Operator Interface in Process Control: Introducing 3D Graphics and Virtual Reality into Decision Support Systems

Ivan Burmistrov

Moscow State University
Department of Psychology
HCI Laboratory
8-5 Mokhovaya Ul., K-9 Moscow, 103009 Russia
Phone/Fax: +7 095 203 3123
E-mail: ivan@ht.cogsci.msu.su

Abstract. This paper proposes an interaction style, the Flexible Script Interface (FSI), which is designed to recommend the correct sequence of actions appropriate in relation to the problem-solving tasks. FSI is an intelligent interaction style that is based on the script approach to problem solving and is implemented with an interactive script graph which represents a task structure and canalises user's problem-solving behaviour. The prototype system combines 3D computer graphics visualisation technology with decision support and cognitive engineering to produce an intelligent graphic interface for operators of real-time supervisory control systems, such as those used in power production and industrial process control. We describe the theoretical background, domains, architecture and implementation of the first proof-of-concept prototype.

Keywords: process control, human error, decision support systems, problem solving, intelligent interfaces, 3D visualisation, virtual reality, computer supported collaborative work.

1. Introduction

The progress of technology has eliminated many of the technical limitations regarding the use of computers in complex work situations. In parallel with this, work planning in many domains has changed because the workers (users) today are responsible for a broader, more integrated and more complex part of the work situation in which problem solving and decision-making situations are central characteristics (Carstensen and Schmidt, 1993). Examples of complex work domains are aircraft industry and process control in advanced manufacturing.

The design of industrial control centres is advancing toward totally computer-based man-machine interfaces. Computer based interfaces offer many potential advantages over traditional hardwired control panel interfaces including greater flexibility regarding the type of data displayed and its presentation. However, achieving this potential requires development of new interface concepts that change the way operators interact with the plant.

Current theories and guidelines of human-computer interface design give little attention to users' dynamic problem solving process and strategy. This may not present a big problem for users in small tasks, but when a task is sophisticated and requires the support of massive, various information and system functions at different stages of problem solving, the compatibility between the interface design and users' problem solving strategy becomes crucial (Ye and Salvendy, 1993).

According to Norman (1983), the interface should be designed to allow the individual user to: (1) execute an action by more directly expressing high-level intentions; and (2) receive direct help in planning how carry out the task. This paper proposes an interaction style, the Flexible Script Interface (FSI), which is designed to recommend the correct sequence of actions appropriate in relation to the actual task. FSI is an intelligent interaction style that is based on the script approach to problem solving and is implemented with an interactive script graph which represents a task structure and canalises user's problem-solving behaviour. The main ideas implemented in the FSI are the following:

(A) Representation of a script in the form of graph directly on-screen. Thus, a user is provided with spatial cognitive map of the task world while solving a problem.

(B) Making it possible to interact with the system via such a graph. Thus, FSI gains advantages of direct manipulation.

(C) Dynamic visualisation of script transformations as immediate feedback to user's actions and data processing outcomes.

(D) Giving a user on-line assistance with the task by the problem guide. The problem guide is an intelligent basis of the interface, which uses procedural expertise about standard scripts to support a user in realisation of optimum performance during his problem-solving activity.

In this paper, we will discuss theoretical basis for FSI and try to predict those benefits that this interface could provide for its user, as well as present a mock-up of 3D version of FSI aimed at supporting collaborative problem solving in the virtual reality environments.

2. Optimising interface design for process control industries

Historically, the operator was able to directly observe and manipulate the tool. Today, the operator often interacts with a "model" of the process, his commands being transferred by computer systems, sensors and effectors. The lack of system transparency thus becomes an obstacle for the process-skilled operator and the operator's knowledge and overall appreciation of the state of the system could be impaired. The manufacturing process therefore again should be made "visible" to the operator, using computer technology (Stahre, 1993). The problem is how to present information on computer screens in an intelligible and useful form, when the system is hidden from view.

As technology driven systems become more sophisticated the operator's role in system control also becomes increasingly important, even though they become less active in the control process – in highly automated systems human operators are frequently left to cope with the unexpected. Unexpected situations explicitly require a form of mental processing that is deliberate and, therefore, effortful. Thus it would seem axiomatic that information displayed to operators should facilitate "automatic" tasks and support effortful mental processes. However, the display design philosophies of many current industrial processes do not appear to reflect the needs of the operators for tasks of supervisory control and detection and diagnosis of system failures (Lawton, Morrison and Lee, 1993).

Process control remains a complex cognitive activity. This demands special requirements for a human interface. It must support the operator's attention, comprehension of the system functioning and control activities. Attention is especially important if operators are unaware of the current system state, therefore, the operating environment must bring all important issues to the operator's attention. Comprehension is constrained by the operator's knowledge about the physical nature of the processes under control. To overcome this constraint, the operating environment must provide adequate representations as well as domain-specific abstractions, and must give efficient support information.

Design decisions about the forms and specifications of computer assistance for the process control operators are also decisions about the kinds of human error that are possible and will occur. A vital aspect of human-machine interface concerns the detection and prevention of human errors, and minimising their consequences (Stager, 1991).

Contemporary approaches to human reliability applied in the process control industries address the distinction between simple errors (such as errors of omission, errors of selection, and errors in task execution) and errors involving higher level cognitive processes (e. g., inadvertent selection of the wrong control from a bank of controls, skipping a step in a procedure by mistake, confusion regarding system response, or inadequate problem solving) (Gertman, 1991). Rasmussen (1986) proposed a hierarchy of human performance which includes the following levels:

(1) knowledge-based level of performance that typically deals with unfamiliar situations and involves the conscious formulation of plans directed toward a goal,

(2) rule-based performance that is based on the rules obtained during work experience, diagnosing of work environment states and association of rules with those states, and

(3) skill-based level of performance that represents highly routinised activities that usually occur without conscious control.

All of these three performance levels are not free of human errors, although the nature of these errors is different for each type of performance (Rasmussen, 1982; Reason, 1990; Sharit and Chen, 1993). While knowledge-based performance may become erroneous due to the lack of appropriate knowledge structures that could govern problem-solving activity, errors in rule-based behaviour occur when the operators automatically apply normally highly efficient strategies in an unsuitable context or in an inappropriate way simply because they were successfully, frequently or recently used in past circumstances (Masson and De Keyser, 1993).

In this paper we propose a novel approach to the interface design for process control industries aimed at support of operators' problem-solving activities and prevention of human errors at all the above-mentioned performance levels. The interface is based on a script approach to human problem solving and uses spatial representation of the procedural knowledge.

3. Related work

Since our main focus is the development of novel user interfaces, we do not consider here different theoretical approaches to operator support that might lay behind the interface development (e. g., expert systems or operators' knowledge-based decision making models), but only describe two modern types of user interfaces aimed at operator support which are different from conventional interaction styles.

Many operator support systems are based on the hypertext and hypermedia interfaces that incorporate expert knowledge of the system (e. g., Kaarela, Huuskonen and Jaako, 1993; Stahre, 1993). The authors of such systems believe that network nature of hypertext is similar to operators' mental models and the manner of presentation (hypertext) appears quite natural to the users, since the presentation is similar in form to the models (semantic networks) and the domain (plant designs). There is a serious objection against this approach, however. The main problem with hypertext and hypermedia is that users that browse complex hypertext and hypermedia systems tend to get disoriented and lose their current location in these environments due to serious lack of navigational aids and absence of a high-level aggregation of information chunks. The likelihood that users will get disoriented while navigating the hypertext space increases in proportion to its interconnectivity (Nielsen, 1990), which in turn depends on the complexity of the system under control. Users have difficulties understanding where they are, how they got to their present location, where they want to be, and how to get there. This situation often occurs when a user activates a link accidentally and gets transported to an unfamiliar part of the network or to another network entirely (Batra, Bishu and Donohue, 1993). Frequently, in order to acquire a view of the entire landscape, users need to explore it sequentially in a piecemeal fashion. Yet, it is unlikely that process control operators will take the time to explore the entire landscape since their prime concern is to complete the task at hand (Lipner, Strong and Strong, 1993). Moreover, for a number of scenarios, especially those involving the real-time emergency management of risk, many workers will not likely have rapid access to such decision aids. In these situations the prospect of engaging in a dialogue with an expert knowledge based hypertext system would not be realistic. In our opinion, the applicability of hypertext interfaces to operator support in complex real-time tasks is highly questionable. However, these systems may be useful in operators' education and training, where hypertext and hypermedia might have an advantage over conventional plant documentation (cf. Sharit and Chen, 1993).

We are aware of only one attempt to exploit concurrent with FSI graphical notation implemented in a software system. This system has been developed by Retz and Hunger (1993) and is intended to support operators in diagnosing integrated circuits failures. The diagnosis of failed integrated circuits yet can not be fully automatically executed. The fastest solution can be found when all diagnosing tools are used in the right course and any error is avoided as far as possible. As the optimal flow of data is not constant but varies depending on the current failure problem, a corresponding help and support system results in drastic savings of time. Authors developed a procedural graphics interface that leads the operator through a diagnosing session and proposes the optimal session flow (see Figure 1). The whole test and diagnosis process is supported by a processflow management system which supports the operator in terms of decision making (and avoids herewith wrong decisions) and makes the user interface more transparent. A graphics interface supports the operator in terms of observation of process status, guidance through the system and decision help via predefined rules. The interface provides the operator with graph representation of necessary data and operations required to fulfil the task and uses colour coding to orient the operator in passing through necessary steps. The graphical representation can be scrolled and zoomed and each graph object can be activated by clicking the icon with the mouse. A serious advantage of the interface is that it is based on a formal process description language from which the interpreter automatically generates a processflow visualisation.

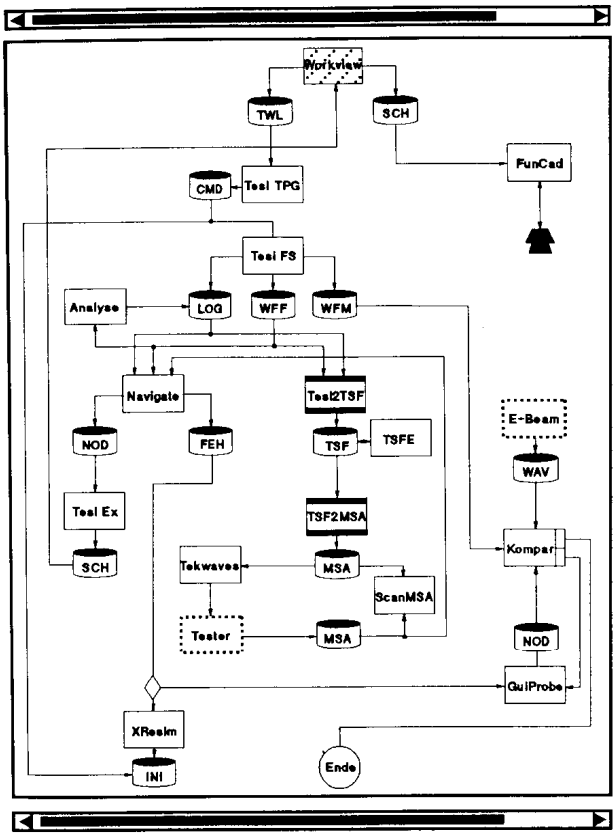


Figure 1. Visualisation of the processflow [from Retz and Hunger (1993)].

Unfortunately, authors do not discuss the cognitive background of their approach and how exactly their "complex tasks were divided into procedural subtasks",

even worse, they do not provide any empirical evidence on the usability or users' appreciation of the interface.

Nevertheless, Retz and Hunger interface was the most similar to FSI approach we could find in the literature. The main difference is that we pay more attention to the operator's mental models than to system designer's view of the task, give more freedom in the ways users can communicate with the system, and apply the cognitive economy principle via aggregation of task goals and steps into more comprehensible chunks.

4. Theoretical background

Generally, for the cognitive approach, human-computer interaction is seen as presenting problems which have to be solved. Human problem solving is guided by a person's understanding of the domain of information which the problem under resolution represents. Such understanding, which can be conceptualised as a mental model or schema, organises and directs a person's selection and usage of information in generating a solution. This paper describes an interaction style, based on the rubric of human problem solvers as builders and users of mental models. In this section, we will outline a number of theoretical assumptions which we have made on the basis of existing theoretical and empirical work in cognitive psychology and cognitive ergonomics. The discussion of differences between the various cognitive models could be the basis of a longer paper and will not be discussed here.

4.1 Mental models

The term mental model has found increasing usage in human computer interaction, although this term can mean many different things to different researchers. For instance, researchers say subjects form a mental model of a task through imposition of a spatial model (Eberts and Eberts, 1989), or by their goals, plans, and strategies (Johnson et al., 1988). For example, Clark (1987) cites Hammond et al. (1982) that state:

> "To learn and to use a complex interactive system, users have to form a mental representation of the information needed for operating on it. The better structured this information is, the more usable the system is likely to be."

Hammond et al. (1982) clearly distinguish between the user model, a body of knowledge existing in the user's memory, and the design interface image, a design concept acting both as an objective for the system development process and a reflection of the conceptualisation that an ideal user might hold. A user's mental model of a system should be an important determinant of performance and as well as a means of understanding why particular user errors occur. In particular, experienced users' models should be in closer agreement with the system than less experienced users' models, and deviations of expert models from the system should correspond to difficulties in performance and suggest ways that system usability could be improved (Kellogg and Breen, 1987). A good design interface image can be judged on whether it helps a user to build up a good mental model (Clark, 1987). Hammond et al. (1982) introduce a concept of "signpost", i. e. a task-to-action mapping rule. Clark (1987) infers that what makes for a good user model is having a set of non-confusable signposts when the user is lead through the design interface image's "maze", and stresses the critical role of good documentation and training material in formation of such signposts.

Script, or event schema, is a more elaborate concept of mental model. The concept of scripts has been used to explain text and story comprehension, as a way of understanding behavioural expectancies, and as a basis for designing artificial intelligence systems. Abelson (1981) defines a script as a hypothesised cognitive structure that when activated organises comprehension of event-based situations. A script represents stereotyped knowledge structure that describes appropriate sequences of goal-directed actions in a particular context, such as "having a dinner at a restaurant" or "attending a workshop".

Scripts consist from a number of scenes, which in turn are constituted by sequences of definite atomic operations. For instance, the restaurant script consists of "entering, ordering, eating, paying, and leaving". Each sequence of operations in scenes has a property of causal chain – every preceding action provides conditions for performing consequent actions.

In their Script-Based Information Processing model, Hershey et al. (1990) posit that scripts provide a framework that organises the set of operations leading to the solution of a problem. They hypothesise that experts, through experience, develop problem-solving scripts, which are streamlined over time so that unimportant variables are dropped from the set of operations. The expert's first step then, is to select the proper script for a particular problem statement. Once this has been accomplished, proceeding to a solution is simply a matter of applying the algorithms called for by the script.

In our opinion, system developers can efficiently improve user interfaces, if they would provide a non-expert user with expert's problem-solving scripts at early stages of user's communication with the system. Such scripts could help a user to form true signposts within the task world and facilitate the building of the good user model of the task. Usually users obtain such knowledge from the manuals or on-line help messages. In both cases the user has to interrupt the

mainstream dialogue with the system in order to obtain some guidance. In FSI this is not needed. One of the basic principles of FSI is to represent the task world in form of a script and allow the user to interact with this script directly on-screen. We hypothesise that such representation facilitates the formation of the true user model of the task and efficiently reduces user's misconceptions and errors during problem solving.

4.2 Spatial reasoning

As cognitive studies have repetitively proven, the representation of our knowledge is key to the effectiveness of our problem solving abilities. Many recent research indicate having the ability to make use of manipulation of spatial information, especially dynamic spatial displays, and, in particular, when users are provided with a display of the functional and procedural structure of a task, may make human-computer interaction tasks easier to perform (Eberts and Eberts, 1989; McCann, Taylor and Tuori, 1988; Fitzgibbon and Patrick, 1987; Patrick and Fitzgibbon, 1988). Computer graphics displays make it possible to display both the topological structure of a system in the form of a schematic diagram and information about its current state using colour-coding and animation. Such displays should be especially valuable as user interfaces for decision support systems and systems for managing complex processes (Kieras, 1992). Computer graphics provides an opportunity to present the user with an explicit representation of the task, resulting in facilitation of how the users think about the task, and it can allow the user to internalise the information which allows the user to solve novel problems, i. e. to form adequate mental models of the task (Eberts and Eberts, 1989).

5. Overview of the Flexible Script Interface

Some main design concepts and dialogue mechanisms used in the FSI are described below.

5.1 The conceptual model

Task representation model in FSI is similar to composition graphs (Inman, 1987) and Jackson Structure Diagrams (Jackson, 1983; Sutcliffe, 1988; Lim, Long and Silcock, 1992; Lim and Long, 1993). Our model also has some features of the dataflow diagrams, the transition networks (Benyon, 1990, 1995), Activity Chaining Diagrams (Bodart et al., 1995), and Task Oriented Flow Diagrams (Kirby and Heathfield, 1993). Some of these notations were developed to bridge the gap between system designers' models which are based on functional decomposition and procedural mental models of system users. However, the most congenial to FSI approach has been developed by Benysh and Koubek (1993) in their Procedural Knowledge Structure Model (PKSM).

The PKSM is a structural model of procedural knowledge which uses a visual representation scheme similar to that of the traditional declarative knowledge representations in order to bring out the structural and organizational aspects of procedural knowledge. The model may be visualized as a three dimensional pyramid which is sectioned into "floors" or levels (see Figure 2). Each of these levels contains a flowchart representation of the task steps. The difference between the flowcharts of each level is that with each descending level, the items in the chart are broken up into smaller and smaller sub-units or subtasks. Each individual subtask at one level will be broken up in to its constituent subtasks at the next level. This decomposition continues, level by level, until at the very bottom level, flowchart consists of basic task actions which cannot be further decomposed. Objects in the flow chart may be task goals (which will be further decomposed), basic task actions (which can be performed or executed), and decision nodes (which control the flow through the chart). In this same figure, basic task actions are represented by the ellipses and decision nodes by diamonds.

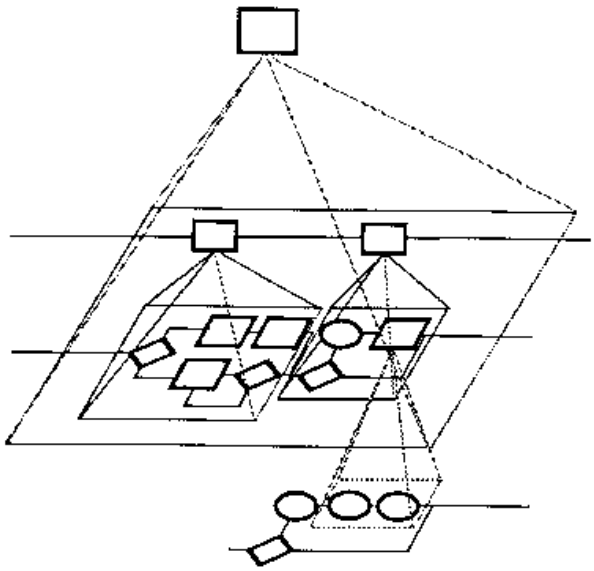


Figure 2. PKSM framework [from Benysh and Koubek (1993)].

Task goals are represented by rectangles. The highest level is the singular task statement or task goal. This goal will be subdivided into smaller goals, task elements, or decision nodes at the next lower level. These task actions and decision nodes, once encountered, will be perpetuated through to lower levels since they cannot be further decomposed.

The procedural elements are laid out in this flowchart type organisation. A horizontal chart displays the actions, computations, decisions, and outputs necessary in performing the task. Unlike traditional flowchart

though, it has the capability of simultaneously representing many different ways of doing the same task, as this figure illustrates. It is not restricted to one particular strategy and does not focus on any best way of doing the work. In this manner, it is able to model the various different approaches used by different people ranging from novice to expert level performance with differing experience and ability.

Flexible Script Interface is based on a similar representation of activities in a man-machine dialogue. In addition to conventional pure textual descriptions of conditions and sequences of actions, we developed a new form of script representation which is a network representation in form of the event graph. Each node of such a graph corresponds to a definite scene (procedural sequence of actions) and may have several entry and exit points. Arcs connecting nodes correspond to transitions from one scene to another (see Figures 3–11 for examples of FSI interface).

5.2 Script graph

The syntax of FSI graphs can be briefly and informally described as follows. There are three types of nodes and three types of arcs in FSI. A node may be:

(1) an atomic one; these nodes correspond to terminal level of nodes' hierarchy, which provides bonds with application part of a program;

(2) a composite node, or sub-script, which can be decomposed into atomic ones; these nodes represent medium level of aggregation in nodes' hierarchy, which is used to make interface representation more structural and better perceivable by the user, because this type of nodes allows organisation of groups of nodes into comprehensible chunks (in 2D version of FSI, the decomposition of such a node is visually represented as exploding pop-up window placed over the current node and containing graphical representation of an appropriate sub-script);

(3) a modifier node, a terminal node which initiates propagation of control information through the script graph; selection of such a node can reconfigure the graph by creating or destroying nodes or by blocking and releasing links between them.

The analogy with hierarchical menu system could make these concepts more clear. A script graph corresponds to a whole menu; atomic nodes correspond to terminal menu items, selection of which makes application to perform definite actions; sub-scripts are similar to submenus; and modifiers are similar to both setup items and those items of intellectualised menus that support item enabling-disabling mechanism (usually implemented as "grey-shading" of menu items, e. g. when "Paste" item becomes accessible only after "Cut" or "Copy" operation being performed).

Each node may hold several inputs and outputs; there is no isolated nodes in a graph.

Arcs, or links, which connect nodes and allow to move from one node to another may be:

(1) permitted,

(2) prohibited (blocked), or

(3) recommended.

The problem guide supervises user in his problem-solving activity and determines one of three above-mentioned possible states for each link. Prohibited links can change their status to permitted dependent on performing of defined prerequisite actions or as a response to activation of modifiers. User is free to choose transition through any permitted links, but usually only one of them would be marked as recommended by the problem guide. Current prototypes of FSI use colour coding (red for prohibited, yellow for permitted, and green for recommended) to represent the current status of a link in a script graph.

It is important to emphasise that the script is a generic set of operations. The actual problem-solving process requires that parameters from the current task be plugged into the problem-solving script (Hershey et al., 1990). FSI reflects contextual dependent parameters in on-the-fly transformations of the generic script graph via mechanism of adding-deleting nodes and blocking-releasing transitions between nodes, and thus, invokes relevant information and removes information irrelevant to user's current focus. FSI operates flexibly to meet different user needs and computing conditions.

5.3 Command input

Direct manipulation is the interaction style used in the FSI driven system. The FSI can support wide choice of input devices such as the function keys, the cursor control buttons, the mouse, the lightpen, the touch sensitive screen, the joystick, and the data tablet, which are useful in supporting spatial dialogues.

Jacob's (1989) view of direct manipulation interfaces is based on clear models of how the interface should operate. Direct manipulation interfaces may be seen as a solution of problems that arise in the conventional interfaces. These

problems typically stem from a failure to implement a task structure that is transparent to the user (Chignell, Hancock and Loewenthal, 1989). Direct manipulation user interfaces reduce a cognitive distance, the mental effort needed to translate from the input actions and output representations to the operations and objects in the problem domain. Another principal advantage of direct manipulation interfaces is that they decrease the demands on user's memory, because the system continuously displays its internal data, rather than requiring the user to remember them (Jacob, 1989). Moreover, the user has less information to keep in short-term memory about "where the system is" and fewer "places" it could be in. Jacob (1989) also states that ultimate success of a direct manipulation interface hinges on the choice of a good visual metaphor for representing the task world in terms of screen objects and input actions. It would be the difficult problem in the domains that involve abstract objects, which do not have a direct graphical image, such as a time sequence, hierarchy, or conditional statements.

5.4 Visual metaphor

The visual metaphor chosen in FSI is the graph representation of the event schema (script), which shows clearly what the user can do to move from one node (scene) to another, and emphasises the temporal and causal sequence of user and system actions in the dialogue. FSI permits the user to manipulate conditional statements (switches) and select actions and, as he does, it immediately redisplays the portions of the graph affected by these changes. We can adduce some plausible reasons to support our choice of the network visual metaphor.

The spatial representation of a script is "natural" in some sense. The natural language lexical collocations that are using to describe the area of problem-solving behaviour in both everyday and scientific speech, such as "to reach a goal", "to pose a problem", "to find a solution", "a deep study", "a decision space", etc., as well as extensive use of graphical notational tools in scientific representations of task structures, show that spatial metaphors are using widely to express features of problem solving. Thus, a script graph may provide user with an interface which could be akin to the hypothetical structure of his mental model of problem-solving process, and, therefore, could minimise the cognitive distance between the user's model of the task and appearance of the task that is implied by the interface.

Regarding the idea of that a spatial map of the task as well as the dynamic visualisation of changes in this map during problem-solving process could make interface more transparent to a user, it could be pointed out that the mechanism of path blocking and releasing has proved its intelligibility and usability in the area of labyrinth adventure video games, where preliminary performed actions such as "press a secret button" or "clear the room from the enemies" are used as prerequisites for successful passing from one compartment of a maze to another.

5.5 Guidance

Gritzman, Kluge and Lovett (1995) define the concept of guidance as that the user interface should at any given time be directed towards giving the user maximum help in choosing among a limited set of relevant possibilities to fulfil the task. User guidance is especially relevant to the operator's support in the continuous process industries. A multitude of possibilities offered by traditional human-computer interfaces is not what a user needs when confronted with a task in a complex use setting with many other phenomena requiring attention.

An intelligent constituent of FSI, the *problem guide*, uses procedural expertise about standard scripts and their permissible transformations to direct user's focus, to determine current subgoals, and to correct possible user's misconceptions. It controls dynamics of the script graph and uses mechanism of path blocking and releasing to inform user about consequences of his actions and choices and to direct his goal seeking behaviour. It also provides a user with on-line assistance by suggesting him the shortest way to the final goal through pointing out current subgoals.

5.6 Mixed initiative dialogue

Cochran (1992) in his view of control room user interface technology for the year 2000 suggests that future technologies would replace supervisory control with collaborative operations in which the system and the operator will share tasks associated with planning, conducting, and optimising operations. The FSI makes it possible to implement *various types of task sharing* between user and machine, from cases when user appears to be commandingly telling the machine what to do, to the higher levels of automation when machine uses script graph as a dynamic flow diagram to inform user about what system does currently. FSI can provide a framework for adaptive interfaces where machine regulates the workload of the human operator in a shared task.

Our approach to the user interface design is somewhere between traditional user interface design principles based on the object-action paradigm, which leaves the user as the initiator of all actions to be taken throughout the dialogue and present him virtually unrestricted set of possible actions with selected objects (e. g., Apple, 1987), and *what-you-see-is-what-you-can-do-right-now* paradigm (Gritzman, Kluge and Lovett, 1995), where the user is imperatively led through the task and the screen offers *only* actions appropriate in relation to the actual task while actions which are not crucial in a given situation in order to accomplish a certain task are not visible to the user. We believe the domain of the

supervisory decision making requires *mixed initiative* in the dialogue between the operator and the system where system initiative is reserved for the situations when things go wrong and the operator lacks procedural knowledge to prevent or correct faults.

The FSI has a reasonable symmetry of initiative in dialogue between human and computer provided by the mixed initiative approach. In the FSI, although the problem guide provides on-line assistance for the operator, he can refuse to act upon advice and is free in choosing any permitted trajectory through a decision space, because this interface does not determine a fixed hierarchy of operations to satisfy goals. As the system has incomplete knowledge about user's current needs, support is achieved by the system advising on possible ways of operation. The user can choose to take or ignore the system's advice, he is always in control. The system offers advise to the user, but never takes over.

In particular, this approach is also in concordance with the requirement that operators should not be forced to immediately intervene during an emergency and interfaces should provide time for reflection about critical decisions (Pew, Miller and Feehrer, 1981).

6. Current state of the work

Currently, we have two prototype systems implemented that exhibit some of the ideas presented above.

6.1 Domain selection

When designed FSI, we have hypothesised that this type of interface will be maximally efficient at facilitating the complex and/or novel tasks.

Chignell, Hancock and Loewenthal (1989) have suggested that the high level of the task complexity may be a preliminary criterion to warrant the development of intelligent interface. They have formulated the empirical notion of task complexity:

> "If users typically need a large amount of training and then continue to make errors after training, or generally fail to understand the model of the task presented in the interface, this indicates sufficient complexity to necessitate intelligent interface development."

Although our discussion is mainly based on mental models approach to human-computer interaction, the artificial intelligence oriented notion of problem solving, the problem space, will be adopted to conceptualise our understanding of task complexity.

The problem space model was formulated by Newell and Simon (1972). A problem space consists of a set of states, and a set of operators that are mappings from states to states. A problem is composed of a problem space together with an initial state and a set of goal states. Korf (1988) expresses the complexity of a problem in form of two parameters: the branching factor of the problem space, and the depth of solution of the problem. The edge branching factor is the average number of different operators which are applicable to a given state. The depth of a solution of a problem is the length of a shortest sequence of operations that map the initial state to a goal state.

In fact, this notation of task complexity is a topological one. Complex tasks have a lot of possible paths from initial state to goal states, and require a lot of operators to be performed to reach a goal. Clark (1987) also examines the topological aspect of task complexity and states that residual difficulties with human-computer interface after prolonged use may have more to do with the structure of the task itself, rather than the repertoire of mental information structures the user brings to the problem.

In our opinion, complex tasks require high cognitive loads on user's working memory, because they involve many different operators and possible states, and require long sequences of actions to be performed to arrive a goal. Such tasks are perplexing for both novice and expert users.

Task novelty has a profound effect on the information search and selection strategies of problem solvers, because novel tasks keep the subject's knowledge from playing a role in generating a solution and may force a subject into "weak" problem solving behaviour due to their elimination of the role of task or domain specific knowledge in performance (Hershey et al., 1990).

Since the main goal of FSI development was to assess its efficiency in facilitation of solving complex and/or novel tasks, we sought a domain that is sufficiently complex and novel to encourage adding more intelligence to the interface. Decision support systems were selected as meeting these criteria. Decision support systems are informationally and operationally complex: knowledge of the domain specifies what variables are relevant, and how they must be manipulated in order to achieve a correct solution. Additionally, decision making tasks can be analysed to derive a prescriptive sequence of "correct" steps by conducting a task analysis to identify the conceptual elements one need to consider, as well as the procedures one must follow in order to reach a correct solution.

6.2 2D version of FSI

Flexible Script Interface has been implemented for the PersoPlan (Personal Planning), a fuzzy decision support system based on psychological analysis of individual's motivation in decision making. This program is intended to assist user in multicriterial choices, helping him in comparison and evaluation of multiple desirable and undesirable consequences of some decision. PersoPlan can be applied to various decision choice problems, e. g. selection of goods, vocational decision, etc. User can adapt the program to the domain of his interest, viz. to create his individual *plan* (this term in the PersoPlan's context stands for what is usually named in software systems as "project") by determining his personal goals, motives for selection these goals, and means for reaching these goals.

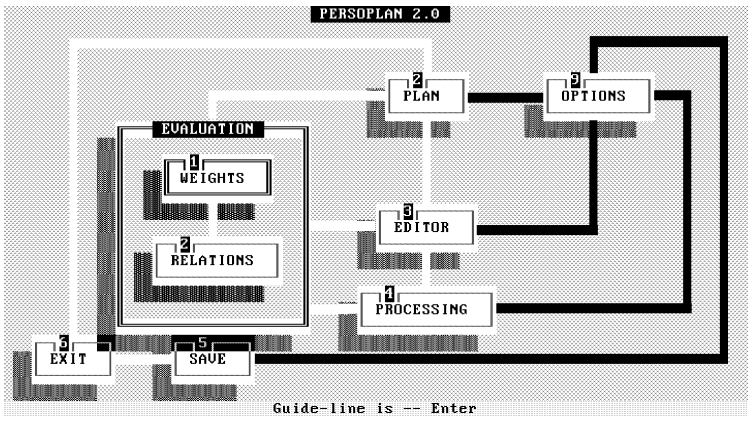


Figure 3. An example of PersoPlan interface [from Burmistrov (1992)].

The last version of the PersoPlan's user interface was restricted to a 2D interface with pop-up sub-script windows. Sample screen dump of PersoPlan's 2.0 interface is shown in Figure 3. This implementation of FSI has been described in detail in our previous paper (Burmistrov, 1992).

6.3 3D version of FSI

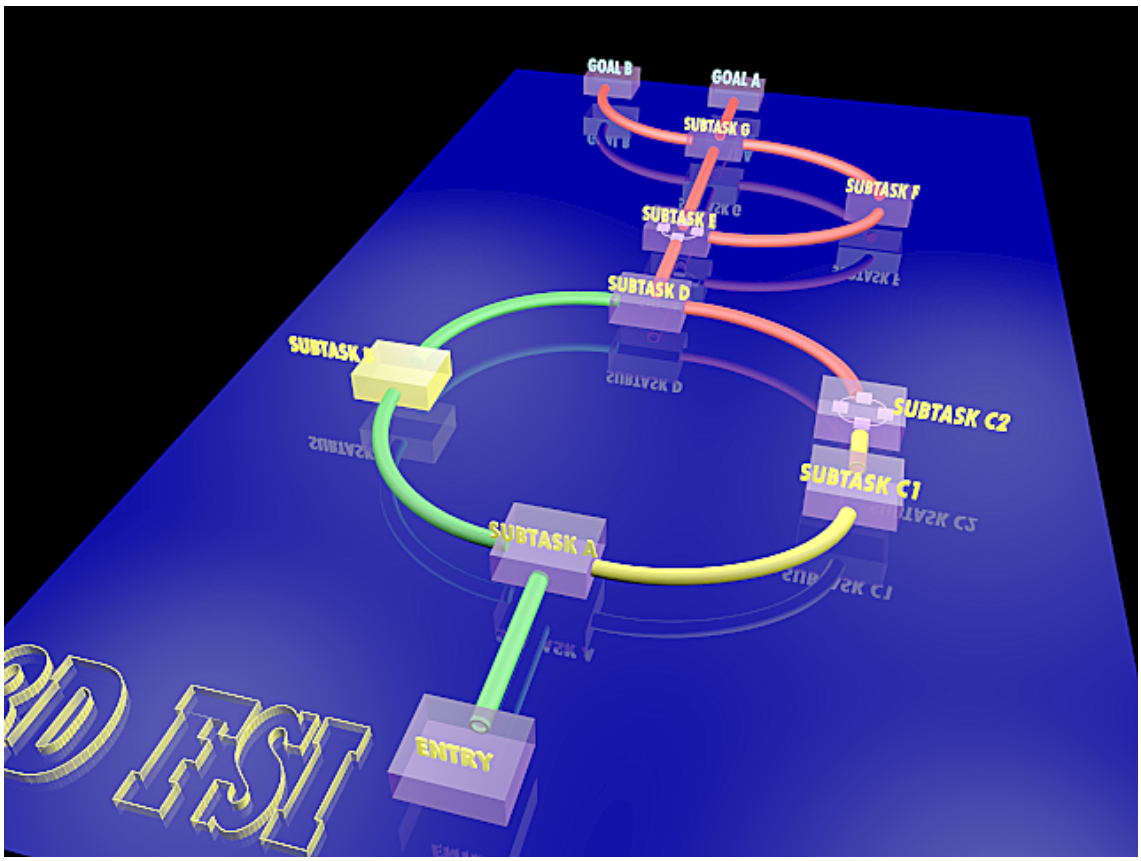
Our current work focuses on the development of the 3D version of the FSI which is mainly directed at supporting operators' decision support in process control industries. This section presents a mock-up of 3D interface which in particular is aimed at supporting collaborative problem solving in the virtual reality (VR) environments.

The definitions of Virtual Reality are wide and varied though all imply the same meaning, that of the illusion of participation in a synthetic environment rather than external observation of such an environment. Du Pont (1993) describes VR as "a set of computer technologies which, when combined, provide an interface to the computer with which the user can believe he or she is actually in a computer-generated world". Roberts (1993) states that, "Virtual reality can be described as the science of integrating man with information. It consists of three-dimensional, interactive, computer generated environments". But Warnecke (1993), in a rather hardware specific manner, asserts that, "Virtual Reality technology takes 3D data of a scene and presents this information as a stereoscopic computer generated image to the user equipped with an eyephone". Obviously, each of these definitions for VR has been influenced by the domain to which they are applied, the technologies used for constructing virtual environments, and, of course, the individual perspectives of the authors.

Recently, virtual reality has been applied to a wide range of problems associated with industrial maintenance and manufacturing. Most applications can be placed in one or more of six main categories: visualisation of complex data, controlling industrial robots, remote operation of equipment, enhancing communications, operations training, and virtual prototyping and design (Neugebauer, 1992; Warnecke, 1993; Sherrard and Narayanan, 1994; du Pont, 1993; Roberts, 1993). These applications are but an initial step in identifying opportunities for using VR to improve the competitiveness of manufacturing organisations. In particular, VR technologies can play a substantial role in the interactive design and control of complex production systems and processes. The three-dimensional nature of VR, and the mechanisms for interacting with objects in VR environments, makes the enormous amount of technical data available much more accessible to decision makers; travelling through, and manipulating objects within, the virtual facility offers a much more natural and direct interaction than working indirectly through programs providing only two-dimensional representations of the problem. 3D techniques may be more causal, direct and natural than 2D interfaces (Venolia, 1993).

Current technology is far from perfect in creating full illusion of participation in a synthetic environment though the present day systems offer a level of interaction and feeling of immersion in such a world that is sufficient for a wide range of applications. With future increases in technological capability VR holds the promise to provide significant improvements in many new application areas.

With VR technology, a human and a computer operate together as a combined system, one that is far more capable than either taken individually when performing a large variety of tasks. For example, the environment is particularly well suited to support reaction to unanticipated scenarios, ones that can exploit the corrective judgement of a person or of skilled teams of experts. VR must improve the manner in which decisions are made or improve the likelihood of



Fugure 4.

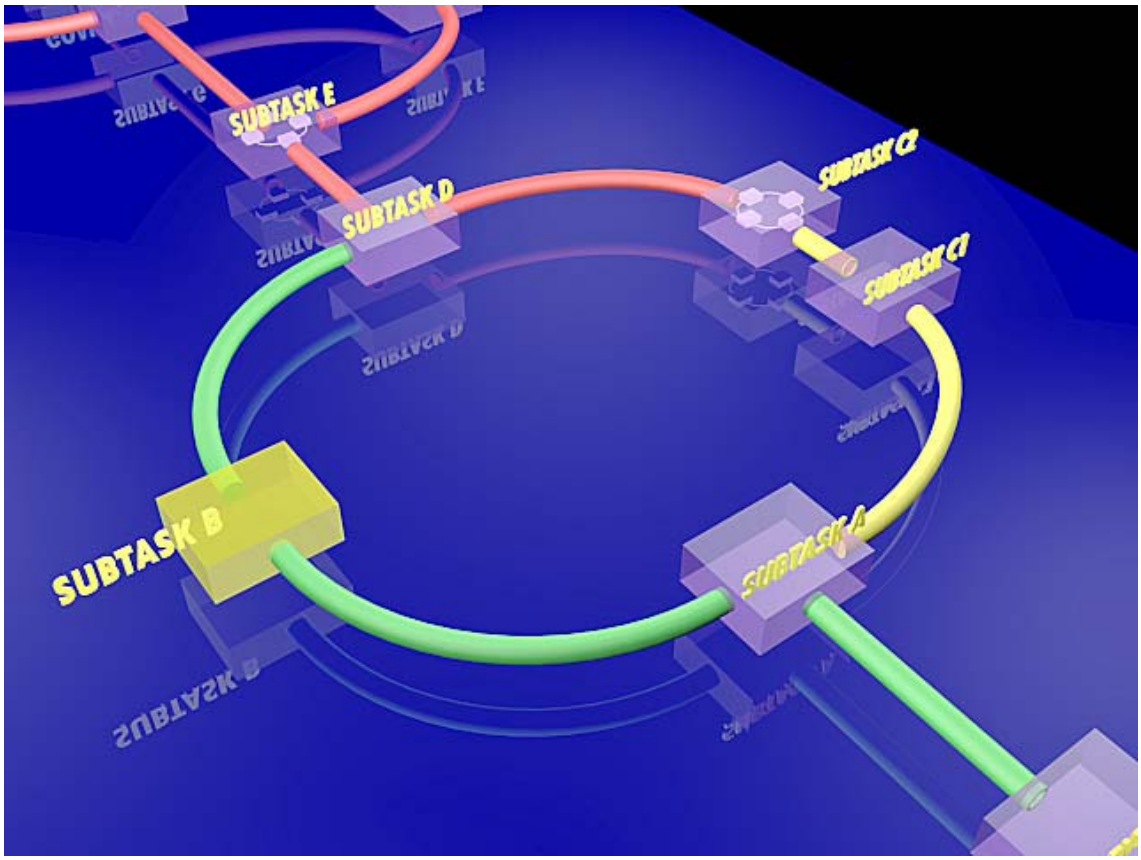


Figure 5.

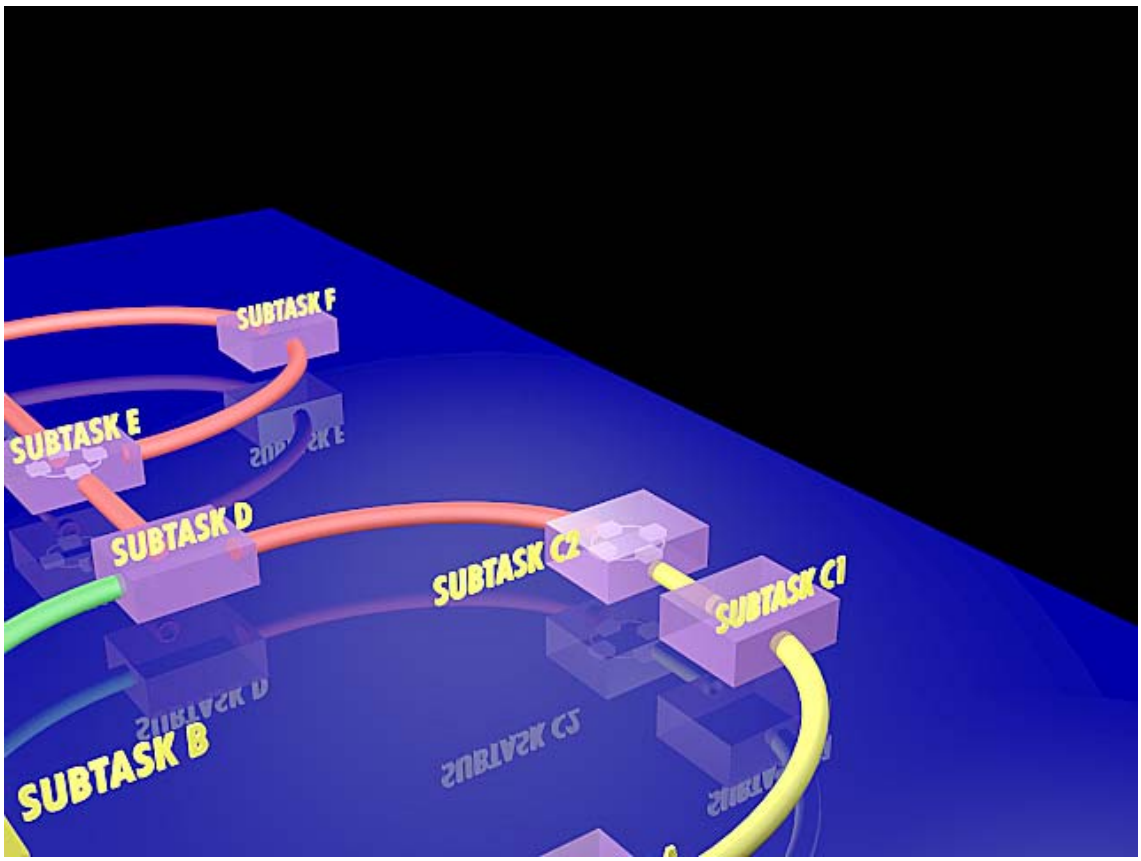


Figure 6.

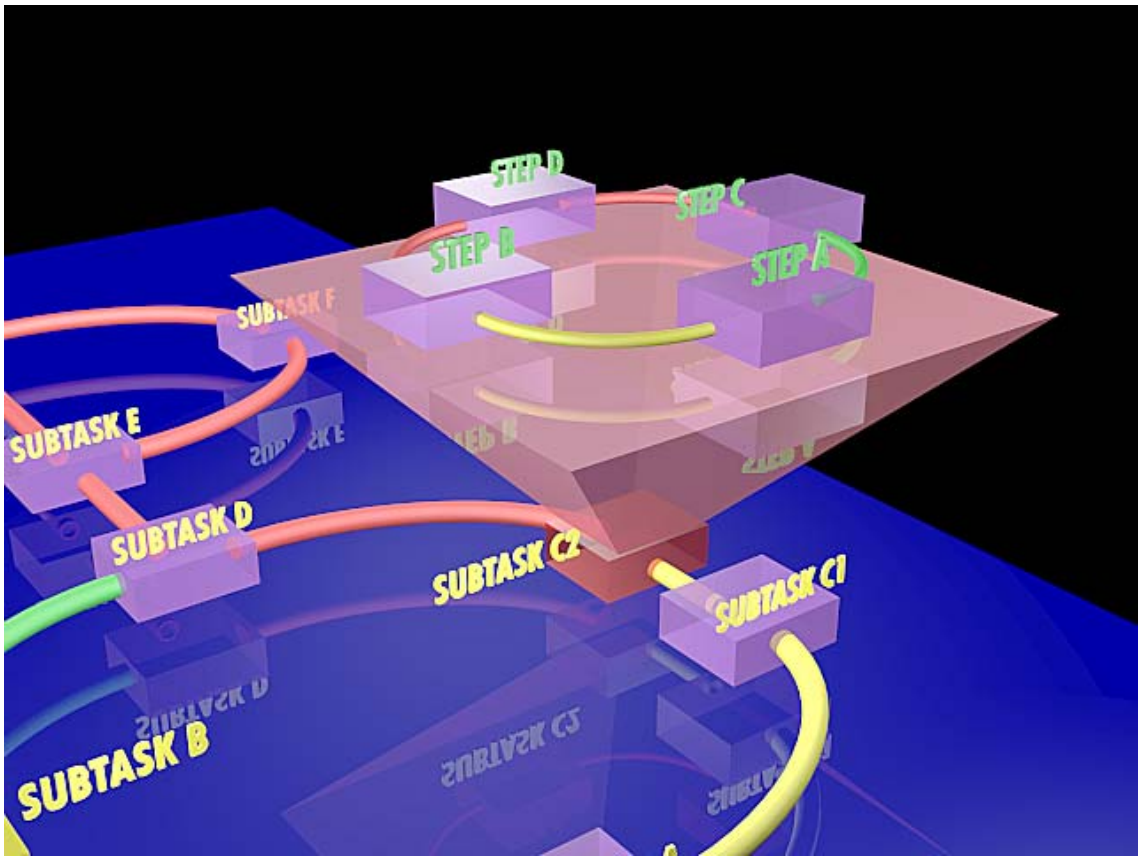


Figure 7.

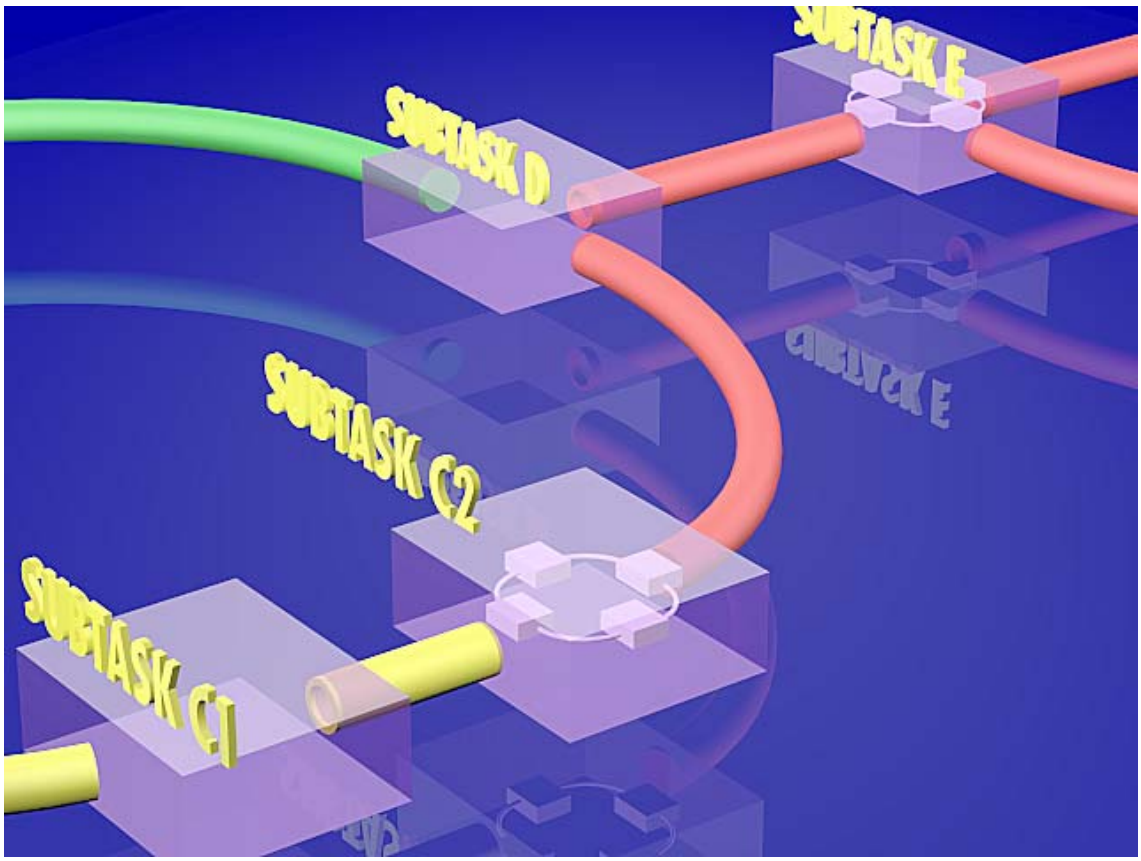


Figure 8.

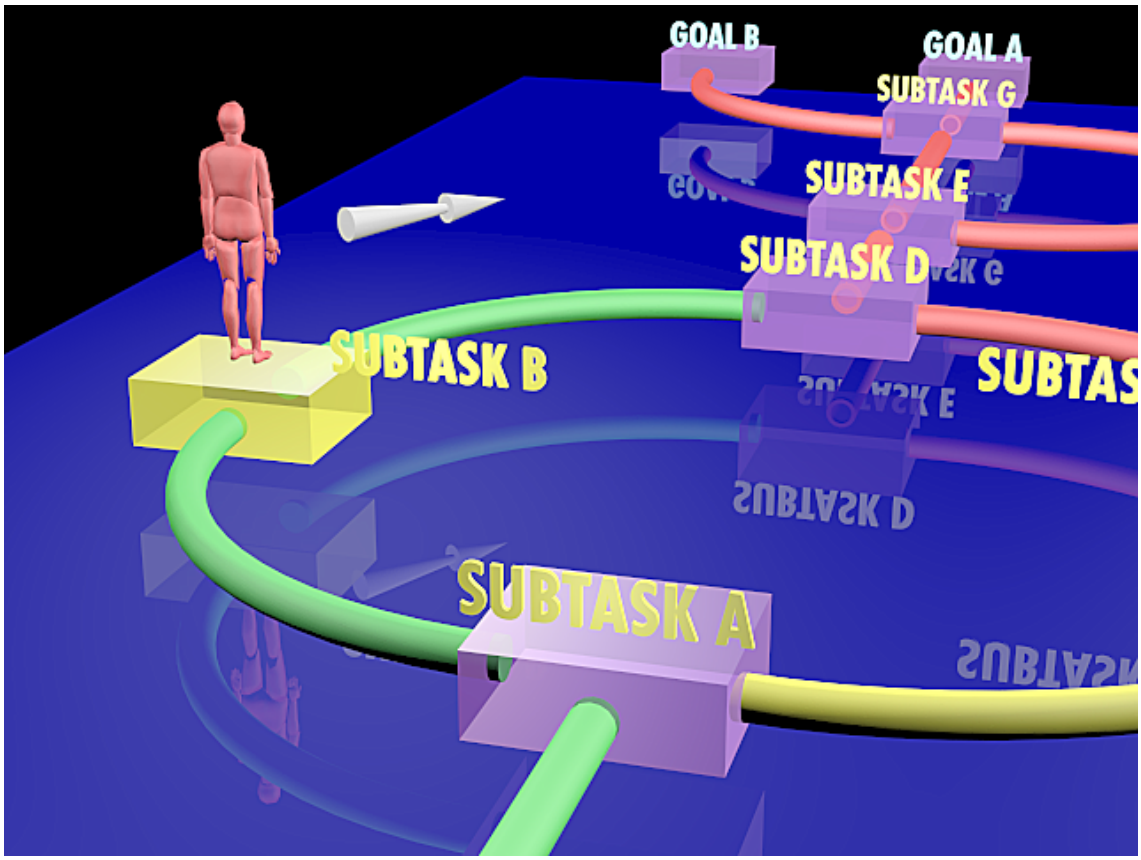


Figure 9.

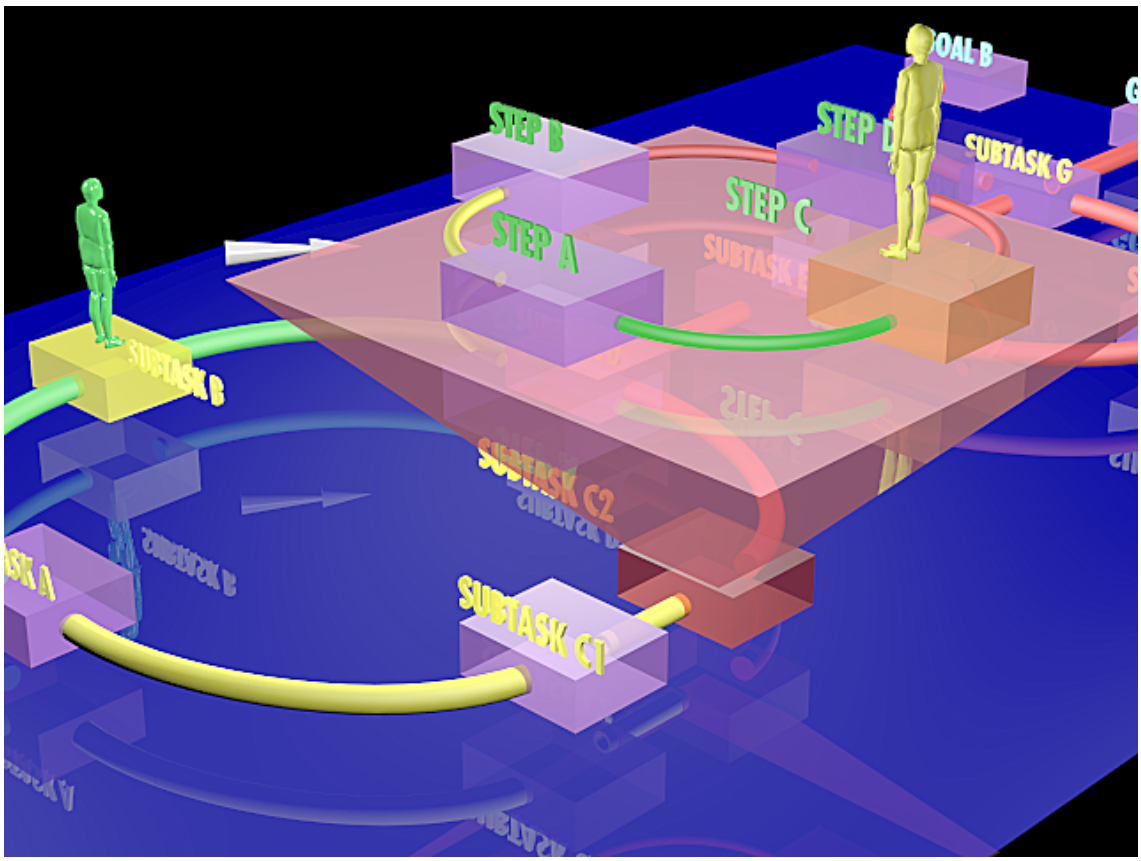


Figure 10.

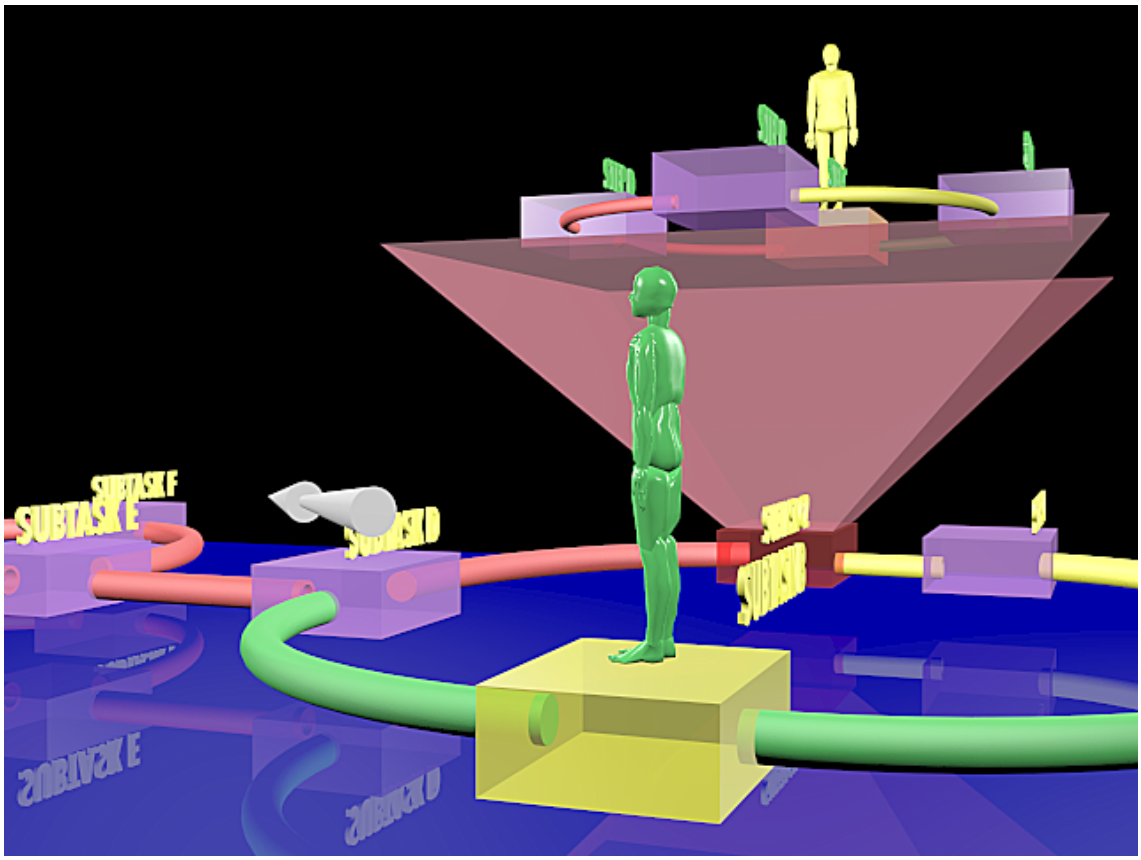


Figure 11.

making correct choices among alternatives if the costs associated with the technology are to be justified. We believe, however, that one of the most significant aspects of the technology is its ability to improve decision-making processes from both qualitative and quantitative perspectives.

Figures 4–11 show our vision of future decision support interface in process control. These pictures were rendered with presentation quality to provide the basis for discussion with and evaluation by the experts in the domain, software engineers and operators.

Figure 4 presents a bird eye view of the 3D graph with ENTRY node nearest to the viewer, two goal nodes – GOAL A and GOAL B – on the horizon, and a number of intermediate nodes and links between them. The graph represents the hierarchical structure of actions which are performed within the problem-solving activity and the sequence in which they are performed. Sequence generally flows forward throughout the graph from initial node (ENTRY node in the Figure 4) to end node(s) (GOAL A and GOAL B nodes). One of the main characteristic features of the FSI graph is that it does not present simultaneously the complete collection of actions and transitions, which are permissible in the system. We believe such complete representation would be useful for a system developer, but not for a real user who would be confused with a lot of irrelevant information. FSI is based on the principles of task context and cognitive economy in its representation of a task structure (cf. Gritzman, Kluge and Lovett, 1995). In FSI a user is faced with a predefined "standard" script graph of the top level of task structure hierarchy. This provides a user with a "general view" of a task structure (see Figures 4 and 5).

Graph nodes can be further decomposed into smaller procedural sequences of actions. Decomposition results in presenting new "floors" in the representation of the task world. Figures 6 and 7 show the decomposition of the node SUBTASK C2 into lower level sub-script of procedural actions, named here STEP A, STEP B, STEP C, and STEP D.

In order to provide visibility of possible decomposition of particular node, all the objects in the scene are rendered as translucent. For example, Figure 8 shows a close-up of the node SUBTASK C2 with an "embryo" of sub-script inside.

Figure 9 presents the concept of system guide whose recommendation is represented here as a flying arrow in front of the body icon.

3D FSI may naturally provide support for collaborative problem solving, allowing different team members to perform different actions in parallel in a shared simulated world. Team-mates are represented in the scene by body icons (avatars) of different colour, allowing each team member to get an overview of what is going on in other parts of the task world, to co-ordinate individual efforts, and to keep track of the progress in joint task performance (see Figures 10 and 11). Thus, 3D FSI is a computer supported collaborative work application.

7. Conclusion

HCI specialists, human factors and systems engineers must co-ordinate their responses to high-risk failures in modern process control industries. This paper has argued that concepts and approaches developed in the field of cognitive psychology can be used to guide interface design for process control systems.

This paper has presented an intelligent, direct manipulation interface for complex systems. We have developed a novel approach to the design of mixed initiative dialogues in the process control tasks. Commonly expressed opinion in the human factors literature on the interface design for control systems is that the information requirements for the two main tasks of the control room operator – supervisory control and fault management – are likely to be completely different, and that this should be reflected in the design of different interfaces for each condition (Hoonhout and Zwaga, 1993; Kaarela, Huuskonen and Jaako, 1993). The uniform interface of the whole process control activity could provide serious advantage in operator support, but today seldom found in existing process control environments. Due to its mixed initiative approach, the FSI may give a chance to system developers to provide a smooth transition among above-mentioned two modes of operation which may be covered by a consistent interface.

One of the most serious weaknesses of the FSI at the current state of its development is the absence of the formal specification languages for describing of system states, scripts, user interface layouts and dialogue rules. This deficiency would obviously impede our approach from implementing it into the real-life large-scale industry systems. The construction of an FSI-interface is far from trivial and requires thorough understanding of the processes under control and laborious knowledge elicitation procedures in order to acquire optimal expert scripts. The availability of formal notation tools for both coding of system states and appropriate operator's actions is of crucial importance for industrial-level implementation of the FSI solutions. Only limited and particular software implementations for specific cases (such as start-up, shut-down, mode change, or some special occasions) are now attainable. However, starting from these local tasks we could obtain important data on the usability of the interface designs and on the feasibility of our approach for more complex tasks.

Practical implementations could span the range from true virtual reality with stereoscopic immersion display helmets and gloves to a more modest evolution of 2D graphic user interfaces into 3D versions. It is important to note that VR systems can co-exist and communicate with more conventional systems (for example, this can be done in form of "augmented VR", the purpose of which is to blend conventional work environments and 3D graphics using head mounted partially transparent displays with a wide viewing angle) (cf. Ware, Arthur and Booth, 1993; Fahlén et al., 1993).

8. Acknowledgements

This work has been partly supported by INTAS grant #93-3493 "Prevention of Human Errors in Systems for Energy Production and Process Industry".

References

Abelson R. P. (1981). Psychological status of the script concept. *American Psychologist*, 36, 715-729.

Apple (1987). *Human Interface Guidelines: The Apple Desktop Interface*. Reading: Addison-Wesley Publishing Company.

Batra S., Bishu R. R. and Donohue B. (1993). Effects of hypertext topology on navigation performance. In: G. Salvendy and M. J. Smith (Eds.), *Human-Computer Interaction: Software and Hardware Interfaces. Proceedings of HCI International'93, Vol. 2*, pp. 175-180. Amsterdam: Elsevier.

Benyon D. (1995). A data centred framework for user-centred design. In: K. Nordby, P. Helmersen, D. J. Gilmore and S. A. Arnesen (Eds.), *Human-Computer Interaction: Interact'95*, pp. 197-202. London: Chapman and Hall.

Benyon D. R. (1990). *Information and Data Modelling*. Oxford: Blackwell Scientific Publications.

Benysh D. V. and Koubek R. J. (1993). The implementation of knowledge structures in cognitive simulation environments. In: G. Salvendy and M. J. Smith (Eds.), *Human-Computer Interaction: Software and Hardware Interfaces. Proceedings of HCI International'93, Vol. 2*, pp. 309-314. Amsterdam: Elsevier.

Bodart F., Hennebert A.-M., Leheureux J.-M. and Vanderdonckt J. (1995). Computer-aided window identification in TRIDENT. In: K. Nordby, P. Helmersen, D. J. Gilmore and S. A. Arnesen (Eds.), *Human-Computer Interaction: Interact'95*, pp. 330-336. London: Chapman and Hall.

Burmistrov I. (1992). Flexible Script Interface: An Intelligent Spatial Interaction Style. In: G. C. van der Veer, M. J. Tauber, S. Bagnara, M. Antalovits (Eds.), *Human-Computer Interaction: Tasks and Organization. Proceedings of ECCE 6*, pp. 219-229. Rome: CUD.

Carstensen P. H. and Schmidt K. (1993). Work analysis – perspectives on and requirements for a methodology. In: M. J. Smith and G. Salvendy (Eds.), *Human-Computer Interaction: Applications and Case Studies. Proceedings of HCI International'93, Vol. 1*, pp. 575-580. Amsterdam: Elsevier.

Chignell M. H., Hancock P. A. and Loewenthal A. (1989). An introduction to intelligent interfaces. In: P. A. Hancock and M. H. Chignell (Eds.), *Intelligent Interfaces: Theory, Research and Design*, pp. 1-26. Amsterdam: North-Holland.

Clark I. A. (1987). Designing a human interface by minimising cognitive complexity. In: H.-J. Billinger, B. Shakel and K. Kornwachs (Eds.), *Human-Computer Interaction: INTERACT'87*, pp. 101-108. Amsterdam: North-Holland.

Cochran E. L. (1992). Control room user interface technology for the year 2000: evolution or revolution? In: *Proceedings of the Human Factors Society 36th Annual Meeting, Vol. 1*, pp. 460-464. Santa Monica: Human Factors Society.

du Pont P. (1993). Applied virtual reality. In: *Virtual Reality in Engineering*, pp. 153-167. London: IEE.

Eberts R. E. and Eberts C. G. (1989). Four approaches to human computer interaction. In: P. A. Hancock and M. H. Chignell (Eds.), *Intelligent Interfaces: Theory, Research and Design*, pp. 69-127. Amsterdam: North-Holland.

Fahlén L. E., Brown C. G., Ståhl O. and Carlsson C. A. (1993). Space based model for user interaction in shared synthetic environments. In: S. Ashlund, K. Mullet, A. Henderson, E. Hollnagel and T. White (Eds.), *Proceedings of INTERCHI'93*, pp. 43-48. New York: ACM.

Fitzgibbon L. and Patrick J. (1987). The use of structural displays to facilitate learning. In: H.-J. Billinger, B. Shakel and K. Kornwachs (Eds.), *Human-Computer Interaction: INTERACT'87*, pp. 611-616. Amsterdam: North-Holland.

Gertman D. I. (1991). INTENT: A method for calculating HEP estimates for decision based errors. In: *Proceedings of the Human Factors Society 35th Annual Meeting, Vol. 2*, pp. 1090-1094. Santa Monica: Human Factors Society.

Gritzman M., Kluge A. and Lovett H. (1995). Task orientation in user interface design. In: K. Nordby, P. Helmersen, D. J. Gilmore and S. A. Arnesen (Eds.), *Human-Computer Interaction: Interact'95*, pp. 96-102. London: Chapman and Hall.

Hammond N. V., Morton J., MacLean A., Barnard P. J. and Long J. B. (1982). *Knowledge Fragments and Users' Models of Systems. IBM Hursley Human Factors Report No HF071*. Winchester: Hursley.

Hershey D. A., Walsh D. A., Read S. J. and Chulef A. S. (1990). The effects of expertise on financial problem solving: evidence for goal-directed, problem-solving scripts. *Organizational Behavior and Human Decision Processes*, 46, 77-101.

Hoonhout H. C. M. and Zwaga H. J. G. (1993). Operator behavior and supervisory control systems in the chemical process industry. In: M. J. Smith and G. Salvendy (Eds.), *Human-Computer Interaction: Applications and Case Studies. Proceedings of HCI International'93, Vol. 1*, pp. 109-114. Amsterdam: Elsevier.

Inman E. (1987). A syntax-directed graphics editor. In: H.-J. Billinger, B. Shakel and K. Kornwachs (Eds.), *Human-Computer Interaction: INTERACT'87*, pp. 119-124. Amsterdam: North-Holland.

Jackson M. A. (1983). *System Development*. Prentice-Hall.

Jacob R. J. K. (1989). Direct manipulation in the intelligent interface. In: P. A. Hancock and M. H. Chignell (Eds.), *Intelligent Interfaces: Theory, Research and Design*, pp. 165-212. Amsterdam: North-Holland.

Johnson C. W. (1995). Decision theory and safety-critical interfaces. In: K. Nordby, P. Helmersen, D. J. Gilmore and S. A. Arnesen (Eds.), *Human-Computer Interaction: Interact'95*, pp. 127-132. London: Chapman and Hall.

Johnson P., Johnson H., Waddington R. and Shouls A. (1988). Task-related knowledge structures: analysis, modelling and application. In: D. M. Jones and R. Winder (Eds.), *People and Computers IV*, pp. 35-62. Cambridge: Cambridge University Press.

Kaarela K., Huuskonen P. and Jaako J. (1993). Providing plant design knowledge to the operators. In: M. J. Smith and G. Salvendy (Eds.), *Human-Computer Interaction: Applications and Case Studies. Proceedings of HCI International'93, Vol. 1*, pp. 546-551. Amsterdam: Elsevier.

Kellogg W. A. and Breen T. J. (1987). Evaluating user and system models: applying scaling techniques to problems in human-computer interaction. In: J. M. Carroll and P. P. Tanner (Eds.), *Proceedings of ACM CHI+GI'87 Conference on Human Factors in Computing Systems and Graphics Interface*, pp. 303-308. New York: Association for Computing Machinery.

Kieras D. (1992). Diagrammatic displays for engineered systems: effects on human performance in interacting with malfunctioning systems. *International Journal of Man-Machine Studies*. 36, 6, 861-895.

Kirby J. and Heathfield H. A. (1993). An analysis technique for user centred design. In: G. Salvendy and M. J. Smith (Eds.), *Human-Computer Interaction: Software and Hardware Interfaces. Proceedings of HCI International'93, Vol. 2*, pp. 991-996. Amsterdam: Elsevier.

Korf R. E. (1988). Optimal path-finding algorithms. In: L. Kanal and V. Kumar (Eds.), *Search in Artificial Intelligence*, pp. 223-267. New York: Springer-Verlag.

Lawton G. L., Morrison D. L. and Lee P. L. (1993). Managing complexity: display design in process control. In: M. J. Smith and G. Salvendy (Eds.), *Human-Computer Interaction: Applications and Case Studies. Proceedings of HCI International'93, Vol. 1*, pp. 139-144. Amsterdam: Elsevier.

Lim K. Y., Long J. B. and Silcock N. (1992). Integrating HF with the JSD method: an illustrated overview. *Ergonomics*, 35, 10, 1135-1161.

Lipner R. S., Strong G. W. and Strong K. E. O. (1993). The relationship between the task structure and choice of navigational aid in human computer interface design. In: M. J. Smith and G. Salvendy (Eds.), *Human-Computer Interaction: Applications and Case Studies. Proceedings of HCI International'93, Vol. 1*, pp. 522-527. Amsterdam: Elsevier.

Masson M. and De Keyser V. (1993). Preventing human errors in skilled activities through a computerized support system. In: G. Salvendy and M. J. Smith (Eds.), *Human-Computer Interaction: Software and Hardware Interfaces. Proceedings of HCI International'93, Vol. 2*, pp. 802-807. Amsterdam: Elsevier.

McCann C. A., Taylor M. M. and Tuori M. I. (1988). ISIS: the interactive spatial information system. *International Journal of Man-Machine Studies*, 28, 2-3, 101-138.

McCarthy J., Wright P. and Harrison M. (1995). A requirements space for group-work systems. In: K. Nordby, P. Helmersen, D. J. Gilmore and S. A. Arnesen (Eds.), *Human-Computer Interaction: Interact'95*, pp. 282-288. London: Chapman and Hall.

Neugebauer J.-G. (1992). Virtual reality – more than just simulation. *Industrial Robot*, 19, 3, 30-34.

Newell A. and Simon H. A. (1972). *Human Problem Solving*. Englewood Cliffs: Prentice-Hall.

Nielsen J. (1990). The art of navigating through hypertext. *Communications of the ACM*, 33, 297-310.

Norman D. A. (1983). Some observations on mental models. In: D. Gentner and A. Stevens (Eds.), *Mental Models*. Hillsdale: Lawrence Erlbaum Associates.

Patrick J. and Fitzgibbon L. (1988). Structural displays as learning aids. *International Journal of Man-Machine Studies*, 28, 6, 625-635.

Pew R. M., Miller D. C. and Feehrer C. E. (1981). Evaluating nuclear control room improvements through analysis of critical operator decisions. In: R. C. Sugarman (Ed.), *Proceedings of the Human Factors Society 25th Annual Meeting*, pp. 100-104. Santa Monica: Human Factors Society.

Rasmussen J. (1982). Human errors: a taxonomy for describing human malfunction in industrial installations. *Journal of Occupational Accidents*, 4, 311-333.

Rasmussen J. (1986). *Information Processing and Human-Machine Interaction: An Approach to Cognitive Engineering*. Amsterdam: North-Holland.

Reason J. (1990). *Human Error*. New York: Cambridge University Press.

Retz H. and Hunger A. (1993). Development of a processflow manager for an IC-diagnosis systems. In: G. Salvendy and M. J. Smith (Eds.), *Human-Computer Interaction: Software and Hardware Interfaces. Proceedings of HCI International'93, Vol. 2*, pp. 730-735. Amsterdam: Elsevier.

Roberts D. and Warwick K. (1993). An overview of virtual reality. In: *Virtual Reality in Engineering*, pp. 153-67. London: IEE.

Schank R. C. and Abelson R. P. (1977). *Scripts, Plans, Goals and Understanding*. New York: Wiley.

Sharit J. and Chen S. (1993). The use of expert systems for training humans in rule-based reasoning. In: G. Salvendy and M. J. Smith (Eds.), *Human-Computer Interaction: Software and Hardware Interfaces. Proceedings of HCI International'93, Vol. 2*, pp. 819-824. Amsterdam: Elsevier.

Sherrard D. C. and Narayanan M. (1994). The aid of virtual reality in the industry. In: *Wescon/94 Conference Record*, pp. 374-377. Anaheim: IEEE., ERA.

Shneiderman B. (1983). Direct manipulation: a step beyond programming languages. *IEEE Computer*, 16, 57-69.

Stager P. (1991). Error models for operating inequalities: implications for automation. In: J. A. Wise, V. D. Hopkin and M. J. Smith (Eds.), *Automation and Systems Issues in Air Traffic Control*, pp. 321-338. Berlin: Springer-Verlag. (NATO ASI Series, Vol. 73).

Stahre J. (1993). Humanufacturing – operator decision support in a CIM environment. In: M. J. Smith and G. Salvendy (Eds.), *Human-Computer Interaction: Applications and Case Studies. Proceedings of HCI International'93, Vol. 1*, pp. 2-7. Amsterdam: Elsevier.

Sutcliffe A. (1988). Some experiences in integrating specification of human computer interaction within a structured system development method. In: D. M. Jones and R. Winder (Eds.), *People and Computers IV*, pp. 145-160. Cambridge: Cambridge University Press.

Venolia D. (1993). Facile 3D direct manipulation. In: S. Ashlund, K. Mullet, A. Henderson, E. Hollnagel and T. White (Eds.), *Proceedings of INTERCHI'93*, pp. 31-36. New York: ACM.

Ware C., Arthur K. and Booth K. S. (1993). Fish tank virtual reality. In: S. Ashlund, K. Mullet, A. Henderson, E. Hollnagel and T. White (Eds.), *Proceedings of INTERCHI'93*, pp. 37-42. New York: ACM.

Warnecke H.-J., Flaig T. and Walper M. (1993). Virtual reality for improved human-computer interaction in robotics and medicine. In: M. J. Smith and G. Salvendy (Eds.), *Human-Computer Interaction: Applications and Case Studies. Proceedings of HCI International'93, Vol. 1*, pp. 207-212. Amsterdam: Elsevier.

Ye N. and Salvendy G. (1993). Introducing problem solving strategies of users into the interface design. In: G. Salvendy and M. J. Smith (Eds.), *Human-Computer Interaction: Software and Hardware Interfaces. Proceedings of HCI International'93, Vol. 2*, pp. 873-878. Amsterdam: Elsevier.