A Visual Language for the Acquisition and Display of Plans

Dirk E. Mahling and W. Bruce Croft
Department of Computer and Information Science
University of Massachusetts

Abstract
One of the major obstacles to the use of planning systems has been the difficulty of acquiring descriptions of how tasks are carried out in particular environments. In order to address this problem, we propose a visual language for the acquisition, display and debugging of plan knowledge. The basis of this language is a model of the user's view of tasks, supported by a series of psychological experiments. DACRON supports the acquisition of plan knowledge by providing graphical representations of domain entities from the users' point of view. It will also allow users to replay the specification of plans and interactively debug this animated display.

1 Introduction
Knowledge based systems, such as rule based expert systems and planners, can potentially be used in many environments to support people in their daily work. The main obstacle to the use of these systems is the acquisition of knowledge about the users' environments and the intelligibility of the systems advice.

The slow and costly process of employing knowledge engineers to supply the system with this knowledge has led to more direct means of knowledge acquisition: end user programming [11,7,22] and machine learning [15]. While a considerable amount of work has been done in knowledge acquisition and display of advice in the rule based expert system area [3,4], user models and visualization have not played a major role. Planners have received even less attention. The acquisition of plan knowledge is particularly important given the increasing interest in the use of planners to support cooperative work [2]. We are proposing a visual plan acquisition and display language for naive users based on a cognitive model.

Plans differ in structure and use from the if-then rules used in expert systems. This difference poses special demands on the acquisition of plans from domain experts who are usually non-programmers, and it also presents special problems in the display of the stages and results of the planning process, which must be presented to the user in an immediately understandable form.

Figure 1 shows a plan schema from the POLYMER planning system developed at the University of Massachusetts [10]. This plan schema contains information about the goal, preconditions, subgoals, and effects of the activity it represents. It describes how to purchase an item and specifies causal dependencies, responsible agents, constraints and affected objects as well as the goal and precondition information. It is obvious that such a representation is not immediately understandable to a naive user, neither for the purpose of acquisition nor for the display of advice.

To build an immediate-use language for a planner we have to have a model that relates the elicitable knowledge of the end-user, who is the domain expert, to the requirements of the planner. We present a model for the recall and understanding of plans in section 2. Building on this cognitive model we present a visual language for plan acquisition and display in section 3. The visual language is part of the DACRON plan interface system. DACRON is designed to be used in conjunction with the POLYMER planning system, and issues such as the construction of plans, execution of plans, and handling of exceptions are POLYMER's responsibility. Section 4 reports user reactions to the visual language from a study conducted as the last stage of the first design cycle in the iterative design process [17,26]. The final section discusses our future research plans.

2 A Framework for the Users' View of Tasks
Designers who intend to build systems that acquire and display task knowledge from end users have to carefully identify those aspects of the users' knowledge they want the system to acquire. In building a model, these aspects must be distinguished from the non-conscious cognitive processes used in explaining human behavior.

We need a model that makes predictions about what humans know about tasks. A task is an activity which is viewed by the human as a unit at its standard level of abstraction, e.g. purchase an item for a secretary or receive reimbursement for travel for a manager. This focus eliminates many existing models of human cognition as possible candidates because they either make no predictions on the task level [6,5,20], evaluate work done with computer systems themselves [9,18] or address interaction and discourse issues only [23,24,21]. None of these theories makes statements about structures and processes involved in the recall of complex tasks from long term memory. Therefore, we started a series of interviews and experiments to build a framework for the users' view of complex tasks.
Interviewing domain experts about how they conduct certain complex tasks usually leads to a description of an example. In these examples, we observed the grouping of operations, which are the lowest level of description, into units. These units fit what is called *Handlung* by German psychologists and philosophers [1,13,8]. A *Handlung* is a conscious, goal-directed act of a human being, controlled by will, directed towards shaping reality. It contains three aspects: an intended goal, an analysis of means for its achievement and the decision to do so. A *Handlung* contains operations, conducted by a human, which transform states of reality into other states, serving a certain purpose [12]. (Clearly a *Handlung* is more complex than the English equivalent act, however, in the remainder of this paper we will refer to it this way).

The act is the smallest coherent unit in the description of a task that appears to be at the appropriate level of abstraction. This individually perceived appropriateness as smallest unit, varying from person to person and from task to task, makes the concept very suitable for our purpose. It also distinguishes the *act* from GPS-operators [16]. While operators are established by task analysis, acts are the representation of the human's perception of these tasks.

As the above definition of acts does not lend itself to immediate operationalization, we try to formalize its properties in information processing terms [19]. The representation of an act consists of four properties which are discussed below.

The first property of an act is the conscious goal; in our case it is the intention to complete a certain task. During work on one task, this goal remains the same, only situations (states) change, not intention. People consciously know about this goal and should be ready to report it without difficulty.

The current state of reality, as mentioned in the above definition [12], constitutes the second property. We call this property the pre-situation. During execution of the same task (same goal) the situation determines which operations are to be applied. Only when the task is completed or interrupted does the goal change. When people are actually performing a task, they directly perceive the pre-situation. During recall, the content of the working memory mirrors this state. People who are imagining working on a certain task should be readily able to report the current state of affairs.

The third property of the definition is the decision to generate behavior, which in turn is observable. We represent this as a list of names of the operations to be generated. As we are not concerned with actual execution of tasks, but only the part of them that can be reported by recall, we must be satisfied with the name the person ascribes to a certain operation. The mapping of these names to primitives of the system is a question of coding and will be discussed in the next section.

The result is also available in the recalled act. We call this fourth property, describing the situation after the application of operations, the post-situation. The post-situation is described in the same terms as the pre-situation but it includes the changes caused by the operations. The justification for this property comes from the explanation that acts transform states of reality into other states of reality.

The complete representation of a formal act is given by the structure in figure 2. We are aware that the operationalization of such a complex concept as an act cannot capture every nuance in meaning and must fall short in certain aspects of description. The structure we present here will certainly have to be amended. Acts are the product of recall of deeper cognitive structures. We assume that recall can be directed to decompose and sequentialize acts. Sequentialization is the process of finding a sequence of acts that leads from one state of the world to another one. Decomposition is the process of breaking an act into lower level acts, making the higher level act the goal and generating all its constituting operations as lower level acts. As our framework is concerned with the existence of recall processes and structures and not with low level, underlying cognitive processes, we make no assertions as to how this is accomplished. It is sufficient for our purposes to know that these processes and structures exist at all, not how and why.

In addition to the recall of activities, we assume the recallability of objects, relations and states of the world. As these entities comply to a certain degree with those items usually encountered in recall experiments, we need make no additional assumptions.

With the operationalization of acts and the proposed subprocesses, we were able to propose and test the following hypotheses about the recall of subject performed activities from long term

### Figure 1: Purchase task as POLYMER plan.

<table>
<thead>
<tr>
<th>Goal:</th>
<th>Intention of the Whole Act</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pre-Situation:</td>
<td>List of Properties</td>
</tr>
<tr>
<td>Operations:</td>
<td>List of Operations</td>
</tr>
<tr>
<td>Post-Situation:</td>
<td>List of Properties</td>
</tr>
</tbody>
</table>

Goal: Purchase

Pre-Situation: List of Properties

Operations: List of Operations

Post-Situation: List of Properties
memory. It is important to realize that these are only one set of many possible descriptions for the phenomena occurring but that they seem to be a reasonable one for our purpose.

1. A general task (goal) and a specific situation (pre-situation) will result in the recall of a specific set of operations.
2. Changing the pre-situation or the general task (goal) will result in the recall of different operations.
3. A goal, a start situation and an end situation will result in the recall of a sequence of acts. This sequence leads from the start situation over intermediate situations to the end situation.
4. An operation may itself be composed of acts and those acts are recallable (decomposition).
5. If the pre-situation of an act is not achievable, other acts with the same goal but different pre-situation can be recalled instead (alternative acts).
6. Operations can be distinguished from the post-situation created by these operations.
7. Causal dependencies between operations can be indicated.

To test these hypotheses, we first conducted a pilot study with four departmental secretaries and then a larger series of experiments. Our subjects in those experiments were undergraduate and graduate students at the University of Massachusetts and Smith College. We could verify all hypotheses. Special care had to be taken with hypothesis number six (distinguishing post-situation from operations). Subjects were able to report the post-situation if asked directly about it but they could not report the post-situation if queried in general terms. For a detailed discussion of the experiments and their results see [14].

3 The DACRON Visual Plan Language

DACRON is a visual plan acquisition and display system with the goal of immediate usability for the domain expert. We developed the DACRON interface and the visual language based on the model described in the previous section. Our goal was a system that would help domain experts to specify their knowledge in a language that is close to their individual way of operational knowledge (acts) in the domain and to give advice that is understandable to people unfamiliar with both, the task and the planning system.

Every item in the knowledge base of the planner is represented by an icon. A permanent window, the archive, presents all or part of the knowledge base (figure 3). If there are too many icons to fit the current window for the archive, a zooming, scrolling and reshape facility exists that allows users to bring other parts of the knowledge base into view.

To reorganize the clustering of icons in the knowledge base, users may employ a facility from the archive. This facility allows users to order the icons in the knowledge base by certain keys like name, date of creation, etc. or to search for icons by name or components such as goal, pre-situation or any other property. The commands can be invoked over a menu on the archive.

The icons in the archive fall into two classes:

- icons for acts;
- icons for relations and objects.

Every icon can be opened with the mouse and presents the user with a graphic form editor for the particular entity that is represented by that icon (figure 4). In the case of acts, the editor shows compartments that refer to the pre-situation, the operations and the post-situation of the particular act, which correspond directly to the same entities in the cognitive framework. In the case of knowledge acquisition, these compartments are ready to accept copied icons from the archive as input. In the display case, icons representing values appear in these compartments.

Upon opening an icon for an object or a relation, the editor shows rows that hold names for arguments and icons representing values (figure 4). Users can scroll up and down in these editors.

To specify a new task, users can either copy and modify an existing act icon or begin with a blank one. Users start by clicking on the icon and choosing the create new class option from the pop-up menu. Upon this, the editor opens and the users type in the name for the new act or objects, e.g. purchase-item-1 (figure 3). Then they can specify the goal by copying one or more object or relation icons to the goal compartment and setting the arguments of these types to the desired goal-values. To specify, for example, that the goal of the act purchase-item-1 is that the users themselves will possess the item they would have to do the following:

1. copy the icon for self from the archive to the goal compartment of purchase-task-1;
2. open that icon, to see the editor for it;
3. scroll to the argument row possess;
4. copy the item icon from the archive into that row.

Figure 5 illustrates this process. The other parts of the plan (before-situation, operations, after-situation) are specified in the same way.

Causal dependencies between acts in the operation compartment are indicated by arrows. Users click in the actions compartment and select the set dependencies option. Then they click on the icon for the plan step that enables some other plan step. A rubber line appears and can be dropped on the icon for the plan step that is enabled.

52
Constraints are placed in the compartment that contains the object to which a certain constraint applies. In the case of inter-object constraints which might apply to objects in different compartments the users might choose the more convenient one.

Another implication of the framework was the decomposition of the goal of a general task into a sequence of acts. As the granularity of these acts is dependent on the view of the domain expert, DACRON does not enforce a decomposition hierarchy but lets the users follow their own decomposition path. The users give the sequence of acts at a level they think is appropriate. The users do this by specifying an act type. DACRON guarantees completeness of this specification by prompting the user for unspecified compartments.

DACRON has two other facilities which are intended to aid users at the display stage and the debugging stage of the knowledge acquisition process. A debugger can be invoked to give an animated demonstration of the specification process of a type. The animation shows how the type was created, in which order arguments were given, what icons were placed into the type, etc. As this debugger is interactive, it is possible to stop it at any given moment and change the specification of the type shown. The animation can also be used for training purposes, teaching new users how DACRON is used.

The second facility is the reviewer. It is used to present sequences of plans and alternatives at crucial points in the planning process to the users. In this situation, we use the visual language for the animated visualization of the planning process in POLYMER.

4 User Feedback

In this section we report user reactions to the visual language. We tried to involve users as early as possible in the actual interface and language design to guarantee usability and appropriateness [25]. Our subjects were two secretaries and four graduate students from the computer science department.

Every subject was given a ten minute introduction to the system, with the experimenter working with the system and explaining while letting the subject gradually take over control, until the subject was working with the system completely on its own. The whole session lasted about fifty minutes. After the ten minute introduction the experimenter only observed the subject's actions and descriptions of the DACRON displays.

In our analysis we divided the subjects into two groups. The first group was defined by the fact that they were shown activities and objects in the knowledge base that had been obtained from them in interviews fifteen months earlier. The second group was shown the same activities and objects. This second group did not know how to perform these activities beforehand.

The first group had to recognize activities, their goals and the way they were decomposed into subgoals. Some of the activities that were shown to these subjects had semantic errors in them, e.g. a wrong decomposition or a meaningless goal. The subjects were expected to fix these bugs.

The subjects we tested were able to recognize the activities they had reported to us fifteen month ago. They could even recognize subtle preconditions and decomposition details. As some of the office activities in the department had changed they could identify these changes and specify the correct way using DACRON. They could also recognize and correct the wrong and meaningless goals and decompositions that had been added to some of the activities. Subjects were able to understand the meaning of the different icons, editors and compartments. The semantics of specification by moving icons and setting graphic values was also immediately understood and used with very few errors. Minor problems were discovered in the sequencing of menu items and navigation in the archive.

The second group was asked to describe how they carry out a certain task, e.g. purchase an item for the computer science department. These subjects did not know about certain agents and objects that were involved, like accounting department, purchase order, accounts, etc. and they did not know what activities to perform. Their only source of information was the DACRON display of the activities and objects. In this setting we did not add wrong or meaningless information. Using the DACRON interface subjects were able to give a correct verbal description of the activity. Goals, preconditions and decompositions were correctly identified. Subjects were also able to specify preconditions they felt necessary for certain activities.

5 Future Work

The results we obtained in testing the cognitive framework and the user feedback are very encouraging. The major design decisions appear to be correct. To support the validity of our design, we are planning to test and evaluate a complete version of DACRON with users in a local office environment. These user studies would involve the specification of office tasks by clerks and secretaries using DACRON and at a later point the display of whole office activities to office workers in the process of accomplishing complex tasks.

In the future we plan to add the component for the display of effects and a major support system for the specification of effects. We also plan to add a component for the specification of more sophisticated constraints. In order to give users better advice for their work and making the results of the planning process of POLYMER more transparent, we will devote a large part of our work on display to the animated presentation of the planning process. This work will also be useful in the interactive debugging of activities.

Figure 4: Icon and Editor displaying the purchase-task.
Figure 5: Specifying the Goal.

References


