The Cognitive Architecture of Decision Support Systems for Industrial Process Control

Vicente, K.J.; Rasmussen, Jens

Publication date:
1988

Document Version
Publisher's PDF, also known as Version of record

Risø-M-2696

RISØ

Risø-M-2696

The Cognitive Architecture of Decision Support Systems for Industrial Process Control

Kim J. Vicente and Jens Rasmussen

Risø National Laboratory, DK-4000 Roskilde, Denmark
March 1988

RISO-M-2696

THE COGNITIVE ARCHITECTURE OF DECISION SUPPORT SYSTEMS FOR INDUSTRIAL PROCESS CONTROL

Kim J. Vicente and Jens Rasmussen

Abstract. There are two properties of process control environments that create a need for effective computerized decision support. First, the fact that the system is usually quite reliable means that faults are relatively infrequent. Consequently, operators have great difficulty in dealing with these situations because they are so unfamiliar to them. To make matters even worse, it Ls under those very same rare, abnormal conditions that the risk of endangering system safety is greatest. Decision support systems are being built to help operators cope with these demands. As a result, the operating staff and the computerized control system are involved in a complex cognitive system, in which several different task allocation policies can be chosen. This paper discusses how to determine what the cognitive architecture of the decision support system should be in order to achieve effective and reliable performance. A taxonomy of decision support techniques is proposed, and the appropriateness of each for the various problem solving tasks that are typically encountered in process control is discussed. The considerations associated with the presentation of evidence in the form of a problem representation and the presentation of expert advice are treated in greater detail.

March 1988

Riso National Laboratory, DK-4000 Roskilde, Denmark

An earlier and shorter version of this paper was presented at the First European Meeting on Cognitive Science Approaches to Process Control. October 19-20, 1987. Marcoussis, France.

ISBN 87-550-1405-4

ISSN 0418-6435

LIST OF CONTENTS

		page
1.	INTRODUCTION	5
2.	A FRAMEWORK FOR COGNITIVE TASK ANALYSIS	6
3.	A TAXONOMY OF DECISION SUPPORT TECHNIQUES	7
	3.1. Process Control Activities	8
	3.2. Data Retrieval, Analysis, and Presentation	10
	3.3. Fully Automated Decision System	10
	3.4. Data Filtering and Selection	11
	3.5. Data Integration to the Decision Level	12
	3.6. Problem Representation Matching User	13
	3.7. Computerized Advice Giving	14
	3.8. Conclusion	15
4.	PROVIDING THE OPERATOR WITH EVIDENCE	15
	4.1. Ecological Interface Design	15
	4.1.1. Making visible the invisible	16
	4.2. Cognitive Control of a Process Plant	16
	4.2.1. Skill-based level	17
	4.2.2. Rule-based level	17
	4.2.3. Knowledge-based level	17
	4.3. Summary	20
5.	PROVIDING THE OPERATOR WITH ADVICE	21
	5.1. Limitations of the Expert System Approach	21
	5.2. The Epistemology of Expert Systems	23
	5.2.1. Heuristic classification	23
	5.2.2. Support knowledge	27
	5.2.3. Structural knowledge	28
	5.2.4. Strategic knowledge	29
	5.3. Implications for Process Control	29
	5.3.1. State identification	30
	5.3.2. Decision making	32
	5.4. An Alternative Approach to Providing Advice	33
6.	CONCLUSIONS	34
	ACKNOWLEDGEMENTS	35
	REFERENCES	35

1. INTRODUCTION

Several recent trends in technological development have some significant implications for the problems encountered in designing decision support systems. Centralization leads to large and, hence, potentially risky installations. Therefore, it is not enough to design decision support systems that are very efficient and successful for frequently encountered tasks. It is important that system performance measures reflect performance during complex rare events as well. Another consideration is that automation has moved the operating staff to higher level supervisory control tasks. Thus, operator support is essential for complex cognitive tasks such as diagnosis and planning. As a result, the operating staff and the computerized control system are involved in a complex cognitive system, in which several basically different task and responsibility allocation policies what we call the cognitive architecture of the decision support system - can be chosen.

Clearly, providing effective decision support for industrial process control systems is a very important, but very complex, issue. In this paper, we discuss the problems and considerations that should be taken into account when selecting an appropriate form of decision support for such systems. First, we describe a framework for cognitive task analysis (CTA) which provides a methodology for selecting an appropriate cognitive architecture for the DSS. Secondly, we propose a taxonomy of decision support methods, and discuss the appropriateness of each category of support for typical plant requirements. The CTA framework will be used as a reference for evaluating the match between the demands of the task and the form of support being provided. Finally, we provide a more detailed discussion of two support functions: decision support through presentation of evidence, and decision support through presentation of advice. While we will only be concerned with process systems in this paper, much of the discussion generalizes to other complex domains (cf. Vicente, 1987 for the domain of emergency management).

2. A FRAMEWORK FOR COGNITIVE TASK ANALYSIS

A methodology is required f or matching the characteristics of a problem domain to the different forms of computerized decision support that technology has made available. Rasmussen (1986) has proposed a framework for cognitive task analysis (CTA) that serves this purpose. In this section, we will provide a brief description of the rationale behind the framework, as well as the different phases of the methodology. The approach described here is

in contrast to the more traditional method of designing according to normative procedures.

In general terms, the goal of a CTA is to design a DSS that provides the operator with a resource envelope within which he can act in normal, as well as unforeseen situations, without violating his resource limitations. This goal is accomplished through a four phase methodology. First, the functional properties of the system are represented in a problem space defined by a part-whole dimension and a means-end dimension. This is a technical analysis of the system from an engineering perspective with the purpose of identifying the system's control requirements. The second phase consists of analyzing the decision making activities associated with meeting the control demands of the domain. This should be conducted within a device independent framework, such as the decision ladder described by Rasmussen (1986). During this stage, it is also important to evaluate the extent to which stereotypical bypasses in the decision sequence can be analyzed and implemented by automatic functions in order to simplify the decision task during actual operations for well structured and foreseen situations. The third phase in the CTA methodology consists of identifying the mental strategies and heuristics that can be used to effectively and reliably carry out the decisions outlined in the previous phase. This usually requires empirical studies of the strategies people use in their work context so that the resource requirements for each strategy, and the criteria that people adopt f or strategy selection can be determined. Finally, the fourth phase is an evaluation of the match between the resources available for implementation of the strategies by means of the three decision making agents: designer, operator, and process computer. The product of this phase should be a specification for the allocation of the decision making activities between these three agents. For this, a set of models of human information capabilities and limitations is necessary, together with knowledge of the subjective task formulation and performance criteria which are likely to control the choice of strategy in the actual situation.

The procedure outlined above will allow designers to develop an appropriate cognitive architecture for the DSS, one that maximizes the match between the demands of the domain and the resources available to each of the three decision making agents. Similar procedures have been applied to determine the information support requirements for various work domains (cf. Cohen, May, and Pople, 1987; Hoc, 1987). Typically in process control systems, the resulting cognitive architecture will require an intimate cooperation between designer, operator, and process computer.

3. A TAXONOMY OF DECISION SUPPORT TECHNIQUES

In this section, we propose a taxonomy of decision support techniques. Using the CTA framework as a reference, we will discuss how appropriate each form of support is for the various problem solving tasks that are typically encountered in process control. Before entering into the description of the taxonomy, it is important to briefly describe the problem solving activities that we are attempting to support.

3.1. Process Control Activities

Problem solving in a process plant consists of three generic activities: state identification, decision making, and planning. These are illustrated in Figure 1, which is a simplified version of Rasmussen's (1986) decision ladder. State identification, which in unfamiliar situations becomes diagnosis, creates a need for integrating the individual physical variables that are measured by the system's sensors into higher level information characterizing the situation to act upon. The decision making activity consists of selecting a target state to be achieved. Usually, it takes place at a higher conceptual level than either data measurement or action. Finally, planning involves the decomposition of intentions onto actions on elementary components, such as switches and valves. The discussion below is centered around how to match the form of decision support provided by the different categories in the taxonomy to the requirements of each of these different processing tasks.

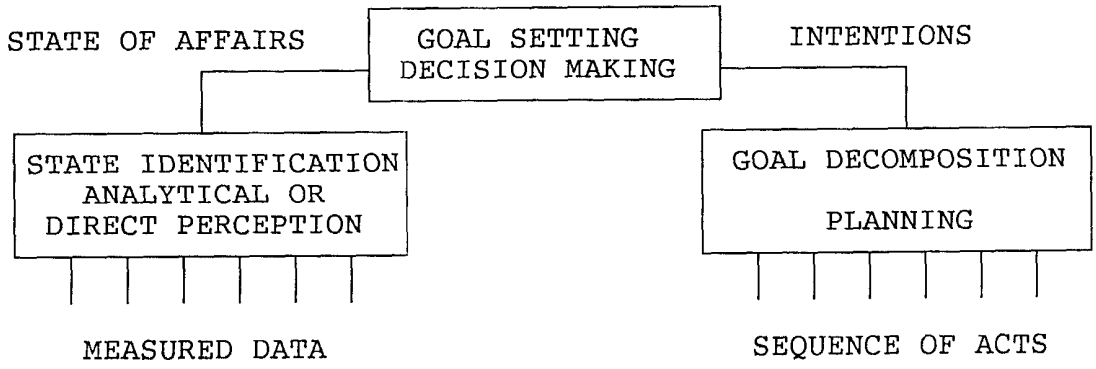

Figure 1. Generic Problem Solving Activities in Process Control.

There are many different forms of computerized decision support that could be provided for industrial process control. Table 1 describes a taxonomy of the general categories of computer support. The list is not meant to be exhaustive nor definitive, but merely illustrates the range of alternatives that can be considered for information support. The categories are listed in decreasing order according to the degree of user involvement in the decision making process. The categories can also be described with reference to Fig-

ure 1, since they represent a logical progression of support of the operator's problem solving activities. Thus, categories 1 and 2 are related to the retrieval of measured data. The third category is concerned with relieving the operator's load by having the computer perform the data integration associated with state identification, the goal decomposition involved in planning of action sequences, and the execution of actions, thereby allowing the operator to concentrate on goal setting and decision making activities. The fourth category of decision support goes even further by adapting the presentation of information to the mental strategy that the operator is adopting. The fifth category is different from all the preceding ones since it supports the decision making activity itself by providing advice. Finally, in the last category, the computer actually carries out the decision making activities itself. Each of the categories is described below in more detail, and the important considerations for each are discussed.

Table 1. Categories of Computerized Decision Support for Industrial Process Control.

CategorV	Form of Decision Support Provided
1	Data retrieval, analysis, and presentation.
2	Data filtering and selection.
3	Data integration to decision level.
4	Problem representation matching user.
5	Advice with explanation facility.
6	Fully automated decision system.

3.2. Data Retrieval, Analysis-,- and Presentation

The first category represents the case where the computer system serves as a means f or communicating raw data to its users. This stage merely represents a simple transfer of the one-sensor-one-indicator technology found in traditional control rooms to a computerized medium. Data are presented individually even though they may be displayed in various screen formats (e.g., grouped by function). Computer support would be in the form of sensor and data validation and display formatting. The operator has unlimited access to all primary data but he is left with a data retrieval problem, searching for the proper format, that he must deal with. This function may require special retrieval tools. With this category of support, the locus of decision making control resides entirely with the user, and he must base his decision on primary data alone.

3.3. Fully Automated Decision System

At the other extreme, category six, we have the case where the entire decision making process is completely automated. In this case, the locus of control resides entirely with the computer (representing the designer's foresight); no operator involvement is required. This form of decision support is most appropriate for well-structured situations. For example, in the case of a plant start-up sequence, the designer can carry through all the decision steps, except the execution itself, and store the design in a decision table in a computer. Complete automation of decisions is also typically applied for safety functions, such as automatic shut-down. In order to cope with rare, abnormal events in a reliable way, it is important to adopt a very conservative definition of symptom patterns to guarantee the integrity of the plant. Therefore, the operator is left with an important task. To protect plant operation from unnecessary intervention due to conservative safety measures, he must keep the state of the system away from the operating regions that trigger the automatic protection mechanisms.

It should be noted that, even when the computer performs stereotypical control based on design decisions, the operator will be required, or desire, to monitor performance of the automatic system. Thus, he must be supplied with information on system states and designers, intentions, thereby enabling him to verify decisions by his own preferred decision strategy. The importance of this type of support cannot be overemphasized (cf. Lehner and Zirk, 1987). Without it, the operator will be ill equipped to take over control of the system when an abnormality occurs. Perhaps even more importantly, due to a lack of understanding, he may also attempt to take over control of the system when he should not, as was the case with the Dresden 2 accident. Making the activities and strategies of automated functions transparent to the operator will thus allow him to retain a more detailed knowledge of the current state of the system. This will provide him with a stronger basis for deciding when and how to manually take over the system. Taxonomy categories 3, 4, and 5 indicate the type of support that should be provided to improve understanding of automated functions.

We have actually provided a very simplified description of this type of support. While the discussion has benn limited to the case where the operator has no involvement in the decision process, instances of automation can be further classified into finer categories (e.g., whether the operator is notified of the computer's action, whether an explanation is given, and so on). For a more in depth discussion of these factors and their relationship to supervisory control, see Sheridan (1987) and Moray (1986).

3.4. Data Filtering and Selection

In the second category, the operator only has access to the information that is relevant to the current decision making activities. The computer filters away the extraneous raw data, based on its knowledge of the current process state. An example of this type of support is the use of logical f ilters to reduce the number of alarms that are activated at any one time. This form of support requires that the designer perform an analysis to identify constraints that can be used to successfully minimize the presentation of information, and then program the results, usually in the form of a decision table, into the process computer. An important consideration is that the analysis performed by the designer may be inadequate to cope with multiple faults due to non-additivity of symptoms (see, for instance, the discussion in Rasmussen and Rouse, 1981, p. 350).

3.5. Data Integration to the Decision Level

In the third category, the representation of information is tailored to the user's decision making needs. Unlike the previous categories which only communicate primary data, in this category, the data are processed by the system to provide higher level information about the state of affairs. In a similar manner, the user can express his intentions at a high level of abstraction as these will be decomposed into actions on lower level components by a computerized sequence controller. It is important to note, however, that this form of support does not force the user to adopt a given high level of representation. Rather, the DSS makes different display formats available to the user and allows selection according to the actual need. The result is information that is more appropriate for decision making purposes. The reduction in mental effort allows the users to devote their attention to the actual decision problem, while the computer takes care of data analysis and planning. For an example of a successful implementation of this form of advice, see Mitchell and Saisi (1987).

This form of decision support should be adopted only when it can be based on a consistent engineering analysis. Whether or not this will be possible depends on two factors: first, whether it is possible to accurately describe the process, behavior in a formalized manner, and secondly, whether it is possible to develop a control algorithm that will deal with the degrees of freedom that are necessarily associated with control from a high level of abstraction. In principle, systems based on symptomatic state identification derived from the designers preview of situations and the related engineering analysis are only suited to routine, well structured situations, not for disturbance control. In these cases, a designer can only offer advice, which is discussed in a subsequent section. Another support for the problem identification function is to arrange data in graphic formats representing the relational structure of the process to be controlled in a way that supports direct

perception and manipulation. A theoretical framework that suggests principles for implementing this type of decision support will be presented in a later section.

3.6. Problem Representation Matching User

In the fourth category, the information system is designed to be sensitive to its users. In this case, the DSS would develop a model of each user's subjective preferences. This model would then allow the system to identify the mental strategy and the level of functional representation a user is adopting, and thereby fit the preprocessing of data and level of intentions to that particular strategy. As with the previous category, the user can over-ride the context sensitivity mechanism if he wants to view a different display representation. An example of this type of support would be a DSS that could identify whether an operator was adopting a symptomatic or topographic diagnostic search strategy (Rasmussen, 1986). Because the strategies have very different requirements in terms of resources, information, and actions, the operator's task would be greatly facilitated if the computer would adapt the sequence and format of information presentation according to the strategy being adopted. A first step towards this type of decision support has recently been taken by Rubin, Jones, and Mitchell (1987) who have developed a program, based on a blackboard architecture, that is capable of inferring an operator's intentions.

This form of decision support should be adopted when it is possible for the designer to identify a set of mental strategies that operators adopt, as well as their respective support requirements. This knowledge can then be used to constrain the content or form of the information that should be displayed to the operator. The intent is to, whenever possible, provide a problem representation that is appropriate for the current decision making needs, thereby reducing cognitive load.

3.7. Computerized Advice Giving

In the fifth category, computer support is provided in the form of expert advice. In this case, the computer is not only providing a mediating level between the operator and the process, but it is also acting as an assistant or advisor, performing analyses requested by the operator. Whereas the previous level provided the operator with evidence, this level provides the operator with advice. As an example, the user could provide input data into the system, and the system would provide recommendations for action or hypotheses by accessing its knowledge base of rules. The final decision concerning what action to take, however, resides with the user. In addition, the system

could also provide some form of explanation facility so that the user can know how the system arrived at its conclusion.

An important consideration for an advice giving *DSS is* to provide the operator with information about the complex relationship between overall purposes and goals and the intentions behind the design at the lower levels of functions and equipment (Rasmussen, 1987). While often ignored, top-down information about the designers, intentions and reasons is crucial for judging whether preplanned procedures or other safety measures such as interlocks can be safely overridden.

This category of decision support is appropriate for situations where the designer is able to plan proper tasks and procedures, for instance for system protection, but is not able to foresee the related disturbed system states. In this case, diagnosis must be performed on-line by operator and computer in various degrees of cooperation. This category of support will be discussed in more detail in a later section with reference to the characteristics of expert systems.

3.8. Conclusion

The preceding discussion suggests that two functions are particularly important for the different categories of decision support. One is to present preprocessed information to the system operators in a way that matches their immediate task at the level of their mental processes. Another is the advice giving function. These two aspects will be discussed in more detail in the following sections.

4. PROVIDING THE OPERATOR WITH EVIDENCE

In this section, we will present a discussion of the problems related to the presentation of pre-processed information to an operator in a way supporting direct perception and manipulation, i.e., by designing a high level representation of the problem space.

4.1. Ecological Interface Desiqn

Ecological interface design (EID) is a theoretical framework that provides principles for developing appropriate mappings between the process being controlled, the interface surface, and the operator's mental model. In this section, the fundamentals of EID are discussed and then illustrated by considering the activities associated with cognitive control of a process plant. For a more comprehensive account of EID, see Vicente and Rasmussen (1988) and Rasmussen and Vicente (1987).

4.1.1. Making visible the invisible

EID is based on the skills, rules, and knowledge framework of cognitive control proposed by Rasmussen (1986). The principal goal behind the theory is to design an interface that will not force cognitive control to a level higher than that required by the demands of the task, and yet that provides the appropriate support for each of the three levels. In order to design such an ‚ecological interface', the following factors must be taken into consideration. First, it is necessary to merge the observation and action surfaces so that the time-space loop is maintained, thereby taking advantage of the efficiency of the human sensorimotor system. In addition, it is also necessary to develop a consistent one-to-one mapping between the abstract properties of the internal process to be controlled and the cues provided by the manipulation/observation surface. The idea is to make the invisible, abstract properties of the process (those that should be taken into account for deep control of the process) visible to the operator. In semiotic terms, this means that the cues provided by the interface have a consistent mapping onto the symbolic process properties. In this way, the same conceptual model may act as a symbolic representation when considered in relation to the elements of the environment and the laws controlling their relationships, and as a system of prescriptive signs when considered in relation to the rules for model actions on the system.

4.2. Cognitive Control of a Process Plant

Figure 2 illustrates the mappings between the process, the interface, and the operator's mental model for a typical process system. The activities associated with each of the three levels of cognitive control are described below.

4.2.1. Skill-based level

Because the operator cannot directly observe or act on the process, the sensorimotor control patterns at the skill-based behavior level will only be concerned with the manipulation of items on the interface surface. The use of a mouse or a trackerball is preferred to command languages for this task because it maintains the communication of spatial-temporal aspects of the perception-action loop intact. To allow the development of a high degree of manual skill, the interface must be designed in such a way that the aggregation of elementary movements into more complex routines corresponds with a concurrent integration (i.e., chunking) of visual features into higher level cues for these routines. Thus, the display of information should be isomorphic to the part-whole structure of movements rather than being based on an abstract, combinatorial code like that of command languages.

4.2.2. Rule-based level

The rule-based level governs the choice of control alternatives. The display provides the operator with signs that he uses as cues for the selection of an appropriate action. Typically, the action alternatives consist of a set comprised of operating procedures and routine control strategies. One of the problems with conventional interfaces is that the cues they provide the operators with are not uniquely defining with respect to the current process state. The result is that the cues that operators rely on are optimized for frequently encountered situations, but they can lead to 'procedural traps' in novel situations. EID attempts to overcome this difficulty by developing a unique and consistent mapping between the symbols that govern the behavior of the process, and the signs, or cues, that the interface displays. This will reduce the frequency of errors due to procedural traps because the cues for action, being based on abstract process properties, will be uniquely defining.

4.2.3. Knowledge-based level

Knowledge-based behavior consists of abstract reasoning based on a mental model of the process. EID supports this level of cognitive control through the mapping of signs onto symbols. This mapping turns out to be very complex because the symbolic reference can be to several different conceptual levels representing general functions, physical processes, or equipment anatomy, depending on the actual circumstances (Rasmussen and Goodstein, in press). This means that, in addition to serving as cues for action, the same display configuration can also be interpreted in several ways as symbols for reasoning. Thus, if the display configuration is interpreted symbolically, it presents the operator with a visible model of the process that can support thought experiments and other planning activities. In addition, it is suggested that such a mapping will also support functional understanding necessary for error recovery. If signs can also be interpreted as symbols, then this may force the user to consider informative aspects when looking for action cues.

Display formats having these features can, in some cases, readily be developed from the ‘externalised mental models' which are normally being used as a support of functional reasoning in the form of graphic representation of relational structures such as technical drawings, graphs, and diagrams. Semiotic analyses of the use of such professional representations in actual work (Cuny and Boy6, 1981) have shown that they are actually interpreted as prescriptive signs or descriptive symbols, depending on the requirements of the task. Such an interface based on the engineering representation of

two-phase thermodynamic systems in terms of a Rankine cycle diagram has been proposed by Beltracchi (1987).

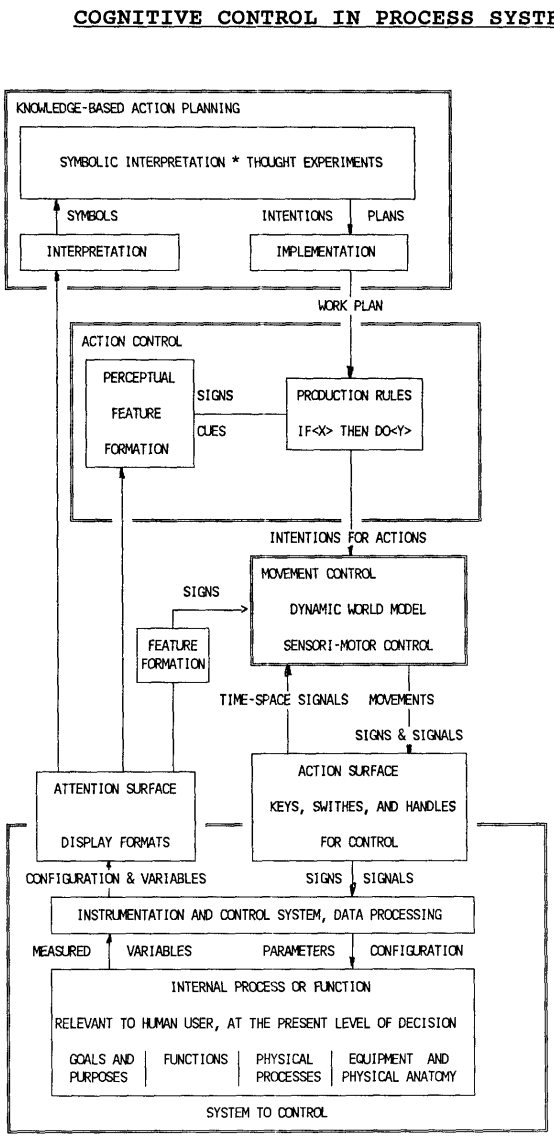

Figure 2. The figure illustrates the complex mapping between the different levels of representation of the invisible process and the different levels of cognitive control of operator action.

4.3. Summary

The EID approach to interface design can be summarized as follows:

1. Synthesize the control and the observation surfaces so that interaction can take place via time-space signals;
2. Have the computer perform the translation task by developing a consistent, one-to-one mapping between the invisible, abstract properties of the process and the cues or signs provided by the interface.

3. Display the process, relational structure directly to serve as an externalised mental model that will support knowledge-based processing.

The framework has the advantage that it is based on fundamental properties of human cognition (the SRK model), which in turn means that its generalisability is greatly enhanced.

5. PROVIDING THE OPERATOR WITH ADVICE

In this section, we will analyse the case where the operator and the computer are involved in co-operative decision making. We begin by considering the classical expert systems approach.

5.1. Limitations of the Expert System Approach

Bobrow, Mittal, and Stefik (1986) provide an excellent review of both the capabilities and the limitations of state of the art expert systems. Their account is founded on the experiences, both successful and otherwise, that knowledge engineers have had in building various expert systems. They begin by stating: "Expert systems are no panacea for achieving the impossible or even the very difficult .... Instead, there are a number of fundamental issues and requirements that must be considered" (Bobrow et al., 1986, p. 881-2). Based on their extensive experience in the area, they then go on to provide a set of guidelines for choosing appropriate problems and developing successful systems. The emphasis is on the fact that the characteristics of the application are the prime factors in determining the success of the expert system. In this section, we will take up some of their guidelines and see how the domain of process control measures up to the types of problems they consider appropriate for expert system use.

In general terms, expert system technology is best suited for tasks that are "fairly routine and mundane, not exotic and rare" (Bobrow et al., 1986, p. 886). Applying this recommendation to process control leads to the conclusion that expert systems may be appropriate f or supporting the operator during normal conditions, but not during the rare situations where abnormalities are present. However, as mentioned earlier, it is during these unforeseen events that decision support is needed the most.

Another prerequisite for a successful expert system is the availability of a domain expert. "The expert must ... understand what the problem is and have actually solved it quite often. it is not enough to have somebody with a theory about how cases like this should be handled or some good ideas about a new way to do things" (Bobrow et al., 1986, p. 887). Again, this description does not compare favorably with the characteristics of rare event detection and diagnosis. Each abnormality presents the operator with a

unique problem, and therefore, it will not be possible for designers to anticipate and effectively support all of the faults that an operator may encounter. Anticipated decision support will always be unreliable. In more direct terms, "there are no Three Mile Island diagnostic experts" (Rasmussen and Rouse, 1981, p. 689).

Also, Bobrow et al. (1986, p. 887) warn that "problems that are known to require English-language understanding, complicated geometrical or spatial models, complex causal or temporal relations, or understanding of human intentions are not good candidates for the current state of the art in expert systems." Clearly, process control posesses several of these problematic qualities. In particular, it is the causal nature of these systems allowing events to propagate over time which makes them so complex and difficult to control.

The evidence presented so far gives strong reasons to believe that expert systems may not be the most appropriate way to aid the operator in process control, at least under abnormal conditions. However, the guidelines presented by Bobrow et al. (1986) are based on the experiences designers have encountered in implementing expert systems to date. It may be the case that the limitations identified above are not inherent in the expert system approach, but rather they may be a result of present-day implementations. If the latter is true, then it may indeed be possible to build effective DSS in the expert system tradition for process control environments. To determine whether or not this is the case, we must determine what, if any, are the limitations that are not inherent in the approach, and what can be done to overcome these.

5.2. The Epistemology of Expert Systems

In order to partition the limitations of existing expert systems into those that can be avoided and those that cannot, it is necessary to have an abstract description of what an expert system does and how it does it. To fit our purposes, such a description should refer to different types of knowledge represented within the expert system, but in a way that is independent of the particular application. Clancey (1985) has developed an epistemological description of expert systems which meets these requirements.

5.2.1. Heuristic classification

Based on his extensive involvement with the MYCIN project and a thorough analysis of many other expert systems, Clancey (1985) has determined that all expert systems solve problems in the same general way. This method, called heuristic classification, involves relating concepts in different classifi-

cation hierarchies by non-hierarchical, uncertain inferences. The general structure is shown in Figure 3. From this f igure it can be seen that, in heuristic classification, data statements are first abstracted and then associated, preferably with specific problem solutions, or alternatively, with features that characterize a solution. As an example, MYCIN heuristically relates an abstract characterization of the patient to a classification of diseases (see Figure 4). This description of heuristic classification is necessarily a simplified account. Clancey (1985) provides a more detailed description of the method along with several references to existing expert systems.

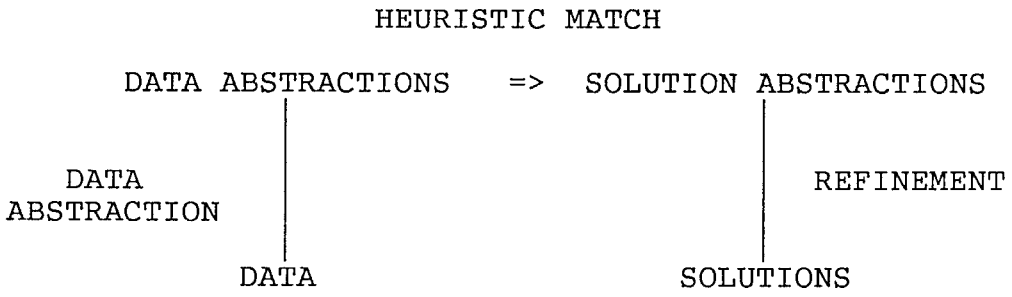

Figure 3. Inference Structure of Heuristic Classification, (adapted from Clancey, 1985).

Since heuristic classification is an abstract concept, it is useful to illustrate it by a concrete example. Figure 4 illustrates the line of reasoning behind a single MYCIN rule: If the patient's white blood count is less than 2.5, then the infection is of type E.coli. There are two interesting things to note. First, most of the reasoning shown in Figure 4 is not explicitly embedded in the rule. The knowledge that allows one to infer that there is a compromised host from the fact that WBC < 2.5 is simply not represented. This presents some problems which we will describe later. Secondly, it should also be noted that the inference from the patient abstraction to the disease class is a heuristic inference, i.e., it may not hold for every possible condition. While the rule in Figure 4 is a very simple one, its characteristics as described above apply to all of the rules in MYCIN. Both of these characteristics of heuristic classification have important implications for the method's generalizability as a means for decision support. These considerations are best illustrated by a case history.

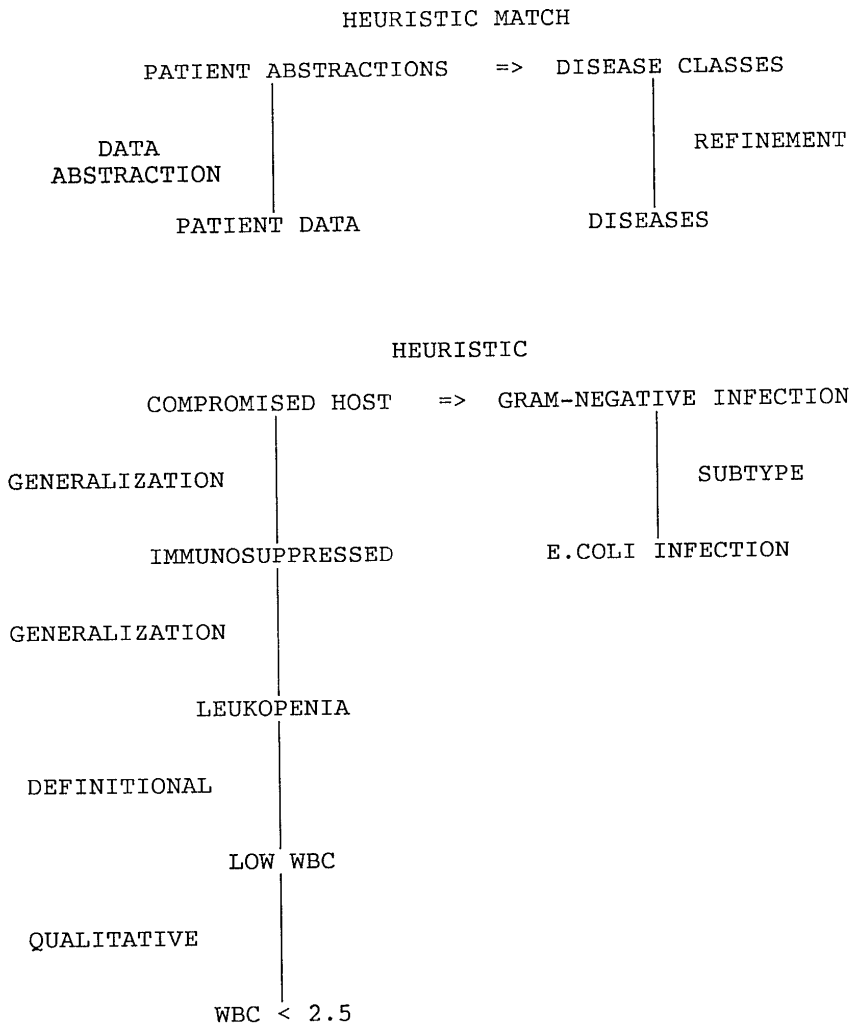

Figure 4. Inference Structure of MYCIN. Adapted from Clancey (1985).

An effort was conducted to adopt MYCIN's explanation facility as a basis for tutorial instruction for medical students. The resulting program was called GUIDON. This seemed like a promising idea since MYCIN was designed to explain its reasoning to the user, i.e., how a request for data relates to a goal, how one goal leads to another, and how a goal is achieved. However, as Clancey (1983, p. 217) states: "It was surprising to find out how little the explanation facility could accomplish for a student". Naturally, the explanations provided by the system are entirely in terms of the rules and goals that are represented in the system. The difficulties that medical students experienced with GUIDON can be attributed to the fact that the knowledge that a student needs to learn and understand is not explicitly in the system. To quote from Clancey (1985, p. 216): "GUIDON cannot justify the rules because MYCIN does not have an encoding of how concepts in a rule fit together. GUIDON cannot fully articulate MYCIN's problem solving strategy

because the structure of the search space and the strategy for traversing it are implicit in the ordering of rule concepts".

The fact that much of the domain knowledge is not explicitly represented in the system not only leads to problems in learning, as was the case in GUIDON, but it can also inhibit error recovery. In order to be able to detect errors, it is not enough to merely monitor the outcome of the decision making process since this will lead to detection beyond the point of recovery. Thus, understanding of the functioning of the system behind the task and knowledge of the intended dynamic behavior is necessary (Rasmussen, 1987). Therefore, in order to provide comprehensive decision support that will facilitate error recovery, it is important that this type of information be explicitly represented in the DSS.

There are three basic classes of knowledge that are not represented within MYCIN: strategic, structural, and support knowledge (Clancey, 1983). Each of these is important for the instructional purpose that GUIDON was designed for, and their absence accounts for GUIDON's failure as a tutoring device. More importantly for our concern is the fact that the absence of strategic, structural, and support knowledge also seriously limits any program's utility as a DSS. We will discuss how each of these knowledge classes is important for effective decision support.

5.2.2. Support knowledge

Support knowledge is theoretical knowledge that allows the system to reason from first principles. An example would be a thermodynamic description of a steam engine. Clancey (1983) found that all of the expert systems he investigated did not have any form of support knowledge. They were all based on heuristics rather than formal process models. He argued that the lack of support knowledge in these systems is due to the fact that the domains for which the expert systems were developed cannot be adequately described in terms of a theoretical model. If support knowledge were available, there would be no need to resort to heuristics.

Recently, there has been more and more discussion about support knowledge in AI systems (cf. Fink and Lusth, 1987; Hollnagel, 1988; Milne, 1987). The type of knowledge that a system has been characterized as a continuum with deep knowledge at one end and shallow knowledge at the other. Shallow knowledge forms the basis of traditional expert systems. In these systems, reasoning is based on uncertain inferences (i.e., heuristic classification or rule-based level support), e.g., an expert's 'tricks of the trade'. While shallow systems are based on experiential knowledge, deep systems are based on a formal model of the process (i.e. , support knowledge, or knowledge-based level support), and thus are much more robust.

While most existing expert systems are based on shallow knowledge, recent research in AI diagnostic systems has emphasized the deep knowledge approach (Fink and Lusth, 1987). Examples of diagnostic systems based on deep knowledge are described by: Hudlicka and Lesser (1987) ; Nawab, Lesser, and Milios (1987) ; and Scarl, Jamieson, and Delaune (1987). Most of the limitations of current expert systems outlined by Bobrow et al. (1986) stem from the fact that the shallow knowledge will fail in certain situations. Whether or not this is a fundamental limitation of expert systems depends on whether or not a formal model of the application domain is available. In the specific case of MYCIN, the lack of formal domain model means that a heuristic based approach is necessary.

5.2.3. Structural knowledge

This type of knowledge consists of the relations that hierarchically abstract data and hypotheses. One way to view structural knowledge is as a set of meta-rules which make all the inferences behind the heuristics explicit. As an example, in Figure 4, structural knowledge allows one to make the chain of inferences linking the datum, WBC < 2.5, to the conclusion, compromised host. As mentioned previously, structural knowledge is not explicitly represented within MYCIN. This is an important limitation, not only for GUIDON's purposes but also for decision support purposes as well. As Clancey (1983) states, structural knowledge provides a top-level explanation of a rule which in turn can serve as a constraint for how the rule should be generalized, or more importantly, when it should be broken. In fact, this is one of the hallmarks of expert performance: an expert knows when to violate a rule because he can reason about the rule's justification. Thus, it is imperative that structural knowledge be explicitly represented in the system in order to support the decision maker in deciding if the advice provided by the heuristics should be over-riden or not.

It should also be mentioned that the absence of structural knowledge is not a limitation inherent in the expert system approach, but rather a limitation of MYCIN. MYCIN is a f lat system of rules. As such, it can only describe its current inf erencing steps, and it cannot explain them on any level of detail. The purpose of embedding structural knowledge in a system is to provide justifications for the rules. Explanations of rules provide levels of detail by referring to more general concepts. Thus, one way to represent structural knowledge in a system is as a generalized rule. In fact, as shown in the example in Figure 4, what is needed is a tree of rules. But how many levels of abstraction are sufficient? In order to provide the necessary bridge between the user's knowledge and the knowledge represented in the computer, the hierarchy should go to a level of abstraction that connects to a pattern of

reasoning that the users have encountered before, i.e., premises that they readily accept (Clancey, 1983).

5.2.4. Strategic knowledge

The third and final type of knowledge discussed by Clancey (1983) is strategic knowledge, which is concerned with planning activities. Typical considerations are where to focus the search and also determining in which priority goals should be pursued. Such considerations are important in human reasoning but in MYCIN they are completely arbitrary: the order in which goals are addressed is determined by the order in which the rules were entered into the system! Thus, it is not surprising that students found it difficult to understand GUIDON's problem solving strategy. The basic lesson is that it is not sufficient to know all the rules in order to understand the program's reasoning. One must also have a plan for which rules to apply and in what order. Because MYCIN applies rules exhaustively, it has no explicit plan.

Just as with structural knowledge, strategic knowledge can be represented in the system as meta-rules. An example for a medical System would be: if there are unusual symptoms, then pursue those first. It is important to develop a rational planning strategy (i.e., on that people can understand), and to make it explicit.

Again, this will support the user in identifying situations in which the expert system should be over-riden. Just as with the absence of structural knowledge, MYCIN's lack of strategic knowledge is a limitation that can be overcome, not one inherent in the heuristic classification approach.

5.3. Implications for Process Lontrol

There is an interesting correspondence between the three types of knowledge described by Clancey and the framework for CTA described earlier. The relationships are illustrated by comparing Figures 1 and 3. Structural knowledge is that knowledge required to perform either state identification (abstracting from measured data to a functional description of the current state of affairs) or planning (specifying an intention in terms of a sequence of acts). Support knowledge is the knowledge required to make a decision based on first principles. In effect, it maps functional states onto appropriate intentions to be carried out. This mapping can take place at any of the various levels of abstraction. The f inal epistemological category, strategic knowledge, represents knowledge concerning intentions about which goals to pursue in the search for an appropriate decision. The distinction between this category and the previous two is similar to that between product and

process criteria. Whereas strategic knowledge describes the goals to be obtained (product criteria), structural and support knowledge decribe how those goals will be obtained (process criteria).

This epistemological description of expert systems allows us to extract some recommendations for providing computerized advice for process control systems. In doing so, we have to distinguish between two cases, state identification and decision making, since the support requirements for each of these will be different.

5.3.1. State identification

In general, advice in this task is typically relevant only in the infrequent, unknown situations. In this case, diagnosis is a critical task which cannot be based on advice that is inherently unreliable, as is the case with operators' heuristics or designers, foresight. The best solution is to supply the operator with the result of consistent engineering analysis of the state of afairs in the plant based on deep knowledge, i.e., basic physical principles, and with a proper presentation of information, as discussed in the previous section. This analysis can be a complex and time consuming process, and heuristic rules may be applied to set priority on different available approaches to the analysis, without endangering the consistency of the results. More work is needed to be able to formulate the most effective combination of engineering analyses and heuristics.

When formal models of the process are not available, a classical expert system approach based on shallow knowledge must be adopted. In these situations, several considerations should be taken into account. All of these follow from the fact that this approach is based on uncertain inferences, implying that there will be situations for which the heuristics are not appropriate. It will be up to the user of the system to recognize these situations, and use his domain knowledge and adaptive ability to deal with them effectively. In order to successfully take on such a flexible role, the user must be provided with strategic and structural knowledge. Thus, such knowledge should be explicitly represented in the system and made available to the user, so that he can cope with novel situations (Woods, Roth, and Bennett, 1988).

The need for structural knowledge is most obvious in rule-based state identification. In this case, rules provide a mapping from observable cues to the appropriate action to be taken. If this is performed directly then data integration, goal setting, and decision making are combined into one function (see Figure 1). This is equivalent to MYCIN's form of direct diagnosis, as shown in the rule in Figure 4. As was found with GUIDON, this cognitive ar-

chitecture will result in an 'opaque' system that may not support operator understanding. This in turn implies that the potential for error recovery is greatly reduced, as mentioned above. In order to avoid this problem, an intermediate stage of determining the functional state of the system is necessary. The implication for design of DSS is that even rule-based DSS for diagnosis should incorporate the bridging level of diagnosis in functional terms. By making the intermediate steps available to the user, understanding and potential for error recovery are greatly enhanced. In Clancey's terms, this is equivalent to making the structural knowledge explicit and available to the user (cf. Hollnagel, 1987; Lehner and Zirk, 1987). Hudlicka and Lesser (1987) describe an automated diagnosis system that adopts this strategy at several levels of abstraction.

5.3.2. Decision Making

A much more promising use of AI technology appears to be the use of complex knowledge-based systems to supply the operators' with design basis information; not only factual information about system anatomy and function, but also the reasons and intentions behind operating procedures and practices, interlocks, and automatic control systems. An efficient query and advice system in this domain can be more trustworthy than a system based on heuristics. The presently available AI tools make such a system practically feasible, but only if the necessary information can be obtained from designers. Unfortunately, much of the design basis depends on subjective choices of designers, who may no longer be accessible, or on implicit industry or company practices.

Regardless of the activates they are intended to support, advice giving systems should be designed so that the operator can verify decisions by his own preferred decision strategy. This means that all of the knowledge that the computer uses to provide advice should be explicitly represented in the system, and made available to the operator. Expert systems designed according to this principle have already been proven to result in better performance (Lehner and Zirk, 1987). The point of designing 'transparent' decision support systems has already been made with reference to automatic systems, but it is just as important for advice giving systems.

Finally, the possibility of allowing the user to select which alternatives are pusued and in what order (the role played by the strategic knowledge in expert systems) should also be considered. This type of support attempts to take advantage of the operator's domain knowledge and adaptive creativity by allowing him to direct the computer's resources to the goals that seem to be most promising or urgent. The capability to direct the computer's focus of attention is a very advanced form of decision support that is relatively un-

explored (but see Pople, 1985 for an example) Thus, little is known about the potential problems that can arise. In principle, however, this form of support provides the operator with the capability to redirect the computer in situations where it deviates from the correct Solution path. This is an important feature since machine problem solvers may sometimes take long inefficient detours before turning to the correct solution path, or can get stuck in one path and never reach a solution (Pople, 1985; Roth, Bennett, and Woods, 1988).

5.4. An Alternative Approach to Providing Advice

The preceding discussion has illuminated the limitations of existing expert systems and how some of these can be overcome. The recommendations that were proposed should result in DSS that are more compatible with the characteristics of the problem and the human operator. In spite of this, the design space of possible DSS is excessively constrained by the expert systems perspective. In this section, we suggest a new perspective for interpreting the demands encountered by the process operator which leads to a recommendation for a different form of computerized decision support.

Due to their inherent complexity, process environments will inevitably present operators with control situations they have never before experienced. The rare but potentially catastrophic fault is the prototypical example. Because the particular characteristics of any fault are unique, the operator is faced with the problem of having to cope with unanticipated variability. He cannot rely on the designer's foresight nor a computerized problem solver to provide him with a solution to his problem. In fact, the operator is faced with a control engineering design problem that he must solve on line in real time. Clearly, this is a very challenging problem, one that requires considerable support to be handled effectively. To obtain a solution, the operator must consider various alternatives, and their probable outcomes, before deciding what the best course of action is.

A fundamental obstacle to effectively carrying out this role is the selection of an appropriate stop-rule. In carrying out their problem solving activities, operators are supposed to ensure that the control task they intend to effect is the correct one by conceptually testing their hypotheses before they act. But when should the operator stop thinking and start acting? Designers have a variety of tools available to them for testing hypotheses (e.g., simulators, computers, laboratory experiments, textbooks, etc.). Operators, on the other hand, only have their experience and the plant. At some point or other they have to test their hypotheses by actually manipulating the plant. Therefore, truly effective decision support should give operators the means for testing hypotheses with the same powerful and reliable tools that design-

ers have access to, not in terms of the solutions prepared by designers, since these will necessarily be incomplete, nor in terms of an lexpert's, heuristics, since these are fallible. A similar point has been made by Woods (1988), who suggests that the most valuable form of decision support that operators could be provided with is the capability to conceptualize the problem domain in various ways. Clearly, very little is known about how to provide this form of support. The topic is one that is in urgent need of research since the benefits of such powerful tools are likely to be great.

6. CONCLUSIONS

In this paper, we have outlined a taxonomy of decision support techniques. It was shown that the appropriateness of each category of support is dependent upon the characteristics of the problem to be solved. We have argued that CTA can be used as a methodological tool to match the demands of the problem domain to the capabilities of the three decision making agents: designer, operator, and process computer. The remainder of the paper was concerned with discussing the suitability of each category of decision support for the various problem solving tasks that are typically encountered in process control, as well as the important considerations that should be taken into account for each category. This general discussion of how to determine the cognitive architecture of a DSS for process control suggests some principles for maximizing the match between the demands of the task and the capabilities of the human operator. The end result should be the design of truly effective and reliable industrial systems.

ACIKNOWLEDGEMENTS

The authors would like to thank Erik Hollnagel for providing comments on an earlier version of this paper.

REFERENCES

Beltracchi, L. (1987). A direct manipulation interface for waterbased rankine cycle heat engines. IEEE Systems-,- Man, and Cybernetics, SMC-17, 478-487.

Bobrow, D. G., Mittal, S., and Stefik, M. J. (1986). Expert systems: Perils and promises. Communications of the ACM, 29, 880-894.

Clancey, W. J. (1985). Heuristic classification. Artificial Intelliaence, 27, 289-350.

Clancey, W. J. (1983). The epistemology of a rule-based expert system - a framework for explanation. Artificial Intelligence, 20, 215-251.

Cohen, R. M. , May, J. H. , and Pople, H. E. Jr. (1987). An intelligent workstation for eletrocenter design. IEEE Transactions on SVstems. Man, and Cybernetics, SMC-17, 240249.

Cuny, X., and Boy6, M. (1981). Analyse semiologique et apprentissage des outils-signes: L'Apprentissage du schema d'electricit6. communications, 33, 103-140.

Fink, P. K. , and Lusth, J. C. (1987). Expert systems and diagnostic expertise in the mechanical and electrical domains. IEEE Transactions on Systems,, Man, and Cybernetics, .EMC-17, 340-349.

Hoc, J.-M. (1987). Analysis of cognitive activities in process control for the design of computer aids. In H.-J. Bullinger and B Shackel (Eds -) , Human-Computer Interaction - INTERACT 187 (pp. 257-262). Amsterdam: North-Holland.

Hollnagel, E. (1988). Issues in knowledge-based decision support. In G. Mancini, D. D. Woods, and E. Hollnagel (Eds.), Cocrnitive encrineerincf in @4@mi@c worlds. New York: Academic Press.

Hollnagel, E. (1987). The complexity of man-machine systems. In C. G. Hoyos and B. Zimolong (Eds.) , Ingenieur-psychologie. Gottingen, F. R. Germany: Verlag fur Psychologie.

Hudlicka, E. , and Lesser, V. (1987). Modelling and diagnosing problem-solving system behavior. IEEE Transactions on Systems,.Man, and Cybernetics, SMC-17, 407-419.

Lehner, P. E., and Zirk, D. A. (1987). Cognitive factors in user/expert-system interaction. Human Factors, 29, 97-109.

Mitchell, C. M., and Saisi, D. L. (1987). Use of model-based qualitative icons and adaptive windows in workstations f or supervisory control systems. IEEE Transactions on Systems.Man, and Cybernetics, SMC-17, 573-593.

Milne,, R. (1987). Strategies for diagnosis. IEEE Transactions on Systems. Man, and Cybernetics, SMC-17, 333-339.

Moray, N. (1986). Monitoring behavior and supervisory control. In K. R. Boff, L. Kaufman, and J. P. Thomas (Eds.), Eandbook of perception and human performance volume 2. New York: Wiley.

Nawab, H., Lesser, V., and Milios, E. (1987). Diagnosis using the formal theory of a signal processing system. IEEE Transactions on Systems, Man, and Cybernetics, 17.. 369-379.

Pople, H. Jr. (1985). Evolution of an expert system: From Internist to Caduceus. In I. De Lotto andM. Stefanelli (Eds.), Artificial intelligence in medicine (pp. 179-208) .New York: Elsevier.

Rasmussen, J. (1987). Risk and information processing. In W. T. Singleton and J. Hovden (Eds.), Risk and decisions (pp. 109122). New York: Wiley.

Rasmussen, J. (1986). Information processing and human-machine interaction: An approach to cognitive engineering. New York: North-Holland.

Rasmussen, J., and Goodstein, L. P. (in press) . Information technology and work. In M. Helander (Ed.), Handbook of humancomputer interaction. New York: North Holland.

Rasmussen, J., and Rouse, W. B. (1981). Human detection and diagnosis of fy@item failures . New York: Plenum.

Rasmussen, J. , and Vicente, K. J. (1987). Cognitive control of h,uman activities and errors: Implications for ecological interface ALLiLgn -M-2660). Roskilde, Denmark: Riso (Riso National Laboratory, Department of Computer and Information Science.

Roth, E. M. , Bennett, K. D. . and Woods, D. D. (1988). Human interaction with an "intelligent" machine. In G. Mancini, D. D. Woods, and E. Hollnagel (Eds.), Cognitive engineering in Aynamic worlds. New York: Academic Press.

Rubin, K. S., Jones, P. M., and Mitchell, C. M. (1987). OFMspert: Application of a blackboard architecture to infer operator intentions in real time decision making. Manuscript submitted for publication.

Scarl, E. A., Jamieson, J. R., and Delaune, C. I. (1987). Diagnosis and sensor validation through knowledge of structure and function. IEEE Transactions on Systems, Man, and Cybernetics, SMC-17, 360-368.

Sheridan, T. B. (1987). Supervisory control. In G. Salvendy (Ed.), !Landbook of human Lactors (pp - 1243-1268). New York: Wiley.

Vicente, K. J. (1987). The role of information technology in emergency management: §Zpert system or cognitive instrument? (Riso-M-2664). Roskilde, Denmark: Riso National Laboratory, Department of Computer and Information Science.

Vicente, K. J.., and Rasmussen, J. (1988). A theoretical framework for ecological interface A2LiLgn (Riso Technical Report in preparation). Roskilde, Denmark: Riso National Laboratory, Department of Computer and Information Science.

Woods, D. D. (1988). Commentary: Cognitive engineering in complex and dynamic worlds. In G. Mancini, D. D. Woods, and E. Hollnagel (Eds.), Cognitive engineering in dynamic worlds. New York: Academic Press.

Woods, D. D., Roth, E. M., and Bennett, K. (1988). Explorations in joint human-machine cognitive systems. In W. Zachary and S. Robertson (Eds.), Cognition, computing, and cooperation. Norwood, NJ: Ablex.

Risø National Laboratory **Risø – M – 2696**

Title and author(s)

THE COGNITIVE ARCHITECTURE OF DECISION SUPPORT SYSTEMS FOR INDUSTRIAL PROCESS CONTROL

Kim J. Vicente and Jens Rasmussen

Date March 1988

Department or group Information Technology

Groups own registration number(s) R-9-88

Project/contract no.

Pages 42 Tables 1 Illustrations 4 References 30

ISBN 87-550-1405-4

Abstract (Max. 2000 char.)

There are two properties of process control environments that create a need for effective computerized decision support. First, the fact that the system is usually quite reliable means that faults are relatively infrequent. Consequently, operators have great difficulty in dealing with these situations because they are so unfamiliar to them. To make matters even worse, it is under those very same rare, abnormal conditions that the risk of endangering system safety is greatest. Decision support systems are being built to help operators cope with these demands. As a result, the operating staff and the computerized control system are involved in a complex cognitive system, in which several different task allocation policies can be chosen. This paper discusses how to determine what the cognitive architecture of the decision support system should be in order to achieve effective and reliable performance. A taxonomy of decision support techniques is proposed, and the appropriateness of each for the various problem solving tasks that are typically encountered in process control is discussed. The considerations associated with the presentation of evidence in the form of a problem representation and the presentation of expert advice are treated in greater detail.

Descriptors

Available on request from Risø Library, Risø National Laboratory, (Risø Bibliotek, Forskningscenter Risø), P.O. Box 49, DK-4000 Roskilde, Denmark.
Telephone 02 37 12 12, ext. 2262. Telex: 43116, Telefax: 02 36 06 09

RR950000 99

Available on request from
Risø Library,
Risø National Laboratory, P.O. Box 49,
DK-4000 Roskilde, Denmark
Phone (02) 37 12 12 ext. 2262

ISBN 87-550-1405-4
ISSN 0418-6435