Pictures in Advice-Giving Dialog Systems: From Knowledge Representation to the User Interface

Thomas Strothotte
Institut für Informatik, Universität Stuttgart
Azenbergstr. 12, D-7000 Stuttgart 1, F. R. Germany
E-mail: thomass%ifistg.uucp@unido.bitnet

Abstract

In knowledge-based advice-giving dialog systems, problem solving is typically carried out using symbolic processing and the output is usually produced in a textual form. This kind of processing, based on symbols and language, makes it very difficult to design systems which give explanations in a pictorial form. To tackle this problem, we present techniques which allow a smooth transition between the symbolic processing of problem solving and the pictorial presentation of a solution. The techniques are based on new data structures, called *picture-frames*, for pictorially representing knowledge. Pictures are manipulated and built up incrementally and upon completion of the problem-solving process, represent the solution to the problem. We discuss our experience with picture-frames on the basis of a prototypical tutoring system implemented in Prolog.

Keywords: advice-giving systems, expert systems, user interfaces, diagrams, pictures, graphics, frames.

1 Introduction

Today's use of pictures and diagrams depends critically on whether their presentation is made with an interactive or a non-interactive medium. Pictures and diagrams are used extensively in materials which can be prepared in a fixed format in advance of being presented, such as in books and films; the communication is one-way, with the reader (or viewer) having no input capabilities. Numeric data are conveniently represented in graphs and charts. More complex non-numeric information, dealing for example with details of processes involving real objects, are often explained with the help of pictures. Examples of such pictorial explanations of non-numeric data can be found in news and weather reports.

In interactive advice-giving systems, on the other hand, the dialog between the user and the system is presently based more on language rather than on pictures. Indeed, in their recent review article on user interfaces to expert systems, Carroll and McKendree [2] never once mention the use of picture-like explanations. While icons are routinely used in today's user interfaces, they are not applied in a knowledge-based manner: They typically represent a specific command, state or object, while the system has no concept of the semantics which a user would associate with particular arrangements of such icons.

A major source of difficulty in graphically presenting non-numeric information in interactive advice-giving systems is that that the problem-solving modules of knowledge-based systems rely almost exclusively on symbolic processing. Problems to be solved, domain knowledge, and the final solutions to the problems are typically represented in a symbolic form - for example with Minsky's frames [6]. The transformation of a piece of non-numeric information from a symbolic to a pictorial form is usually rather difficult to carry out since symbolic representations do not, in practice, contain such details as are necessary for the system to perform an appropriate transformation. The information which is stored in a knowledge base typically reflects the logical structure of the domain being described and is captured in a linguistic form, rather than reflecting the physical structure of the situation.

Our goal in the present study is to develop techniques which endow advice-giving dialog systems with the capability of using pictures to explain operations to a user in situations in which such pictures are superior to written texts. For reasons of compatibility with existing knowledge-based systems, the usual symbolic problem-solving methods of advice-giving dialog systems must remain in place. In addition, we require that system responses be tailored to the individual user's particular needs. This means that at least part of the graphics must be designed by the system at runtime, since pre-drawn pictures cannot take into account information specified by the user in the previous parts of the dialog.

In this paper we develop a method for systematically building up pictorial representations of non-numeric information to be presented to the user of a knowledge-based advice-giving system. In Section 2 we report briefly on previous work toward solving this problem. The experience we gathered led to the development of new data structures which draw graphics into the problem-solving process; these are discussed in Section 3. Section 4 describes an application of our techniques to a prototypical tutoring system for high-school chemistry. We discuss our techniques in Section 5.

2 A First Approach

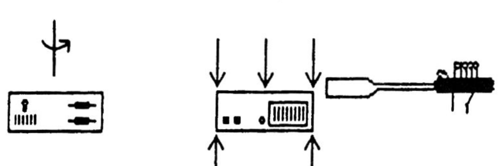
One method of producing pictorial presentations at runtime makes use of techniques drawn from computational linguistics. In this area, considerable work has been carried out on natural language generation from semantic representations (see for example McDonald [4] and McKeown [5]). Input to such systems is typically a frame-based description of an event, output is its natural language description. Building on this work, we designed and implemented a picture generator which accepts as input a frame-like semantic representation and produces a sequence of pictures which express the semantics (see [3]). The system has a lexicon with graphic symbols and descriptions of their movements to express actions (i.e., verbs).

We designed and implemented a prototypical dialog system for providing users with information concerning the installation and maintenance of microcomputers. A user can ask questions like "How do I install a memory expansion board?" Two sample frames of the output of our system is shown in Figure 1: In (a) the user is instructed to turn around his machine, while in (b) he is instructed to undo screws in the 5 positions pointed to by arrows.

While this method is very general, it proved to be too powerful for limited domains. A difficult problem turned out to be the design of the pictorial presentations which appear "aesthetically pleasing." For pre-drawn pictures, this design is carried out by a graphic artist, but we had considerable difficulty in writing algorithms to carry out this task. Further, since the graphics were added after the problem-solving process was completed, it was difficult to ensure that the semantics of the final presentation actually did correspond to the intended meaning. This latter problem is one which also plagues natural language generation systems.

3 Graphics in Knowledge Representation

Our experience with the picture generator led us to conclude that graphics must also be part of the problem-solving process. The challenge is to enable a smooth transition between the usual symbolic process of problem solving to a graphic representation of a solution. The techniques which we describe in this section take steps in this direction.



(a) (b)

Figure 1: Two snapshots of a generated graphic presentation

3.1 Picture-Frames

In order to bring graphics into the representation of knowledge, we designed data structures called *picture-frames*, which are extensions of Minsky's frames [6]. The extensions deal with pictures and techniques to modify them in order to flexibly represent solutions to problems.

The central part of a picture-frame is a picture (or graphic) of a typical scene in the domain of the application. The picture contains the objects in the configuration normally found when working in the domain. Besides the usual slots and methods of Minsky's frames, a picture-frame has *graphic slots* for making additions or other changes in the picture. The parts of a picture-frame can be summarized as follows:

A picture and its name. A (black and white) picture is represented by a pixel-matrix, with each bit representing a pixel. However, any other representation of a picture, from which such a bit-representation can easily be generated, is sufficient.

Slots for manipulations of the picture. In these *picture-manipulation slots*, minor modifications of the basic picture are recorded. A slot is associated with each part of the picture which could conceivably be changed (rotated, translated, etc.). Placing a value into such a slot results in an "if-added" procedure being called and making the corresponding change in the picture.

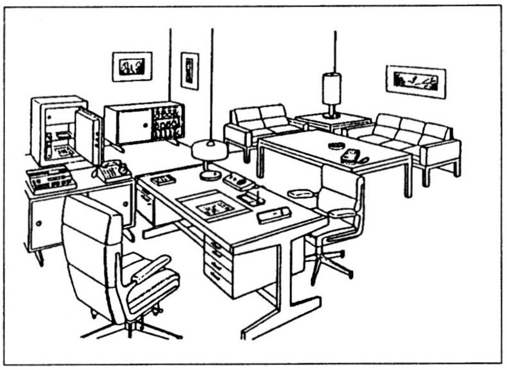
Slots for graphic symbols. These *graphic-symbol slots* serve to add symbols to the basic picture. For example, an object in the picture may have associated with it a graphic-symbol slot called *label*. At the time this slot is filled by a value (for example a character string), a line pointing at the object is automatically drawn into the picture from the object in question to some free space at the side of the picture. The character string is then written into the picture in the free space at the end of the line. To a human viewer of the picture, this has the effect of naming (i.e., labeling) the object.

Slots to define hierarchies. These *hierarchy slots* are used to record names of other picture-frames which represent more detailed views of parts of the current picture.

Further slots and methods. Further knowledge pertaining to the scene of the picture-frame is stored in the usual frame-based manner.

The parts of a picture-frame are summarized in Figure 2. The picture, adapted from one in the German "Duden Bildwörterbuch" shows a typical office scene. Modifications and additions to such a basic picture can be used to produce explanations for users.

Name: *Executive office*



Slots for manipulations of the picture
Slots for graphic symbols
Slots for hierarchially organizing picture-frames
Further slots and methods

Figure 2: Composition of a picture-frame

3.2 Problem Solving with Picture-Frames

The picture-frames are used for problem-solving in a manner which extends that of normal frames. Given a problem, the system searches in its knowledge-base of picture-frames for the one which corresponds most closely to the problem. The facts which are known from the input are recorded directly in appropriate slots; others are computed with the help of methods and then recorded in slots, thereby "solving" the problem. If this process can successfully be carried out, the picture of the picture-frame is finally produced as the solution to the problem. Otherwise, the search for a more appropriate picture-frame continues and the procedure is repeated.

The important aspect of the use of picture-frames is the mechanism to make changes to the basic picture. In normal frames, information is simply recorded in slots; in picture-frames, a graphic representation of this information is recorded in the picture in a manner which can be understood by an external (i.e., human) observer. Thus the solution is obtained incrementally through a series of manipulations of the basic picture and is available as soon as the problem-solving process is complete. This procedure circumvents the process of generation (of natural language or pictures) from an internal form of the solution into an external form which can be understood by humans.

The graphic symbols added to the basic pictures have the effect of leading the user's attention through the picture. They highlight features which are considered important by the system for the understanding of the solution to the problem. Furthermore, these symbols add *language* to the pictorial presentation, since they can include text. This aids the user in verbalizing for himself the solution.

4 A Prototypical Application

To illustrate the concepts introduced in the last section, we designed and implemented a prototypical tutoring system for chemistry problems at the high-school level. The system is written in Prolog.

Chemistry is a subject in which diagrams play an important role in conveying information. Textbooks typically explain experiments with the help of diagrams showing flasks, test tubes, Bunsen burners etc. To a student, such diagrams of the experimental set-up are essential because chemical formulae only tell what happens in a reaction on the molecular level, not how to produce it. For example, information such as the physical state of the chemicals, whether heat must be added, and in what order chemicals must be added to one another are not contained in the equation for a reaction.

In examining high-school chemistry textbooks, we noticed that there are a small number of basic experimental techniques which can be combined to carry out arbitrarily complex experiments. In each of these techniques, the apparatus changes only slightly from one experiment to the next. The fundamental aspect which changes is which chemicals are used and in what order the basic experimental techniques are applied. Thus we designed picture-frames for these basic operations (see Appendix A for examples). Graphic-symbol slots were attached to each picture-frame to represent the chemicals placed into the various containers. The picture-frames can be combined to portray larger experiments.

The system has a set of rules representing chemical reactions and the experimental techniques necessary to carry out the reaction. A user can ask the system questions by calling up an appropriate Prolog predicate.[1] The system determines which reactions are necessary and produces a picture by augmenting the appropriate picture-frames. For example, with the following command, the user asks "How is H_2 produced?" The variable B is unified with the augmented picture:

```
?- produzieren('H2', B).
```

Figure 3 shows B which represents the solution to the problem the user posed. The system has determined that H_2 can be produced by dissolving Zn in HCl (a liquid). Thus the picture-frame "substanz_loesen" ("dissolve a substance") (see again Appendix A) is selected and appropriately augmented with labels for the chemicals.

A more complex example is the question "How is N_2 produced?" Here a sequence of reactions must be carried out. The tutoring system builds up the experiment one step at a time, showing what chemicals are involved. In these examples, a simple, short text appears in the picture, giving a verbal explanation of the key point, while the details

[1] In this prototypical system, we did not write a fancy input system for the pupil, since it would have distracted from the point we are making.

remain encoded in the picture. The system's response is illustrated in Figures 4 to 7. In the first step (Figure 4), NH_3 in fluid form is heated up, producing this same chemical as a gas. Next, the NH_3 is dried with the help of calcium chloride, $CaCl_2$ (Figure 5). In the third step (Figure 6), CuO is heated and the NH_3 passed over it, yielding N_2. Finally, the N_2 is trapped in an upside-down beaker (Figure 7).

As a final example, we show how one picture can be augmented by different graphic-symbol slots in order to answer two very different questions. Figure 8 (a) shows a reaction being described, whereas (b) explains the apparatus required to produce that reaction. The basic picture is the same, but it is augmented in different ways. Both can be produced if the picture-frames have different sets of slots for graphic symbols.

Further details concerning these examples may be found in [1].

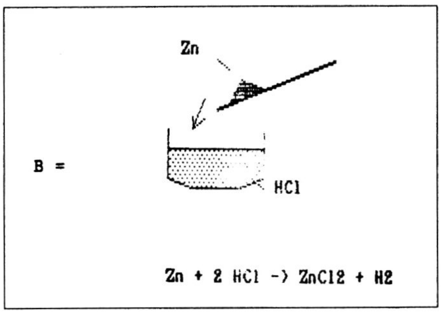


Figure 3: Answer to the question: "How is H_2 produced?"

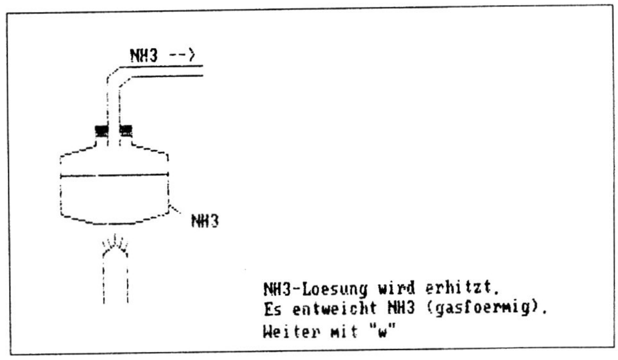


Figure 4: Producing N_2: Step 1.

Translation: NH_3 *solution is heated.* NH_3 *escapes (as a gas). Continue with "w".*

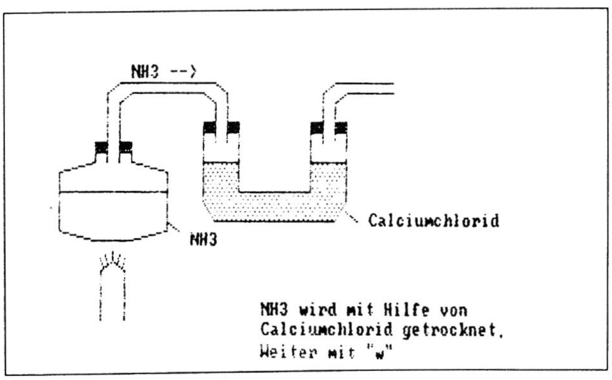


Figure 5: Producing N_2: Step 2.

Translation: NH_3 *is dried with the help of calcium chloride. Continue with "w".*

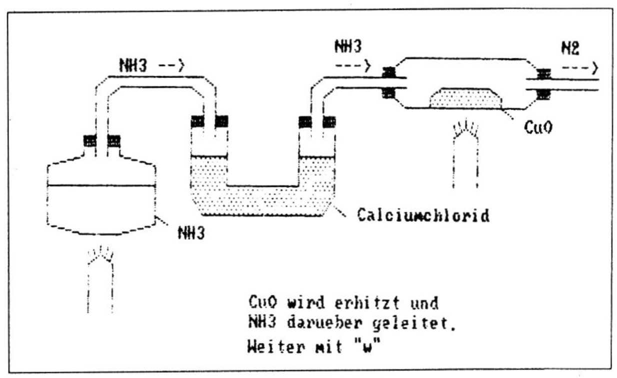


Figure 6: Producing N_2: Step 3.

Translation: CuO *is heated and* NH_3 *passed over it. Continue with "w".*

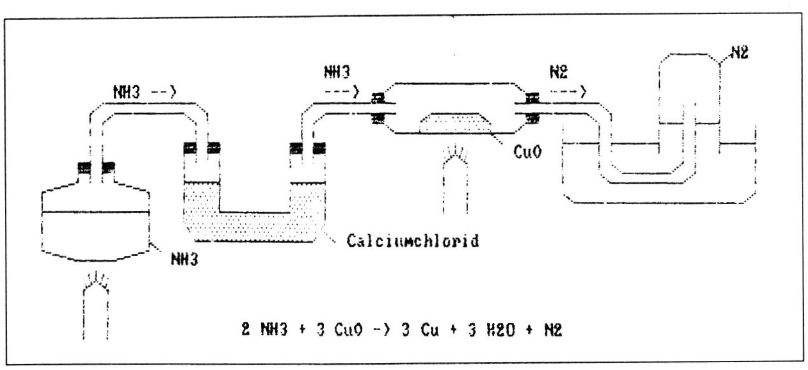
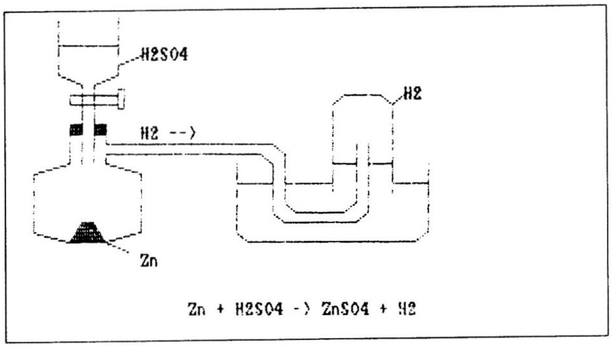
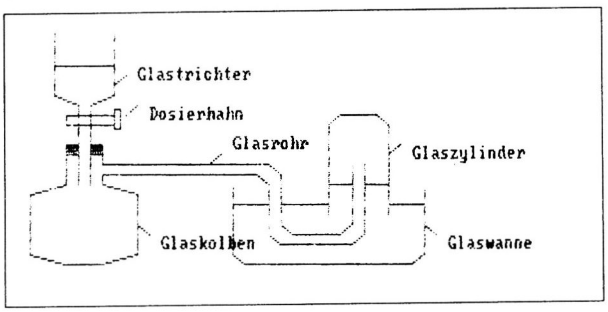


Figure 7: Producing N_2: Step 4.



(a)



(b)

Figure 8: Example of a picture which is augmented in different ways.

5 Discussion

Using graphic techniques for the presentation of information encourages a stronger cooperation between the user and the system. The system aims at *making available* the solution of the problem posed by the user, rather than literally "spelling it out". Instead, the solution is encoded in the pictorial answer and the user is left to ascertain it himself. The user is more involved in the problem solving process because he must think to understand the solution.

A stronger involvement of the user in the problem-solving process has the decided advantage that the system need not consider the level of competence of the individual user to the extent necessary with textual output. Adjustment of output to experts or novices has proved to be a significant problem for designers of advice-giving dialog systems [2]. In the past, user models [7] have been used to try to model the perceived level of competence of the user. By contrast, good graphics tend to remove the problem: An expert can survey the picture to obtain quickly the information which he needs without being bothered by the additional information which a picture necessarily contains. A novice can study the picture's details and discover all it expresses. Such a wide range of users for which a single picture is appropriate can hardly be served by a single natural-language text.

Our work opens up various problems for further study. First, a methodology for the acquisition of pictorial knowledge represented in picture-frames must be developed. At present, all knowledge in our prototypical system is hand-coded into the knowledge base. Second, hierarchies of picture-frames and the inheritance between them must be investigated. In the current system, hierarchies are used to organize discrete enlargements of pictures. Techniques of smoothly zooming into higher levels of detail ought to be realizable. It is unclear how this can be accomplished within the discrete hierarchies of picture-frames. Finally, new techniques for user input, based on the pictorial output of the knowledge-based system, should be developed. An implementation in an object-oriented language would be appropriate.

Acknowledgements

I would like to thank D.Böcke for implementing the system and for contributing ideas to its design and Charles Brown for his comments on an earlier draft of the paper. I would also like to thank Prof.Dr.R.Gunzenhäuser for his support of this work.

References

[1] D. Böcke. *Graphische Wissensrepräsentation zum interaktiven Problemlösen.* Diplomarbeit, Institut für Informatik, Universität Stuttgart, September 1988.

[2] J. M. Carroll and J. McKendree. Interface design issues for advice-giving expert systems. *Communications of the ACM*, 30(1):14–31, January 1987.

[3] Th. Fehrle, Th. Strothotte, and M. Szardenings. Generating pictorial presentations for advice-giving dialog systems. In M. J. Tauber and P. Gorny, editors, *Visualization in Human-Computer Interaction*, Springer-Verlag, Lecture Notes in Computer Science, Berlin-Heidelberg-New York, 1988. (in print).

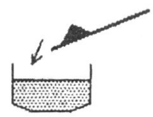
[4] D. D. McDonald, J. D. Pustejovsky, and M. M. Vaughan. Factors contributing to efficiency in natural language generation. In G. Kempen, editor, *Natural Language Generation: New Results in Artificial Intelligence, Psychology and Lingustics*, Kluwer Academic Publishers, Dordrecht/Boston, 1987.

[5] K. R. McKeown. Discourse strategies for generating natural language text. *Artificial Intelligence*, 27:1–41, 1985.

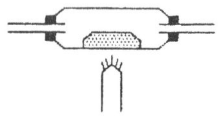
[6] M. Minsky. A framework for representing knowledge. In P. H. Winston, editor, *The Psychology of Computer Vision*, pages 421–452, McGraw-Hill, New York, 1975.

[7] E. Rich. *Building and Exploiting User Models.* Ph.D. Thesis, Computer Science Department, Carnegie-Mellon University, April 1979.

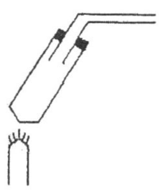
Appendix A: Picture-Frames for a Prototypical Tutoring System



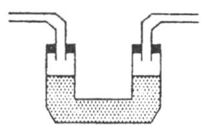
Name: "substanz_loesen"
Dissolve a solid in a fluid



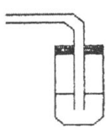
Name: "gas_strom"
Pass a gas over a solid



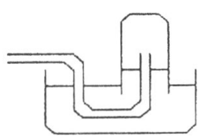
Name: "Substanz_erhitzen"
Heat up a substance



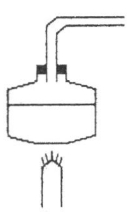
Name: "gas_trocknen"
Dry a gas



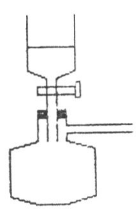
Name: "in_fl_leiten"
Dissolve a gas in a liquid



Name: "pneu_wanne"
Trap gas in a beaker



Name: "fl_verdampfen"
Evaporate a liquid



Name: "fl_tropfen"
Let a fluid drip into a beaker