Neerincx, M.A., Dobbelsteen, G.J.H. van den, Grootjen, M. & Veenendaal, J. van (2003). Assessing cognitive load distributions for envisioned task allocations and support functions. *Thirteenth International Ship Control Systems Symposium (SCSS)*, Orlando, Florida, 7-9 April 2003.

ASSESSING COGNITIVE LOAD DISTRIBUTIONS FOR ENVISIONED TASK ALLOCATIONS AND SUPPORT FUNCTIONS

Mark A. Neerincx, Gerard J.H. van den Dobbelsteen, TNO Human Factors, The Netherlands
Marc Grootjen, Johan van Veenendaal, Royal Netherlands Navy, The Netherlands

ABSTRACT

Information technology provides new possibilities to combine and support operator tasks in order to improve the operational effectiveness and efficiency on board of a ship. The general question is how to design a task allocation and support system that lead to an adequate deployment of the human cognitive resources. Neerincx [1] developed a Cognitive Task Load (CTL) analysis method to address this question systematically during (re)design processes. A brief example application shows the proposed approach: an analysis of different combinations of platform-supervision and navigation tasks with various support functions on the ship's bridge of the Royal Netherlands Navy, aiming at an unmanned ship control centre under non-critical situations. Subsequently, we present a first prototype tool for the proposed, systematic exploration of the "design space" by assessing the operator load for different task allocations and support functions. For envisioned scenarios, the analyst specifies several levels of crew experience, task allocations and support functions, and the simulator subsequently calculates the corresponding load distributions among the crew (including possible occurrences of momentary peak values) and the overall task execution time of the crew.

KEY WORDS

Mental load, task allocation, human-computer interaction, system development, task analysis, and simulation.

1. INTRODUCTION

High operational requirements, such as safety, and economic constraints, such as human resource' costs, have a major impact on the development of complex human-machine systems. Automation will lead to manning reductions, so that fewer personnel will have to manage high-demand situations and supervise complex automated systems in future ships of the Royal Netherlands Navy (RNlN). For example, the RNlN added platform supervision tasks to the navigation tasks on the bridge of an amphibious transport ship, so that the ship control centre can be unmanned under non-critical situations. Addressing Human Factors in the development processes of such complex and dynamic human-machine systems is essential to enhance the operational effectiveness and efficiency. We developed a Cognitive Task Load (CTL) theory and method for analysing (1) task load distributions among control room crew and (2) the effects of different support functions. The CTL theory distinguishes three load factors that affect operator performance and mental effort: the classical measure percentage time occupied, the number of task-set switches and the level of information processing. In addition, the load theory distinguishes four "generic" support functions that affect cognitive load and human performance: information handler, task scheduler, rule provider and diagnosis guide. Recent research provided empirical support for this theory, showing effects of each load factor and support function on performance measures and mental effort. The CTL method focuses on the creation and assessment of scenarios, and has proven to be able to predict Cognitive Task Load satisfactorily for a naval ship control centre. The next step is to develop a tool for effective and efficient application of the method for systematic exploration of the "design space" by assessing the operator load for different task allocations and support functions. This paper summarises the foundation and an example application of the CTL-analysis method, and the first prototype of a CTL simulation tool that is implemented in the Integrated Performance Modelling Environment (IPME).

2. CTL MODEL AND SUPPORT

Neerincx [1] developed a cognitive load model, distinguishing three load factors that have a substantial effect on task performance and mental effort. The first classical load factor, *percentage time occupied,* has been used to assess workload in practice for time-line assessments. Such assessments are often based on the notion that people should not be occupied more than 70 to 80 percent of the total time available. To address the cognitive task demands, the cognitive load model incorporates the Skill-Rule-Knowledge framework of Rasmussen [2] as an indication of the *level of information processing.* At the skill-based level, information is processed automatically resulting into actions that are hardly cognitively demanding. At the rule-based level, input information triggers routine solutions (i.e. procedures with rules of the type ‘if <event/state> then <action>’) resulting into efficient problem solving in terms of required cognitive capacities. At the knowledge-based level, the problem is analysed and solution(s) are planned, in particular to deal with new situations. This type of information processing can involve a high load on the limited capacity of working memory. To address the demands of attention shifts, the cognitive load model distinguishes *task-set switching* as a third load factor. Complex task situations consist of several different tasks, with different goals. These tasks appeal to different sources of human knowledge and capacities and refer to different objects in the environment. We use the term task set to denote the human resources *and* environmental objects with the momentary states, which are involved in the task performance.

The combination of the three load factors determines the cognitive task load: the load is high when the percentage time occupied, the level of information processing and the number of task-set switches are high. Figure 1 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. The middle spot represents the area in which task load matches the operator’s mental capacity in a certain task setting. In the top area task load is too high. The bottom area represents the area in which performance is not optimal due to underload. When the time occupied is high, and level of information processing and number of task-set switches are low, *vigilance* problems can appear [3]. When the time occupied and the number of task-set switches are high, *cognitive lock-up* can appear (i.e., the tendency of people to focus on single faults, ignoring the other subsystems to be controlled [4]).

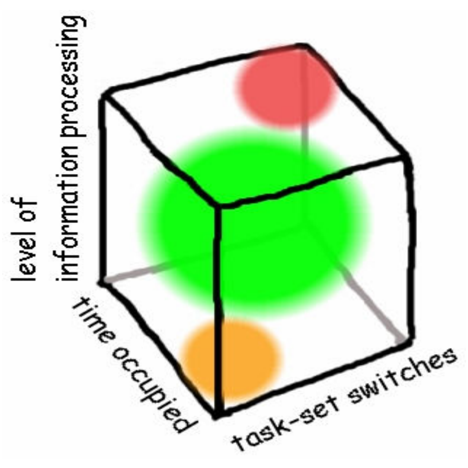

Figure 1. The Cognitive Task Load model.

Based on the theory and our method for cognitive task analysis, we developed 4 support concepts and for each high-level design principles [5] (table 1):

The Information Handler filters and integrates information to improve situation awareness, i.e. knowledge of the state of the system and its environment, and reduces the *time occupied*. Due to the increasing availability of information, situation awareness can deteriorate without support. Correct information should be presented at the right time, at the right abstraction level, and compatible with the human cognitive processing capacity.

The Rule Provider provides normative procedures for solving (a part of) the current problem and affects the *level of information processing*. Due to training and experience, people develop and retain procedures for efficient task performance. Performance deficiencies may arise when the task is performed rarely so that procedures will not be learned or will be forgotten, or when the information does not trigger the corresponding procedure in human memory. For these situations, rule provision aims at supplementing human procedural knowledge.

The Diagnosis Guide affects the *level of information processing.* The level of information processing increases when no complete (executable) procedure is available to deal with the current alarms and situation. This support function guides the operator during the diagnosis resulting in an adequate problem-solving strategy for a specific task.

The Scheduler affects the number of *task-set switches* by providing an overall work plan for emergency handling. Task priorities are dynamically set and shown in a task-overview to the operator resulting in effective and efficient switches.

Table 1: Cognitive load factors and the corresponding support concepts.

Cognitive load factor	**Support concept**
Time occupied	Information Handler
Level of info processing	Rule Provider
	Diagnosis Guide
Task-set switches	Scheduler

3. VALIDATION OF THEORY

We conducted experiments in controlled laboratory settings and in more complex, realistic settings to systematically test the theory and investigate its application in the “real world”. In this research approach, the test environment subsequently increases in complexity and decreases, therefore, in controllability, so that we can test and refine the theory, *and* achieve a good understanding of its applicability in practice. The “alarm 112” experiment used a simple laboratory task in which the user had to do three artificial computer tasks, supervising fire control, ambulance and police crew [6]. The “Ship Control Centre” (SCC) experiment used a test-environment with computer tasks that exhibit important features from damage control on ships explicitly, such as alarm handling, planning and instruction [7]. Both lab experiments showed that “level of information processing” and “task-set switching” can affect operator performance and mental effort substantially, in addition to the classical load measure "time-occupied". Furthermore, the negative effects of the load factors proved to reinforce each other in the lab experiments. Subsequently, we conducted the "SCC simulator" experiment in a realistic ("high-fidelity") SCC simulator of the RNlN's Multi-purpose frigate as a further validation study [8]. Application of the CTL-*method* resulted in the specification of 8 scenarios and crew action sequences from which the three (estimated) load factor values could be derived per operator. Subsequently, 13 crews had to perform these 8 scenarios. The estimated load values proved to correspond well with the actual levels during task performance. In correspondence with the CTL-*model*, the three load factors proved to have a substantial effect on operator performance and effort, showing under- and overload situations.

We applied the four support concepts in the design of user interfaces for several systems. For example, Grootjen et al. [9] designed such a user interface for a ship's bridge. Subsequently, they conducted an experiment to test the effects of the support functions, under high and low task load, on task performance, mental effort and possible side effects (such as operator's loss of situation awareness). In this experiment, 50 RNlN students had to solve damage control problems with the prototype interface. The support proved to result in substantial effectiveness and efficiency profits, i.e. the use of support functions leaded to a substantial improvement of task performance, especially at high task load. Possible costs of being "out of the loop", like not reacting on an implemented wrong advice or a decrease in understanding of performed actions, could not be found.

4. EXAMPLE OF CTL METHOD

Cognitive task load can only be analysed for specific, concrete task contexts. An effective method to create such a context is the use of *scenarios* [10]. Scenarios presuppose a certain *setting*. Within the setting, roles are played by *actors*. In complex scenarios different actors can be involved, possible interacting with each other. Actors have specific *goals or tasks*. To achieve this goal *actions* have to be taken. Neerincx et al. [5] provide a method and description format to systematically create and assess normal *and* critical situations with their corresponding action sequences (figure 2). Such an action sequence displays actions of different actors on a time-line, including the interaction with support systems. The actions can be triggered by events, and are grouped according to their higher-level task (goal). Van Veenendaal [11] assessed the action sequences for alternative designs of the naval ship's bridge, comprising different task allocations and support functions for navigation and platform supervision. Normal and critical *scenarios* were specified with domain experts. Furthermore, for every scenario, support functions were specified and included in the action sequence specifications (i.e. information handler, rule provider, diagnosis guide and task scheduler). The action sequences were validated with domain experts. The *cognitive load* model was used to assess these action sequences, each sequence with and without the four support functions. First, the three load factors were calculated per 6 minutes task performance, showing the dynamic load fluctuations in the 3-dimensional load cube of figure 1. Subsequently, via questionnaires experts assessed the action sequences to acquire subjective load measures and estimations of the support effects.
The analysis showed that the task of the Officer of the Watch could be extended with platform control tasks under normal conditions. The support functions will complement the knowledge and experience of the bridge crew to realise an adequate performance level. In critical situations however, extra, technical personnel has to be called up. This study provided the first indicators for implementing such a dynamic task allocation. It should be noted that this example analysed a limited set of scenarios and design alternatives. To support a more extensive analysis, one needs a software tool that aids with the creation of scenarios and action sequences, and that calculates the consequences of different design alternatives.

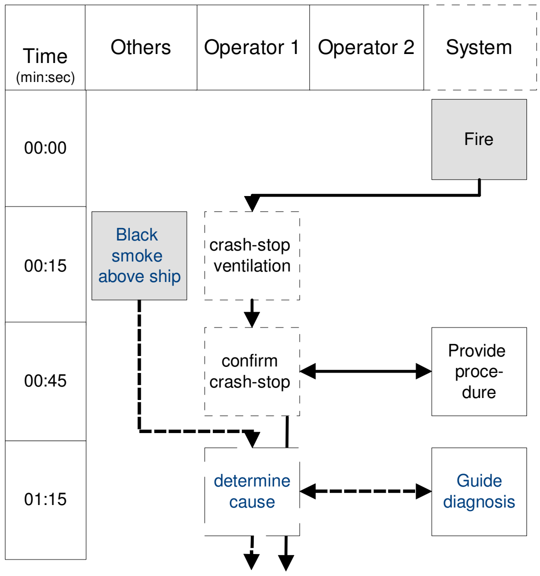

Figure 2: Outline of a Compound Action Sequence (CAS) consisting of 2 Basic Action Sequences (BAS) (i.e. handling of both the fire and the black smoke events).

5. THE CTL SIMULATOR

To support the assessments of task and interaction design proposals, we are developing a CTL simulation tool. Based on the specification of basic action sequences (BAS), the time of event occurrences, the human actor's experience and the support functions, the tool derives the resulting compound action sequence (CAS, figure 2). For this CAS, it calculates the cognitive load per human actor and the overall duration to finish all the actions. Thus, a CAS consists of several Basic Action Sequences (BAS), i.e. the procedures—including the events—to accomplish a goal or task. Each BAS has a relative priority (i.e. dependent on the context, such as Readiness State). When an event of a BAS occurs that has the highest priority, this BAS will be performed first. It should be noted that there could be "wait periods" in a BAS (e.g. waiting for the outcome of an action that has been delegated to another person), in which lower priority action sequences can be active. Each action has a "baseline time" (in seconds), i.e. the time an expert—who has all the required ready knowledge and is in an optimal mental state—needs to perform the action.

The CTL simulator distinguishes 5 Levels of Information Processing (LIP):

5: Knowledge Based (KB) high
4: Knowledge Based (KB) low
3: Rule Based (RB) high
2: Rule Based (RB) low
1: Skill Based (SB)

The LIP is defined by three parameters (Neerincx [12] provides a decision tree that supports the specification of these parameters):

- the experience of the performer the BAS (little, medium or much)
- the learnability of the action (i.e. the duration of the learning process: long, medium or short),
- the type of support that will be used (none, diagnosis guide, rule provider).

Table 2 shows the different values for the LIP in all possible conditions.

In addition to the Rule Provider and Diagnosis Guide, we distinguish an Information Handler function that reduces the action time of the task performer. The CTL simulator does not yet include a Scheduler effect. This leads to the following formula for the action time of a human actor:

Actiontime = IHfactor(LIPfactor*baseline)

Baseline=[secs]
LIPfactor=[1, 2, 3, 4, 5]
IHfactor=[0.75]

The implementation

The Integrated Performance Modelling Environment (IPME) is a network simulation software package for building models that can simulate both human and system performance [13]. We have used this program to simulate a scenario, in order to get results about the cognitive task load and performance (overall time needed to complete all the actions). The cognitive task load of an actor is defined by three variables: the time occupied (TO), the level of information processing (LIP), and the number of task set switches (TSS). The scenario that has been simulated consisted of three basic action sequences (BAS). Four persons were involved in performing these actions (2 operators and 2 managers).

With the Task Network Model in IPME you can construct the action sequences of the process that has to be simulated. Figure 3 shows an example of an action sequence, drawn in a network diagram. Arrows represent the order of the actions. Diamond shapes represent decision points.

Table 2: The CTL simulator distinguishes 5 Levels of Information Processing (LIP), defined by the experience (exp) of the task performer, the duration of the learning process and the support.

Learning process	**Without support**			**With rule provider**			**With diagnosis guide**		
	little exp	medi exp	much exp	little exp	medi exp	much exp	little exp	medi exp	much exp
long	5	4	3	3	3	2	4	3	2
medium	4	3	2	3	3	2	3	3	2
short	3	2	1	2	2	1	3	2	1

Neerincx, M.A., Dobbelsteen, G.J.H. van den, Grootjen, M. & Veenendaal, J. van (2003). Assessing cognitive load distributions for envisioned task allocations and support functions. *Thirteenth International Ship Control Systems Symposium (SCSS)*, Orlando, Florida, 7-9 April 2003.

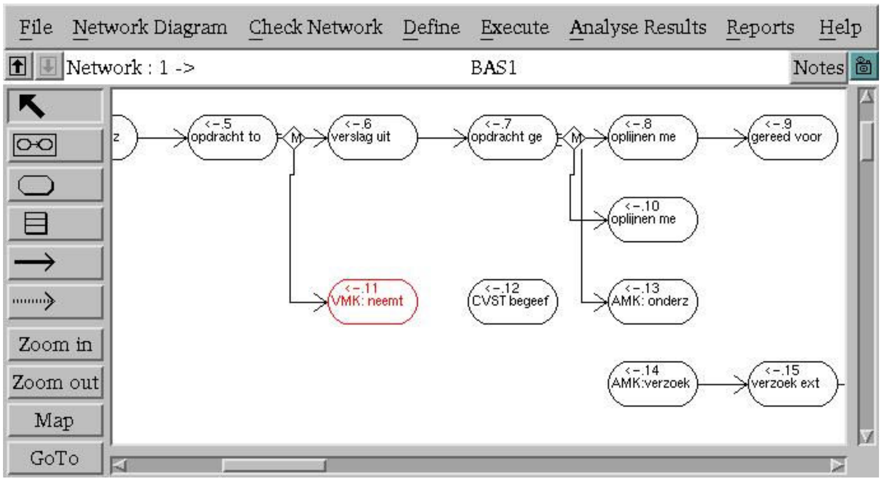

Figure 3: Part of a Basic Action Sequence (BAS) in an IPME network diagram. Arrows represent the order of the actions. Diamond shapes represent decision points.

Table 3**:** Example results of a simulation: time occupied (TO), level of information processing (LIP) and task-set switches (TSS) for an operator and an officer. These three load factors and the total crew execution time are calculated for 2 task allocations (4 and 2 persons), different support functions (DG=Diagnosis Guide, IH=Information Handler), and crew experience levels.

4 Persons			M-Operator			M-Officer			**Total**
Sup	IH	Exper	TO	LIP	TSS	TO	LIP	TSS	**Time**
no	no	Little	55.61	3.56	2	41.47	4.15	6	223.00
no	yes	Little	46.41	3.56	2	34.62	4.15	6	200.38
no	no	Average	54.61	2.56	2	40.78	3.15	6	217.00
no	yes	Average	44.13	2.56	1	32.96	3.15	6	201.38
…	…	…	…	…	…	…	…	…	…
DG	no	Much	52.69	1.47	2	38.88	1.75	6	213.50
DG	yes	Much	43.24	1.47	1	31.90	1.75	6	195.13
2 Persons			M+D-Operator			M+D-Officer			**Total**
Sup	IH	Exper	TO	LIP	TSS	TO	LIP	TSS	**Time**
no	no	Little	71.40	3.65	7	45.76	4.22	5	253.50
no	yes	Little	66.26	3.65	6	42.46	4.22	5	204.88
no	no	Average	70.90	2.65	6	45.49	3.23	5	244.00
no	yes	Average	65.49	2.65	6	42.02	3.23	5	198.13
…	…	…	…	…	…	…	…	…	…
DG	no	Much	69.94	1.51	6	44.35	1.81	5	234.50
DG	yes	Much	63.04	1.51	6	39.97	1.81	5	195.13

For each action you can specify the mean time of the action, standard deviation of this mean time, release conditions, and beginning and ending effects in a separate window. Also you can assign an actor to the action.

The start of an action sequence is caused by an event. These events can be specified in a separate scenario window. For each scenario event you can specify at what time it should occur or if it has to be started by some other event.

The scenario for our simulation consists of three basic action sequences, all with different priorities. When an event happens with a higher priority, an actor finishes the action he is performing, and then he switches to the action sequence triggered by the high-priority event. After finishing this complete action sequence, he returns to his lower priority action sequence.

The CTL simulator shows how cognitive load and overall scenario time change according to the input parameters: actor's experience (little, average or much), action's learnability (easy to learn, average or hard), and the support (none, Diagnosis Guide or Rule Provider). These parameters affect the LIP. The LIP was calculated by means of a function that would return a value according to table 2.

The Information Handler (IH) support was either not given (then IH would be 1 and of no influence on the mean time) or given (then IH would be 0.75 and therefore would decrease the meantime with 25 percent).

Running the simulation

The simulation was run 18 times while varying the different variables. At the start of each run the values of the variables were initialised. Support and experience were changed for the overall CAS (so the level of support was constantly the same for all three basic action sequences). For each run, The CTL simulator logs the following results:

- *Time Occupied*: The time that an actor was actually performing actions. This time was calculated by taking the sum of the duration of all actions that the actor had performed divided by the overall scenario time. The same calculation was done for each basic action sequence.
- *Average LIP*: The average lip was calculated by summing all LIPs of the actions an actor performed divided by the amount of actions.
- *Task Set Switches*: Every time an actor switched to a different BAS the amount of task set switches would be increased with 1.
- The *total scenario time*.

The simulation was run in two experiments, in which we varied the number of people that had to conduct the actions. In the first one we had two managers ("M- *and* D-Officer") and two operators ("M- *and* D-Operator") to handle the scenario. In the second experiment, we used only one operator and one officer to handle the same scenario. Table 3 provides an overview of the results for two human actors (i.e. for two actors who put out Damage control actions to other actors versus the same actors who will do these actions themselves).

Results

Simulation is a powerful tool to systematically analyse design alternatives and task contexts. It provides insight in the consequences of specific design choices. See for example the effects of manning reduction on the Task Set Switches of the "M+D-operator" in table 3. Another example is the condition "no support", "IH" and "average experience", for which the total scenario time is slightly lower for two persons than for 4 persons, while cognitive load increases mostly for the operator (TO *and* TSS).

Especially when both the scenarios and the amount of variables are becoming voluminous, a simulation tool is required. Because of the interaction between various variables, and the parallelism and interleaving in the action sequences, unexpected effects can occur. IPME seems to be a useful software environment for simulating cognitive task load and performance. The possibility to run experiments enables the researcher to quickly get results about many different conditions. One can easily define scenario events and task related information. Also the assignment of actors to tasks is quite flexible.

However, IPME has yet some shortcomings with respect to usability and performance. Our experiments were rather easy to implement, because the actions were already specified in flowchart diagrams (see figure 2). These neat diagrams could be easily translated into the task network of IPME (see figure 3). However, when the number and size of the action sequences grow, overview and translation may become cumbersome. The same issues apply to the assignment of actors to actions, which you have to do for every action. Because it was possible to initialise the values of the variables 'experience' and 'support' by means of a scenario event at time = 0, it was very simple to change these values during the different runs. At the end of every run the output was collected by "snapshots" (saving data at a certain point in time), which resulted in an output file with raw data that afterwards had to be translated in a better format. The overall conclusion is that the implementation of the CTL simulator in IPME was not difficult, but that running the simulations can still be rather time consuming.

Furthermore, we must emphasise that the formulas in the simulator need further validation to enable a "direct" derivation of conclusions out of the absolute load and performance values. The current version allows a qualitative comparison of design proposals for different task contexts, showing the relative consequences of design choices. Domain expertise is required to interpret the results. As such, the CTL simulator can support CTL analyses as described in section 4. For this purpose, the next version of the CTL-simulator will include simulations that make use of time frames. Instead of calculating the cognitive load for a complete scenario, it provides calculations for small periods of time (for instance every five minutes; see the example of the CTL-method in section 4). This will provide an even more detailed view on the actors' cognitive task load. In addition, the next simulator version should provide clear presentations of the action sequences and tool outcomes for adequate interpretation of the results.

6. CONCLUSION

The human role in complex task environments will be more and more focused on handling non-routine situations supported by information technology. Human task complexity increases as well as the information velocity and ubiquity. Cognitive task analyses are needed to realise an adequate human resource deployment by training, selection, task allocation and cognitive support systems. Current task analysis approaches are however diverse and differ on a number of dimensions such as scope, theoretical and empirical foundation, and utility. There is a tension between basic and applied research and insufficient correspondence between individual and team-oriented perspectives. Methods based on cognitive theory, models and architectures made progress, but are still in research state or prove to be hard to apply for real-world, complex tasks (e.g., see [14]). To enable well-founded analyses in such task environments, we have been developing a CTL theory, model, method and tool in an iterative process. Although, there is already sufficient empirical foundation for applying the current version of the model and method, further refinement and validation is required to derive absolute measures for the critical load regions of figure 1.

The CTL-simulator tool allows a systematic, qualitative comparison of design proposals for different task contexts, showing the relative consequences of design choices. It should be noted that IPME contains a workload model that we do not use in our CTL simulator [13]. The IPME model supports micro-level analyses, whereas our CTL method provides a meso-level analysis. The CTL method requires less detailed design specifications, so that it can be used in early system design stages and, in our view, is more cost-effective. Furthermore, our method is based on a model that has been developed and applied in the complex task environments of our target domain. The current version of the tool, however, needs further improvement with respect to its usability and empirical foundation. In addition to the assessment of design alternatives, the simulator can also aid with deriving *user requirements* for support systems. Such requirement might be stated in a form as: “System component X should provide task support for Y scenarios, so that the performance and cognitive task load of Z operators will not exceed the following measures…”.

7. ACKNOWLEDGEMENT

Research is teamwork and studies in complex task environments require extensive domain knowledge and Human Factors expertise. This study is the result of a continuing co-operation between TNO Human Factors and the Royal Netherlands Navy with contributions of a lot of people. Thanks are due to all of them, in particular Eddie Flohr (RNlN) and Sasja van Besouw (TNO HF).

REFERENCES

[1] Neerincx, M.A. (in press). Cognitive task load design: model, methods and examples. In: E. Hollnagel (ed.), *Handbook of Cognitive Task Design*. Chapter 13. Mahwah, NJ: Lawrence Erlbaum Associates.

[2] Rasmussen, J. (1986). *Information processing and human-machine interaction: An approach to cognitive engineering*. Amsterdam: Elsevier

[3] Parasuraman, R. (1986). Vigilance, monitoring, and search. In Boff, K.R., Kaufman, L. & Thomas, J.P. (eds.), *Handbook of perception and human performance*, volume 2: cognitive processes and performance, chapter 43. New York, Wiley.

[4] Kerstholt, J.H., & Passenier, P.O. (2000). Fault management in supervisory control: the effect of false alarms and support. *Ergonomics, 43*(9), 1371-1389.

[5] Neerincx, M.A., van Doorne, H. & Ruijsendaal, M. (2000). Attuning computer-supported work to human knowledge and processing capacities in Ship Control Centres. In Schraagen, J.M.C., Chipman, S.E. & Shalin, V.L. (Eds.), *Cognitive Task Analysis*. Mahwah, NJ: Erlbaum.

[6] Neerincx, M.A. & Besouw, N.J.P. van (in preparation). *Cognitive task load: memory capacity and switch costs.* (Report TM-03-AXXX). Soesterberg, the Netherlands: TNO Human Factors.

[7] Neerincx, M.A. & van Besouw, N.J.P. (2001). Cognitive task load: a function of time occupied, level of information processing and task-set switches. In: D. Harris (ed.), *Engineering Psychology and Cognitive Ergonomics, Volume Six: Industrial Ergonomics, HCI, and Applied Cognitive Psychology* (Chapter 31, pages 247-254). Aldershot, etc.: Ashgate.

[8] Neerincx, M.A., Grootjen, M. & Veltman, J.A. (in preparation). *Cognitive task load in a Naval Ship Control Centre.* (Report TM-03-AXXX). Soesterberg, the Netherlands: TNO Human Factors.

[9] Grootjen, M., Neerincx, M.A., Passenier, P.O. (2002). Cognitive task load and support on a ship’s bridge: Design and evaluation of a prototype user interface, *INEC 2002 Conference Proceedings,* pp. 198-207. London, England: IMarEST.

[10] Carroll, J.M. (2000). *Making use: Scenario-based design of human-computer interactions*. Cambridge: MIT press.

[11] Veenendaal, J. van (2002). Mentale belasting Officier van de Wacht: Een verkennend onderzoek naar de mogelijkheid tot taakuitbreiding en computer-ondersteuning op de brug [Mental load Officer of the Watch: An explorative study on the possibilities to add tasks and computer support on the bridge]. Master’s thesis. Royal Institute for the Navy, Den Helder, The Netherlands.

[12] Neerincx, M.A. (1999). Optimising cognitive task load in naval ship control centres. In: D. Stapersma (editor), *Proceedings of the Twelfth Ship Control Systems Symposium Vol. 1*. The Hague, The Netherlands: Royal Netherlands Navy, Directorate of Materiel.

[13] Hendy, K.C. & Farrell (1997). *Implementing a Model of Human Information Processing in a Task Network Simulation Environment.* Report DCIEM No 97-R-71. North York, Ontario, Canada: Defence and Civil Institute of Environmental Medicine.

[14] Schraagen, J.M., Chipman, S.F. & Shalin, V.L. (2000) (eds.). *Cognitive task analysis*. Mahwah, NJ: Lawrence Erlbaum Associates

BIOGRAPHY AND CONTACT INFORMATION

Mark Neerincx is head of the Usability Engineering group at TNO Human Factors, the Netherlands. His PhD-research centred on the harmonisation of tasks to human knowledge and capacities, resulting in a method for cognitive task load analysis and the design of cognitive support. Current research is focussing on cognitive models and their integration into methods that guide the development of human-computer systems and provide innovative user interface concepts (e.g. for damage control support on a ship or a personal assistant for mobile services). For one day in a week, Mark Neerincx is professor Human-Machine Interaction at the Technical University of Delft, the Netherlands.

Mark Neerincx can be reached at Neerincx@tm.tno.nl

Presented at the Thirteenth International Ship Control Systems Symposium (SCSS) in Orlando, Florida, on 7-9 April 2003.