Script-based spatial user interface: an approach to supporting operators of process control systems

Ivan Burmistrov
Moscow State University

Abstract

In this paper we propose a novel approach to the interface design for process control industries aimed at support of operators' problem-solving activities. The interface is based on a script approach to human problem solving and uses spatial representation of the experts' procedural knowledge. We describe a prototype intelligent user interface for real-time control systems, the Flexible Script Interface (FSI), which is designed to recommend the correct sequence of actions appropriate in relation to the actual task. FSI is implemented with an interactive script graph which represents a task structure and canalizes user's problem-solving behaviour. The prototype system combines 3D computer graphics visualization 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.

Introduction

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).

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 canalizes 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.

(B) Making it possible to interact with the system via such a graph.

(C) Dynamic visualization of script graph 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.

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.

Domain

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 detection and diagnosis of system failures.

Background

Problem solving scripts

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 conceptualized as a mental model or schema, organizes and directs a person's selection and usage of information in generating a solution.

One of the concepts of mental model is script, or event schema. Abelson (1981) defines a script as a hypothesized cognitive structure that when activated organizes comprehension of event-based situations. A script represents stereotyped knowledge structure that describes appropriate sequences of goal-directed actions in a particular context. Scripts consist from a number of scenes, which in turn are constituted by sequences of definite atomic operations. 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 organizes the set of operations leading to the solution of a problem. They hypothesize 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.

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. Computer graphics displays make it possible to display both the topological structure of a system 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.

Virtual reality

Recently, virtual reality (VR) 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: visualization of complex data, controlling industrial robots, remote operation of equipment, enhancing communications, operations training, and virtual prototyping and design. These applications are but an initial step in identifying opportunities for using VR in manufacturing organizations. The three-dimensional nature of VR, and the mechanisms for interacting with objects in VR environments, makes the enormous amount of data available much more accessible to decision makers; traveling through, and manipulating objects within, the virtual facility offers a much more causal, natural and direct interaction than working indirectly through programs providing only two-dimensional representations of the problem. We believe that one of the most significant aspects of the technology is its ability to improve decision-making processes from both qualitative and quantitative perspectives.

Overview of the Flexible Script Interface

Script graph

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.

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 links to the 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;

(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.

Arcs, or links, which connect nodes and allow to move from one node to another may be: (1) permitted, (2) prohibited (temporarily blocked), or (3) recommended. 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.

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 agent 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.

Current state of the work

Flexible Script Interface has been implemented for the PersoPlan (Personal Planning), a decision support system based on psychological analysis of individual's motivation in decision making. The last version of the PersoPlan's user interface was restricted to a 2D interface with pop-up script windows. This implementation of FSI has been described in detail in our previous paper (Burmistrov, 1992).

Our current work focuses on the development of the 3D version of 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 VR environments.

Figures 1–5 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.

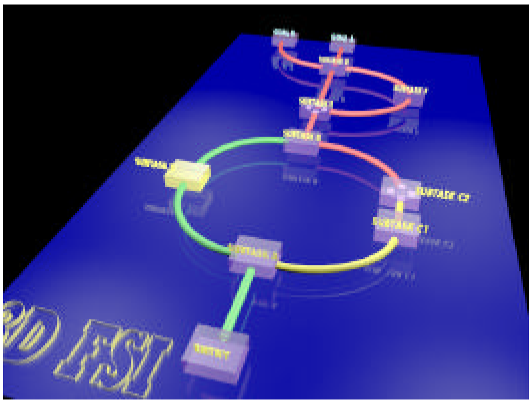

Figure 1

Figure 1 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) to end nodes (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. FSI is based on the principles of task context and cognitive economy in its representation of a task structure. 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.

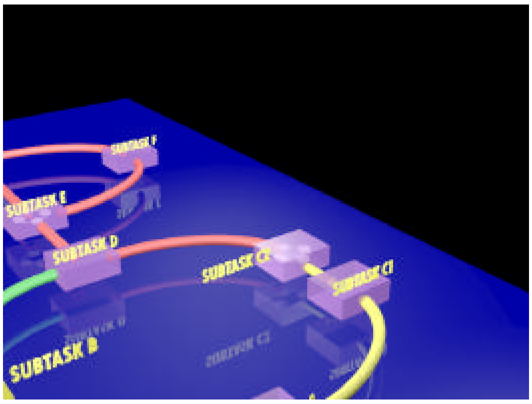

Figure 2

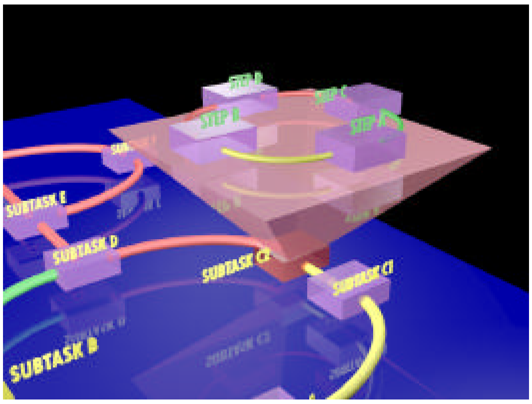

Figure 3

Figure 4

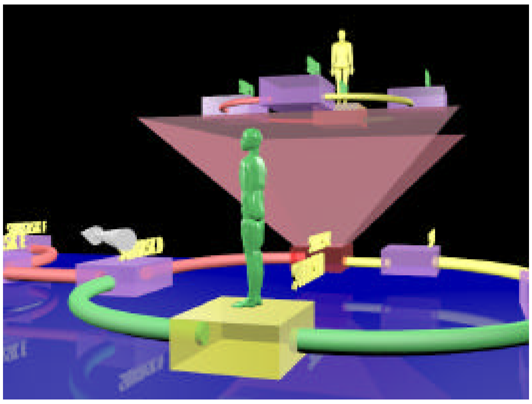

Figure 5

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. Figure 2 and Figure 3 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.

Figure 4 presents the concept of the 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 Figure 5).

Practical implementations of FSI 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.

References

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

Burmistrov, I. (1992), 'Flexible Script Interface: an intelligent spatial interaction style', in G. C. van der Veer et al. (eds.), *Human-Computer Interaction: Tasks and Organization. Proceedings of ECCE6*. CUD: Rome.

Gritzman, M., Kluge, A. & Lovett, H. (1995), 'Task orientation in user interface design', in K. Nordby et al. (eds.), *Human-Computer Interaction: Interact'95*. Chapman and Hall: London.

Hershey, D. A., Walsh, D. A., Read, S. J. & 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*, Vol. 46, pp. 77-101.

Stahre, J. (1993), 'Humanufacturing – operator decision support in a CIM environment', in M. J. Smith & G. Salvendy (eds.), *Human-Computer Interaction: Applications and Case Studies*. Elsevier: Amsterdam.

Ye, N. & Salvendy, G. (1993), 'Introducing problem solving strategies of users into the interface design', in G. Salvendy & M. J. Smith (eds.), *Human-Computer Interaction: Software and Hardware Interfaces*. Elsevier: Amsterdam.