988).
9. Brown, D., Totterdell, P., and Nonnan, M.: “Adaptive user interfaces”, Academic Press, London (1990).
10. Oppermann, R.: “Adaptive User Support - Ergonomic Design of Manually and Automatically Adaptable Software”, Lawrence Erlbaum Associates, Hillsdale, New Jersey, (1994).
11. Neerincx, M.A. and Lindenberg, J.: “DISCII: Design of interface support for task-set switching and integration”, TM-00-D006, TNO Human Factors Research Institute, (2000).
12. Grootjen, M., Neerincx, M.A., and Passenier, P.O.: “Cognitive task load and support on a ship's bridge: Design and evaluation of a prototype user interface”, *Proceedings INEC 2002, p198*, (2002).
13. Neerincx, M.A. and van Besouw, N.J.P.: “Cognitive task load: a function of time occupied, level of information processing and task-set switches”, *Industrial Ergonomics, HCI, and Applied Cognitive Psychology six[31], 247-254*, (2001).
14. Neerincx, M.A., Streefkerk, J.W., and Lindenberg, J.: “Adaptive User Interfaces on Naval Ships”, TNO Human Factors Research Institute, (in progress).
15. van der Kruit, V.L.I.: “Adaptive automation in command and control decision making”, Technical University of Delft, TNO HF, (2004).
16. Veltman, J.A. and Jansen, C.: “The role of operator state assessment in adaptive automation”, TNO Human Factors Research Institute, (in progress).