Chapter 7

Human-Centered Design of Man-Machine Systems

This research centered on fitting man’s work in man-machine systems to man’s knowledge and capacities. First, a method for the analysis and design of cognitive task load was developed aiming at the allocation of a set of tasks to persons which corresponds to their capacities. This cognitive task load analysis detects problems and, subsequently, formulates adjustments of the task allocation to reduce the problems. An application of the method for a process control task showed its efficiency. In the second part of this research, it was investigated whether for some (remaining) problems cognitive support can be designed. A method was developed and applied for the design of an aiding function to compensate possible knowledge and capacity deficiencies of the users. Experiments showed that such aiding, comprising automatic presentation of context-specific procedural task knowledge, improves the performance of a statistical task and a process control task. The cognitive task load analysis and the design method for aiding complement each other; they can jointly contribute to a better harmonization of the task to the human knowledge and capacities.


Keywords: Cognitive task analysis, cooperative problem-solving, system design, mental load, expertise.

7.1 Introduction

A major break-through in the development of information technology was the invention of graphical user interfaces. Such interfaces can improve the usability and learn-ability of information technology and, consequently, the human task performance in man-machine systems. Computers seem to be continually better fitted to the users, because more and more computers provide a kind of “easy-to-use” graphical interface. Furthermore, the users seem to be continually better adapted to computers, because more and more people have experience with such interfaces. However, harmonizing information technology to the human task performance involves more than providing a nice interface for a software system with a pre-defined functionality. The introduction of computers at work places can change man’s work considerably causing specific human errors (see, for example, Rasmussen, 1986; Reason, 1990). An important change is the growth of abstract work which makes higher demands on the cognitive capabilities of the workers. Railway
traffic controllers, for example, controlled the switches and lights manually in former days, while, in future, they will be mainly involved in the planning of train routes (i.e., the computer will transform the plan to correct positions and colors of the switches and lights on the railway net). A graphical user interface does not guarantee that this future work is harmonized to the knowledge and capacities of future railway traffic controllers or, in other words, that they can do the planning well. Furthermore, such an interface, comprising Windows, Icons, Mouse, Pull-down or pop-up menus (WIMP), is no solution in itself; it should be designed well providing accurate guidance or help.

The title of this chapter “Human-centered design of man-machine systems” is slightly different from Norman & Draper’s (1986) book “User-centered system design”. The difference is based on our view that design should not center only around the construction of a software system. The design of a software system goes with an, implicit or explicit, design of tasks for future users. The challenge is to design a joint man-machine system in which the tasks are performed well; by the human, the machine or, cooperatively, by the human and machine. We have sought to improve human task performance by harmonizing these tasks to human knowledge and capacities.

7.2 Designing Task Load and Aiding

This thesis centered around two possibilities to harmonize tasks to human knowledge and capacities and, consequently, improve human task performance in man-machine systems. In the first part of this study, a method was developed to analyze and adjust the distribution of cognitive task load among persons. In the second part, a method was developed for the design of an easy-to-use interface and an aiding function which supplements human knowledge. Both methods are based on principles of human task performance derived from cognitive theory. Their efficiency has been tested in different domains. This section discusses the two methods and proposes to combine them to harmonize tasks to human knowledge and capacities.

Cognitive Task Load Analysis

Task analysis (TA) encompasses a family of methods, each characterized by its objectives, analysis techniques and procedure. An appropriate TA can be selected or constructed only after the formulation of its objectives (e.g., the derivation of a training program; Drury, 1987). Next to its objectives, a task analysis should be explicit about its techniques. The analysis techniques can be used for several objectives and, consequently, be part of several methods. Hierarchical modeling, for example, is being used in the classical task analysis to establish training requirements, but can also be used to analyze human information needs in man-machine systems (Shepherd, 1993). Finally, a TA-method should comprise an explicit procedure for the acquisition, recording (e.g., modeling), and interpretation of task information.

Classical methods to analyze task load hardly address the covert, cognitive actions of human problem solving. Chapter 3 presented the development of a cognitive task load (CTL) analysis which should be able to assess the task load of such actions. The objective of the CTL-analysis is to establish an allocation of a set of tasks to each person which corresponds to his or her capacities. Cognitive theories (e.g., Rasmussen, 1986) provide some important principles of human capacities and task performance. Four guidelines have been derived from these principles with respect to the number of actions in a period, the complexity of these actions (knowledge-, rule-, or skill-based), and the process of task execution (i.e., the occurrence of lengthy uninterrupted actions
and momentary overloading). Each task set allocated to a person should be in correspondence with these guidelines as far as possible.

In the CTL-analysis two analysis techniques are used and integrated: (i) hierarchical task modeling comprising a task allocation and job breakdown and (ii) a time-line analysis. The first technique provides a neat and complete overview of the task. Furthermore, the hierarchical task model is used as a “design object” which can be moulded to meet the guidelines as far as possible. The second technique, time-line analysis, centers around the process of task execution to convey actual, possibly temporary, task load problems.

The CTL-analysis detects problems in the current situation and, subsequently, formulates adjustments of the task allocation to reduce the problems. The procedure consists of three stages (see Figure 7.1). First, a hierarchical task model is constructed to establish the current task allocation among persons (i.e., jobs) and, at a lower level, their actions. In the second stage, a time-line analysis is employed to detect possible breaches of the guidelines in the process of task execution. If the guidelines are violated, the third stage is entered to adjust the task to the guidelines.

The CTL-analysis is a method to assess the cognitive demands of complex work. Not the ingredients of the method, but the integration and use of these ingredients are new. Most importantly is the integration of a selected set of human task performance principles into current task analysis techniques. Further, the method follows a rather new approach to quantify cognitive task load by, first, establishing acceptable load levels for a specific job and, then, proposing adjustments of the task for unacceptable levels. This procedure can be followed, because acceptable load levels are defined in terms of task characteristics.

The method proved to be applicable for railway traffic control and to be more useful than the “old” non-cognitive method for load assessment of the Netherlands Railways. This result is promising. The explicit description of the objective, modeling techniques, and procedure seems to make it usable for task analysts in practice. However, more empirical research is needed. Important issues for future research are (i) the correspondence of cognitive task load, as defined in the method, to the mental effort and task performance (validity), (ii) the correspondence of results from CTL-analyses on similar situations (reliability), and (iii) the usability in other domains (generality).

**Design Method for Aiding**

In practice, it will hardly be possible to allocate tasks in such a manner that no person ever has problems due to knowledge or capacity deficiencies. In railway traffic control, for example,
unforeseen traffic disturbances can lead to such a high task load for the traffic controllers that error-free human task performance is hardly possible. Such a problem might be overcome by a help facility which aids the person with his or her task performance. However, the evaluation of the SPSS help facility in chapter 4 showed that the costs of such a facility can be larger than the benefits.

Current software engineering (SE) methods and classical human-computer interaction (HCI) approaches insufficiently prescribe how to establish an optimal trade-off between the costs and benefits. The SE methods hardly address the users’ task and user-system interaction (e.g., Yourdon, 1989; Rumbaugh, 1991). Consequently, a “pure” application of these methods does not ensure the design of help the future users really need. Classical HCI approaches towards system design center around these needs, but they hardly provide techniques to design a complete help function as part of the overall software system (e.g., Norman, 1986). Current model-based approaches to interface design provide such techniques. However, it is insufficiently known how far these approaches meet general human needs, especially because the resulting interfaces are hardly evaluated (e.g., Fischer et al., 1993). Chapter 5 presented the development of a design method for a specific help function, called aiding, which is founded on general human needs and empirical evaluations. The method integrates SE and HCI. It can be viewed as a task analysis with its specific objective, modeling techniques and procedure.

The objective of aiding is to compensate human knowledge and capacity deficiencies. An aiding facility should be harmonized to general human needs and invoke minimal extra work. Three requirements for such a facility have been derived from principles of human task performance with respect to procedural task knowledge and information processing capacities. First, the facility should take the initiative to provide information at the right time. Second, the information should consist of context-specific, procedural task knowledge; it should be minimal, but comprise complete routines. Third, the user-aiding communication should be a well-integrated part of the user-system dialogue encompassing a minimal WIMP-interface.

The design method for aiding consists of a number of analysis techniques. The main point is that current modeling techniques from SE are extended with HCI techniques. Such extension is possible, because—in correspondence with current system design methods—the model to be established is divided in several smaller models. The design process consists of two steps: construction of the analysis model and, subsequently, the design model. For both models, the functional, data and behavioral perspective are distinguished.

The design of an aiding interface is part of system design. The procedure consists of three stages. First, a plain interface is designed which is easy to use and learn. The functionality of this interface is just sufficient for expert task performance. Mostly, future users of the system will not show perfect task performance. In the second stage, the users’ deficiencies in expertise are analyzed. In the third stage, an aiding function is designed for the parts users are lacking expertise.

In correspondence with the CTL-analysis, the design method for aiding is the result of integrating techniques and principles from different disciplines. The method proved to be of help to design aiding interfaces in two different domains: statistics (i.e., the HOMALS analysis task) and railway traffic control (RTC). Chapter 5 showed that the HOMALS aiding is beneficial for users with minor knowledge of HOMALS and in chapter 6 the RTC aiding improved task performance, espe-

---

1 A GOMS-approach towards help design, as Elkerton (1991) advocates, focuses on the help-text, leaving important characteristics of a help function—such as initiative and integration in overall system—out of consideration.
cially when task load was high. Thus, aiding proves to be able to compensate human knowledge and capacity deficiencies resulting in less errors and faster task execution.

These results are promising. The integration of the techniques to design an aiding interface into current SE techniques makes the method usable for designers in practice. However, more empirical research is needed to convey the conditions for successful aiding. A simulation environment such as the railway traffic control program in chapter 6 seems to be a good instrument to study the effects of task, person and support characteristics on the task performance, mental effort and learning.

**CTL-Analysis and Design of Aiding**

The *objectives* of the CTL-analysis and the design method for aiding can jointly be present in the design of man-machine systems. The CTL-analysis centers around the distribution of tasks among persons to establish jobs harmonized to human capacities. The design method for aiding centers around the interaction and cooperation between user and system in their task execution to establish a user interface and, if necessary, an aiding function harmonized to human knowledge and capacities. Aiding might provide a solution for problems which cannot be resolved solely by task re-allocation.

This can be exemplified with the railway traffic control task. Irregularities of the traffic (e.g., a switch getting out of order), which can hardly be predicted, cause an increase of task load and can, consequently, result in momentary overloading of the traffic controllers. This cannot be prevented by task re-allocation completely. An aiding facility can provide pre-planned procedures for dealing with different types of irregularities making the task easier for the traffic controller. Further, aiding can make a more flexible task allocation possible and, consequently, may result in a better general task load for railway traffic controllers (chapter 3). Region-specific knowledge could be encompassed by the aiding, so that railway traffic controllers do not have to be in a long learning phase before they can control a new region. Consequently, sudden changes in the number of tasks allocated to persons seem to become a real option to manage task load during irregularities.

In general, it is striking that the *analysis technique* of task decomposition is used in HCI (e.g., GOMS), ergonomics (e.g., Shepherd, 1993) and software engineering (e.g., Yourdon, 1989). This technique seems a good reference for integration. Software engineering can take a central position in the integration, because it provides well-developed modeling languages for task decomposition (functional, data, behavioral, and social perspective). For example, the hierarchical task model of the CTL-analysis and the analysis model (functional perspective) of the design method for aiding should correspond to each other at the level of task allocation (see Figures 3.6 and 6.1).
CTL Analysis

In the (re)design of man-machine systems both the CTL-analysis centering around the design of task load and the design method for aiding can be applied. The procedures for the redesign and design of both task load and aiding will be discussed below.

FIGURE 7.3: The procedure of the CTL-analysis and design of aiding applied to railway traffic control.

CTL-Analysis and Design of Aiding for Railway Traffic Control. Both the CTL-analysis and the design method for aiding have been applied in the railway traffic control domain. Figure 7.3 summarizes the procedure consisting of, first, redesigning the task allocation and, then, designing aiding. In this research, the current base system of the Netherlands Railways was viewed as given. Consequently, the proposed adjustments in task allocation of the CTL-analysis centered around the distribution of tasks among personnel and the design of the plain interface centered around the dialogue with this system. However, the objective of establishing a fine task allocation to persons might invoke automation or de-automation of tasks which could have been established here (i.e., in the hierarchical task model). This task allocation should be present in the design of the plain interface if a new user interface is designed.

Next to the task allocation arrow, Figure 7.3 shows a second arrow connecting the CTL-analysis and the design method for aiding, called remaining problems. The time-line analysis might detect some problems which can be solved by aiding. Consequently, the CTL-analysis can result directly in the advice to design an aiding function for a part of the task such as for dealing with irregularities in railway traffic control (see above).

In sum, the research on railway traffic control shows that the CTL-analysis and the design method for aiding can jointly contribute to a better harmonization of the task to the human knowledge and capacities. In the CTL-analysis, the current situation is analyzed to detect human problems with respect to task load and to establish possible remedies for these problems (especially adjustments in the task allocation). In the design method for aiding, the interactive role of the system is further analyzed and one specific remedy is applied: aiding.

Designing Task Load and Aiding. Above, the procedure for the redesign of task load and the design of aiding for railway traffic control was discussed. The functionality of the current system of the Netherlands Railways was viewed as given. This paragraph will propose a general procedure —consisting of the same ingredients— which can be applied for the design of a new man-machine system. The procedure consists of four stages and an iteration (see Figure 7.4).

Designing Man-Machine Tasks. Both the CTL-analysis and the design method for aiding advocate a top-down approach towards the design of man-machine systems. They start with task
decomposition and allocation of subtasks to human and machine. Based on current knowledge about human task performance, a task allocation is established which comprises the jobs, human actions, machine tasks and man-machine dialogue. Machine tasks are distinguished at a high level, that is, as far as required for the design of the dialogue.

**Human Task Performance Analysis.** Human task performance cannot be fully predicted and, therefore, this performance has to be evaluated in the design of man-machine systems. A number of criterion tasks can be given to prospective task performers with a prototype of the plain interface. The time-line analysis of the CTL-analysis and the user analysis of the design method for aiding can be jointly applied to assess, respectively, the task load and expertise.

**Adjusting the Task and/or Designing an Aiding Function.** If capacity deficiencies or knowledge deficiencies are detected some adjustments of the task allocation can be proposed and/or an aiding function can be designed. The allocation adjustments comprise an iteration in the man-machine design process. It has to be noticed that the design of the aiding function can be an iterative process: the aiding should be evaluated and may be redesigned as a consequence of the evaluation.

**In conclusion,** the CTL-analysis and design method for aiding complement each other. Figure 7.4 shows the integration of both methods into a procedure for the harmonization of man-machine tasks to human knowledge and capacities in system design.

### 7.3 Discussion

The last section showed a new approach to harmonize tasks to human knowledge and capacities consisting of a CTL-analysis and the design of an aiding interface. The approach is based on theory and consists of engineering and empirical techniques.

**Theory.** In human-computer interaction research, cognitive psychology has been one of the most important theoretical bases (e.g., Card et al., 1983). However, there is no consensus about the impact and promise of cognitive theory. Booth (1993, p.12) maintains, for example, that “one may question whether cognitive theory has anything other than a confusing language and a set
of contradictory ideas to offer”, whereas Wickens (1992) maintains that great improvements are possible by combining theories of cognitive psychology with increasingly available computer and display technology. According to Wickens, this area of cognitive ergonomics is one in which relatively little engineering psychology research has been conducted to assess the utility of the recommendations made by such researchers as Rasmussen (1986) and Woods (e.g., Woods & Roth, 1988). He maintains that the potential payoffs of this research are quite high, especially because systems are evolving towards greater complexity with increasing levels of automation.

Our research seems to agree with Wickens’ optimistic view on the usability of cognitive theory. It is especially noteworthy that the work of Rasmussen and Woods were important starting points of the development of the CTL-analysis and design method for aiding. Chapter 2 described some important cognitive and HCI theories which are relevant for the design of man-machine systems. Thus far, the discussion centered around cognitive theory. It is obvious that more disciplines are important. This thesis comprised a multi-disciplinary approach by using theories, methods and techniques from human-computer interaction, cognitive psychology, ergonomics, software engineering, and artificial intelligence.

**Engineering Techniques.** According to Eberts (1994), a user interface designer must be familiar with theory and advances in cognitive psychology. Such a person needs to be able to apply theoretical perspectives to concrete situations in human-computer interaction tasks. Thus, the designer must be a theoretician and an engineer, in addition to a designer, to apply the theories; the application of theory is dedicated to the designer. However, in our view, the bridges between theory and practice should not be build by each individual designer for each design process. Methods for cognitive engineering should be developed which ensure a good design founded on theory (cf., Rasmussen, 1986; Norman, 1986). Therefore, we integrated principles of human task performances into current design methods.

**Empirical Techniques.** Design is most often an iterative process in which unforeseen problems are detected for a first version or prototype and the problems are subsequently removed in the design of the next one. Chapter 5 refined this kind of iteration into a generate-and-test framework for the design of cognitive support. According to this framework, design is a modeling activity. Theories about human task performance should be used to generate a first design which fits the task performance of future users. However, current theories cannot predict human task performance completely. Therefore, the design has to be tested in such a way that the test results can be input in the next design generation.

Further, empirical techniques are required to acquire information about the specific task and user characteristics. The CTL-analysis and the design method for aiding prescribe a human task performance analysis. The CTL-analysis consists of a time-line analysis to establish among other things job-specific norms for cognitive task load and the design method for aiding consists of a user analysis to detect expertise deficiencies.

In general, empirical knowledge about conditions for successful computer support is for a main part lacking, because evaluations of newly designed computer systems are seldomly performed in a rigorous way (cf., Sassen, 1993). It is often maintained that such evaluations are very time-consuming and expensive. Nowadays software for prototyping and simulation is available, so that relatively cheap laboratory experiments can be conducted. These experiments can provide empirical data about the success of computer support in different conditions. Expensive field
evaluations should be carried out only after demonstrating superior performance in such an experiment (Johannsen et al., 1992).

**Toward a Compound Design Approach**

The harmonization of man-machine tasks to human knowledge and capacities summarized in Figure 7.4 comprises only a part of the issues relevant for the design of man-machine systems with fine human task performance. For example, in one of the domains of our research — railway traffic control— a multi-operator system is present in which *resident pathogens* can be developed (Reason, 1990). These pathogens are latent error opportunities, which are embedded in system complexity, procedures, training inadequacies, organizational structure or management policies, and attitudes (Wickens, 1992). Our approach will not detect such error opportunities and, consequently, will not provide remedies.

Further, next to the objective of improving human task performance, other criteria have to be taken into account for system design. The quality of working life is a well-known example (Davis & Wacker, 1987; Pot et al., 1992). In general, man-machine systems can be viewed from several perspectives encompassing of a specific set of criteria and each perspective might provide a different trade-off between the costs and benefits of design decisions. For example, starting from a task-oriented, a communicative, a tool, an ecological or a democratic perspective, the pro’s and cons of Computer-Supported Cooperative Work (CSCW) are compared differently (de Greef et al., 1991). To incorporate such perspectives in the design process jointly and establish one final trade-off, traditional bounds between disciplines have to be over-stepped.

Thus, traditional ergonomics or human factors approaches have to be extended. In ergonomics, it has already been acknowledged that micro-ergonomics fails to significantly improve overall system productivity, worker health, and the intrinsic motivational aspects of work systems (Hendrick, 1994a). In CSCW, for example, the multitude of failure factors requires an expansion of cognitive ergonomics to cope with the complexity introduced by cooperative tasks (de Greef et al., 1991). The need for a macro-ergonomics approach to the design of the work system has been stated in the 80’s (cf., Brown, 1986). Macro-ergonomics may be defined as a top-down socio-technical systems approach to the design of organizations, work systems, jobs, and related human-machine (hardware ergonomics), user-system (software ergonomics), and human-environment (environmental ergonomics) interfaces (Hendrick, 1994b).

In this definition, macro-ergonomics seems to be a kind of umbrella on top of the traditional ergonomics. Hendrick (1994a), for example, allocates cognitive ergonomics to the traditional micro-ergonomics studying individuals. However, if several agents perform a task the cognition is distributed. In this situation issues, studied by different disciplines, are of relevance. The study of distributed cognition should not comprise a simple add-on of traditional approaches (e.g., linguistics, sociology, psychology), but should integrate these approaches into a new compound approach. Such multi-disciplinary research is required for significant progress in human-centered system design. A side-effect may be progress in the separate disciplines (cf., artificial intelligence and cognitive psychology).

**Toward Cognitive Support in Practice**

In traditional HCI-research, the computer was mainly viewed as a tool (e.g., Norman & Draper, 1986). The tool-metaphor entails a “master-slave” relation between user and computer. However,
the users are not always able to play the master role and, consequently, do not make use of the computer very well. They may be lacking knowledge for a successful task execution (Carroll & Rosson, 1987) or they may have, temporarily, insufficient capacities to solve a problem properly (Sassen, 1993). In general, many human performance limitations concern constraints on a person’s ability to meet multiple, possibly concurrent, task demands. Computer support can be designed and used that allows the operator to selectively off-load tasks to automation. However, the burdens associated with managing the automation by the operator can sometimes outweigh the potential benefits of support (Kirlik, 1993; Adelman et al., 1993). Managing the automation is especially burdensome if the initiative to use it and the control of it are fully dedicated to the user. We propose to view the computer as partner of the user in a process of cooperative problem solving instead of viewing the computer as a tool. In a process of cooperative problem solving, the software system can take the initiative to do a part of the task such as the provision of cognitive support. Such support is not only available, but can also take the initiative. Then the user is not burdened with the extra task of managing the automation so that real off-loading may be accomplished. Besides, the user will not “forget” to consult the support.

We developed a design method for one type of cognitive support, called aiding, in which the computer takes the initiative to aid users with their task execution when they are in deficient knowledge or capacity. Such support seems to be especially effective for tasks which have to be done rarely. Morris & Rouse (1985) notice that enabling the operator to deal with unfamiliar situations may not be achieved solely through training, and that more attention should be devoted to providing operators with assistance in using appropriate knowledge during abnormal conditions. The aiding function seems to be able to provide such knowledge and to reduce the task load. Therefore, it may be very effective for different kinds of disturbance management in process control, but also for other emergency situations such as earthquakes, hazardous material spills or medical disasters (cf., Tyler et al., 1991). This can be exemplified with an emergency in the railway traffic control domain: the management of a derailed train. Medical assistance may be necessary, repair orders have to be given, train traffic have to be routed different from the work plan, some extra actions have to be done to prevent new accidents, etc.. Railway traffic controllers may not have the knowledge immediately available to do these rare process control tasks. Furthermore, they may need much time to do these tasks (e.g., get into contact with assistance personnel near the accident) and may make errors. The results of our experiment suggest that a cooperative computer which takes the initiative to present context-specific task knowledge (e.g., send person X to the accident, phone number Y, ...) can aid railway traffic controllers in such situations. Important for emergency management is that in the railway experiment aiding was most beneficial when task load was high.

Aiding is a specific type of cognitive support and other types can be more effective for certain situations. The problem is that the design space is very large and that the exact trade-off between the costs and benefits can only be derived in an empirical evaluation. Chapter 5 presented a generate-and-test framework to deal with this problem. This section will present some alternative types of support for situations when aiding is not appropriate. We restrict ourself to support that is integrated into the task execution of the human task performer (e.g., predictive displays are left out of consideration, because they demand a separate control).

In our research, first, an aiding interface for statistical analyses was designed and evaluated. This interface provides for each statistic a fixed help window. Subsequently, the aiding interface for railway traffic control was designed which provides more intelligent support: the information in the help window is “contextualized” and information only appears if the corresponding conditions
are true. Still, the support is based on pre-planned procedures. If it is difficult to derive procedures, other forms of support might be needed. If complete routines are hard to establish, information support can be a good alternative. A second possibility might be a support function consisting of a knowledge-based component which can execute some problem solving activities such as the generation of questions in a diagnosis process. Aiding centers around procedural knowledge, but a mnemonic device which provides an overview of declarative information might be very useful. Further, aiding comprises the presentation of knowledge about what to do in a certain context. An alternative type of support is to present knowledge about the consequences of task executions, for example, critiquing the number of I-forms in a text or the positioning of a refrigerator in a kitchen design. Below, we will show some examples of these alternative types of cognitive support which are integrated into human task performance.

**Information Support.** Raaijmakers & Voorkamp (1993) showed that operators on board of multi-purpose frigates of the Royal Netherlands Navy have problems diagnosing unfamiliar system failures. Subsequently, Raaijmakers & Verduyn (1993) designed information support which may be used by the operators at their discretion offering a number of most probable causes of an alarm and indicating which sensor information should be checked. The most important advantage of such support is that typically human capabilities remain involved which are of great value for diagnosing unforeseen disturbances. The information provided was derived from handbooks. In a simulation experiment with 16 experienced operators, the percentage of correct diagnosis increased from 62% without support to 89% with information support available. Thus, a “simple” type of support which does not provide a complete solution can be very effective. When the operators did not use the support, this decision proved to be correct (i.e., the support was not really useful in the situation). However, the faster the consult of support, the faster the diagnosis was. This last result might plead for automatic presentation of advice in urgent situations such as the aiding function for railway traffic control in chapter 6.

Raaijmakers and his colleagues followed a procedure which corresponds to the design method for aiding: there was a base system for which a user analysis was applied to detect expertise deficiencies and, subsequently, a support function was designed. However, they did hardly follow a model-based approach and the design of support was not integrated into the design of the base system. For a real implementation, the information support has to be integrated into the base system in order to connect the information to the alarms and, consequently, the design models of support and base system have to be integrated.

**A Questioner and/or Mnemonic Device.** Post et al. (1994) applied the generate-and-test framework of chapter 5 to design support for the task of emergency call handling. Ambulance dispatchers have to decide whether an emergency call from the general public—a “free call”—is urgent and dispatch an ambulance, or to refer the patient to the general practitioner. Two types of support were generated for the ambulance dispatchers.

A knowledge-based system component was developed to support the diagnosis of disorders. This “questioner” starts with asking for the main complaint such as chest pain. Based on this complaint a set of hypotheses is generated and an urgent and most promising one selected (e.g., myocardial infarct). Next, a series of questions and answer alternatives are generated to verify or falsify this hypothesis. Figure 7.5 shows a window with one possible question and corresponding answer alternatives proposed by the questioner. This type of support ensures a rather complete
acquisition of symptoms needed for a correct diagnosis, but it may interfere with dispatchers’ problem solving process leaving their interviewing expertise on-used.

An alternative type of support was designed which does not interfere with human expert task performance: a mnemonic device which provides an overview of symptoms. A session with this device starts, similar to the questioner, by asking for the main complaint. Then, the corresponding form comes up (see Figure 7.6). The dispatcher has full control of what information when to fill in.

The two types of support were evaluated in an experiment in which dispatchers had to handle simulated series of emergency calls. The experimental results showed that decision making performance strongly depends on factors in the task environment, such as initiative of the caller and the urgency of the call. Based on these results, a new type of support is proposed which combines the benefits of the questioner and mnemonic device, and which minimizes their individual costs. The specific role of the future support system may be adapted to the urgency of the call and the available knowledge and capacities of the dispatchers. Thus, adaptive support in Rouse’s (1988) sense—as an element of an overall intelligent interface which includes Artificial Intelligence modules—may be effective in future ambulance dispatching.

Critiquing. Critics “... use knowledge bases of heuristics to affect the judgment and performance of users doing a task (...) they are capable of recognizing when their human collaborator has strayed from the normative problem-solving path. In such a case, the critics are triggered, and they help influence, de-bias, and otherwise affect the human’s biased judgment to reach a more correct task outcome” (Silverman, 1992, p. 170).

Fischer et al. (1993) provide a nice, simple example of a critiquing function, which should support the computer user in designing a kitchen. The user of this system constructs a kitchen using components provided in a palette of kitchen units (e.g., dishwasher, sink). For a good kitchen design, plumbing guidelines have to be accounted which, for example, require the dishwasher to be within one meter of the sink. The critiquing facility of the system encompasses of this knowledge. If the user puts the dishwasher more than one meter from the sink the system critiques this placement and remarks: “The dishwasher is too far from the sink”. Certain critics are contextualized; the system might for example inform the user that if the cook is left-
handed, the dishwasher should be on the left side of the sink. This type of support — deliver the right information at the right time — is built, among others, around insights gained from situated cognition research. However, the main objective seems to be to investigate how such environments can provide timely and relevant knowledge. A solid analysis of the use of such systems and the effect of the critic is not done. It is still an open question whether the critic improves the design of kitchens.

Silverman (1992, p. 193) maintains that as task difficulty increases, a point will be crossed at which subject-matter experts can no longer be assisted by critiquing alone. Thus, under conditions of high task load critiquing seems not to be the best type of support. A further restriction is that the users must have some knowledge to start their task execution. If they do not know which goals to perform, then they cannot be critiqued. Taken together, aiding seems to be a better type of support than critiquing to compensate substantial knowledge and capacity deficiencies of the human task performer.