An Improved Interface for Tutorial Dialogues: Browsing a Visual Dialogue History

*Benoît Lemaire** *Johanna Moore†*

University of Pittsburgh
Learning Research and Development Center
3939 O'Hara Street
Pittsburgh, PA 15260, U.S.A.
Tel: (412) 624-7050
{lemaire, jmoore}@cs.pitt.edu

ABSTRACT

When participating in tutorial dialogues, human tutors freely refer to their own previous explanations. Explanation is an inherently incremental and interactive process. New information must be highlighted and related to what has already been presented. If user interfaces are to reap the benefits of natural language interaction, they must be endowed with the properties that make human natural language interaction so effective. This paper describes the design of a user interface that enables both the system and the user to refer to the past dialogue. The work is based on the notion that the dialogue history is a source of knowledge that can be manipulated like any other. In particular, we describe an interface that allows students to visualize the dialogue history on the screen, highlight its relevant parts and query and manipulate the dialogue history. We expect that these facilities will increase the effectiveness of the student learning of the task.

KEYWORDS: Tutorial interactions, dialogue history, information visualization.

INTRODUCTION

Our research is concerned with improving the effectiveness of tutorial interactions between a computer and a student. It is based on the following claim:

> *Tutorial explanations are more effective if they are linked to what has been previously said.*

This claim is supported by evidence from several psychological studies, which have shown the importance of relating material being taught to what the student has in his mind. In particular, reminding the student of relevant prior knowledge helps the student acquire new material. Hayes-Roth and Thorndyke [9] have shown that how well information is integrated into memory depends on whether or not two related pieces of information are present in working memory at the same time. So, as Gagné and colleagues argue [7], reminding students of information they know but are not thinking of at the time helps them to have related information active in working memory when it can be used to integrate new information.

Our Analysis of Collected Data

In order to understand more about the way humans rely on this pedagogical strategy, we collected a corpus of naturally occurring interactions in which a human tutor critiques a student's problem-solving behavior on an electronics troubleshooting task [13]. To date, we have collected data from 24 student-tutor interactions with 14 different students and 3 different tutors. This corpus contains approximately 1725 sentences in approximately 232 question/answer pairs. Our analysis of this data led to two observations about the way in which tutors and students exploit the previous dialogue.

First, we found that tutors frequently refer to what they have said in a prior explanation, and that often these references were used to point out similarities or differences between the material currently being explained and material presented in earlier explanation(s). For example, consider the following excerpt from our data:

> STUDENT: Don't I need to test pin 36?
> TUTOR: You might have to, but *for the same reasons given when you tested pin 28*, it is generally more efficient to test the main control data signals first.

In this example, in order to explain why it may not be necessary to test pin 36 the tutor refers back to an explanation given when assessing the test of pin 28, and states a generalization explaining why these two actions are considered suboptimal, i.e., that the main data signals should always be tested before secondary data signals. The tutor expects the student to be able to make use of the explanation given about testing pin 28

*This author is supported by a grant from the French institute I.N.R.I.A. (Institut National de Recherche en Informatique et Automatique).

†Dr. Moore's contribution to the research described in this paper was supported by the Office of Naval Research, Cognitive and Neural Sciences Division, and a National Science Foundation Research Initiation Award.

CHI94-4/94 Boston, Massachusetts USA
© 1994 ACM 0-89791-650-6/94/0016...$3.50

by indicating that it is relevant to the current situation ("for the same reasons given ..." serves this purpose). Referring to the prior explanation in this way forces the student to consider how the two situations are similar. Pointing out this similarity may facilitate the student in forming the domain generalization and recognizing how the two instances fit this generalization.

In the tutorial interactions we collected, the most commonly asked question is a request to justify the tutor's assessment of a student action (42% of all questions asked). We found that 27% of the answers to such questions involved references to previous justifications of assessments in order to point out similarities or differences. However, it is important to note that not all justifications of assessment provide opportunities for referring to previous explanations. We found that human tutors explicitly referred to a prior justification (as in the example above) in 73% of the cases where a sufficient similarity existed between the current action being explained and a previously explained action (see [18] for details). Therefore, human explainers refer to previous explanations in the vast majority of the cases where it makes sense to do so, at least when answering this type of question.

The second observation we made is that students ask questions that refer to the tutor's prior explanations, mainly to request a comparison between the current situation and one previously mentioned in the dialogue. This is shown in the following exchange from our data, in which the tutor assesses a measurement the student has performed:

> TUTOR: You made the following measurement: VDC test on A1A3A15 between pin 32 and ground. This step could have been improved.
>
> STUDENT: Is this step bad for the same reasons as testing pins 30 and 28?

Thus, we observed that both the tutor and the student refer to prior utterances: the tutor refers to past explanations in order to point out similarities or differences to a prior problem-solving situation. The student refers to prior explanations in order to ask questions about how prior problem-solving steps relate to the current step. These observations led us to provide facilities that allow both the system and the user to make use of the dialogue history.

VISUALIZING, QUERYING AND MANIPULATING THE DIALOGUE HISTORY

There are at least two ways of applying what we have observed about human tutorial dialogues in order to improve computer-based tutoring systems. One approach is to attempt to reproduce the human tutor's behavior, that is to provide the tutoring system with the ability to alter its natural language explanations to take into account any relevant prior explanation(s). This requires a detailed model of how human explanations are affected by prior discourse. Basic research aimed at developing such a model is being pursued by other members of our research group [2, 18]. Note that this approach attempts to make human-computer interaction (HCI) mimic human-human interaction (HHI).

An alternative approach, and the one we have chosen to explore, is to abandon the assumption that HCI is best if it mimics HHI, and instead to exploit the unique attributes of the computer that are useful for providing the capability to use the dialogue history in ways we have observed. In particular, we can make use of the computer's large and perfect memory, its ability to record and redisplay any portion of the prior dialogue, its ability to highlight the text of interest, and its ability to allow the user to manipulate the dialogue history as a text object via scrolling, selecting, etc.

The basic idea behind our approach is to treat the previous dialogue as an information source that can be directly manipulated by both the system and the user, just as any other text object. This approach frees the dialogue participants from having to refer to prior explanations in precisely the ways that natural language allows. Using our approach, not only can the system refer to previous explanations, but it can allow the user to *visualize* what has been said. We believe that this approach will have many of the benefits that are gained by the ability to refer to previous dialogue, without requiring us to solve many difficult basic research problems in computational discourse.

Very briefly, our interface works as follows. When the system refers to a previous explanation, it scrolls the dialogue history to the appropriate point, displays the portion of the dialogue centered around the relevant prior explanation, and highlights the prior explanation to which it is referring. When students want to ask a question about a prior part of the dialogue, the system allows them to point to the portion of the utterance they wish to ask about. Once they have selected the prior utterance they are interested in, they can choose from among several buttons depending on the type of question they wish to ask about a prior explanation. We now describe the main features of this interface in more detail.

A Concrete Representation of the Dialogue History

Our approach can be viewed as a way of giving a representation to the dialogue history. In spoken conversation, utterances are evanescent. Once they have been uttered, they have no concrete representation and therefore referring back to a prior utterance can be difficult or confusing [1]. The ability of the computer to store and display information allows the dialogue history to be reified. We believe that providing a visual representation of the dialogue history facilitates the referring process: participants know without ambiguity what they are talking about because they can see the text of the utterance to which they are referring.

Moreover, using such a representation, it may be possible to refer to portions of the dialogue that would be much more difficult to refer to using purely linguistic techniques, effectively increasing the amount of the prior discourse that can be exploited in subsequent explanations. For example, in a Wizard of Oz study in which the dialogue was displayed on the screen, researchers found that subjects were able to refer to utterances that appeared far back in the conversation, and they assumed that this effect was due to the dialogue being displayed on the screen [6].

Visualizing what has been done before has been used successfully in several previous systems, e.g., to allow users to

browse, undo and redo actions performed with a graphical editor [10], or to provide a graphical record of a student's solution to a problem [17].

Of course, visualizing a large amount of information such as a dialogue history requires interface tools that permit the user to access it. Visualizing the dialogue history is not simply a matter of displaying its content. The system must have strategies for determining what to display in different situations, and must provide tools that allow the user to browse the dialogue history effectively. We describe these next.

Querying the Dialogue History

As noted above, analysis of human-human tutorial dialogues reveals that students sometimes ask questions about previous explanations.

> *"Is this step bad for the same reasons you gave when I tested pin 19?"*
>
> *"Could you please elaborate on that part about testing signal flow inputs before data inputs?"*

If we allow the user to ask such questions in natural language, they pose a difficult problem for natural language interpretation systems. The difficulty arises because such questions make reference to items in the domain of discourse as well as to the discourse itself, so that the natural language analysis system must be capable of understanding both comments made at the domain level and comments made at the discourse level. Further, as the first example question above shows, the two levels may be intermixed in a single question. These referential problems are beyond the capabilities of current natural language understanding systems (but see [16, 11] for research aimed at solving this problem in task-oriented dialogues). However, if students are allowed to point to the portion of the previous explanation they wish to ask about, these difficult referential problems can be avoided.

Moreover, in a study comparing direct manipulation to natural language, Cohen admits the ambiguity of natural language, but argues that one of the problems with direct manipulation techniques is that they can only be used for information that is represented on the screen [4]. By providing a visual representation of the dialogue history, we overcome this limitation and facilitate users in asking follow-up questions. Our work can be viewed as an extension of the work reported in [14], which allowed users to point to a portion of the last explanation given by the system and to ask a limited set of questions (e.g., "Why?", "How", "What is an ...?") about the highlighted text. In our work, we allow the user to go back and point to a portion of *any* prior utterance, and we allow the user to request comparisons of situations reported in prior explanations. In addition, our system chooses when to make references to prior explanations in order to point out similarities and differences between the current and prior situation(s).

Direct Manipulation of the Dialogue History

In our system, we treat the dialogue history not only as information that can be displayed, but also as information that can be manipulated. Of course, the dialogue history itself cannot be modified, because it is part of the past. But its representation on the screen can be customized according to the user's needs, e.g., the student can choose to display only a part of each previous exchange, or can request that only certain actions be displayed, etc.

Psychological Rationale for these Facilities

As noted above, a requirement for learning is that some connection be established between new and prior knowledge. The idea of the tutorial interaction we have designed is to make the previous information accessible in order to give students the ability to draw links between the concepts that have been taught. As claimed by Collins and Brown [5], computers offer a new way of learning since they can keep track of the actions used to carry out a given task. The recording and replaying of the processes people use to perform a task has the capability to make these processes objects of reflection. Allowing students to examine, query and manipulate the dialogue history is expected to aid them in drawing analogies and forming appropriate abstractions that have been shown to be effective in learning [8].

In our system, we provide the student with two sources of information: (1) comments and explanations about the student's current problem-solving step, and (2) a visual representation of the dialogue history and a set of tools for accessing it. The interaction window on the screen is divided into two parts. The lower pane contains the answer to the user's current question, whereas the upper pane is a window containing the dialogue history. A screen dump showing a typical interaction appears in Figure 1.

OUR APPLICATION

Now, let us describe how the conceptual ideas discussed so far apply within the framework of our application. We are doing this research in the context of the SHERLOCK system, an intelligent apprenticeship environment that trains avionics technicians to troubleshoot complex electronic devices [12]. SHERLOCK's pedagogical strategy is composed of two stages. In the first stage, the student must isolate the faulty component in a simulated electronic device. A solution consists of a sequence of actions which are mainly measurements on electronic cards and replacement of components. During a problem-solving session, SHERLOCK's intelligence is used mainly to provide hints in response to student requests, rather than to intervene actively. After a problem has been solved, trainees enter the second stage in which they can review their problem-solving performance with SHERLOCK's Reflective Follow-Up (RFU) facility. Using this facility, students replay their solution one step at a time. SHERLOCK assesses each step, indicating whether the step was "good" or "could be improved". The system then justifies this assessment, by explaining why the step was considered "good"/"could be improved". If the step could be improved, the system explains what an expert would have done.

In the current version of the system, SHERLOCK steps through the student's actions, one after the other, in a sequential fashion. The student receives comments about each action he has performed, and is allowed to click on a button to go to the next action, and so on. In our work, we designed an interface based on the ideas presented so far to improve this

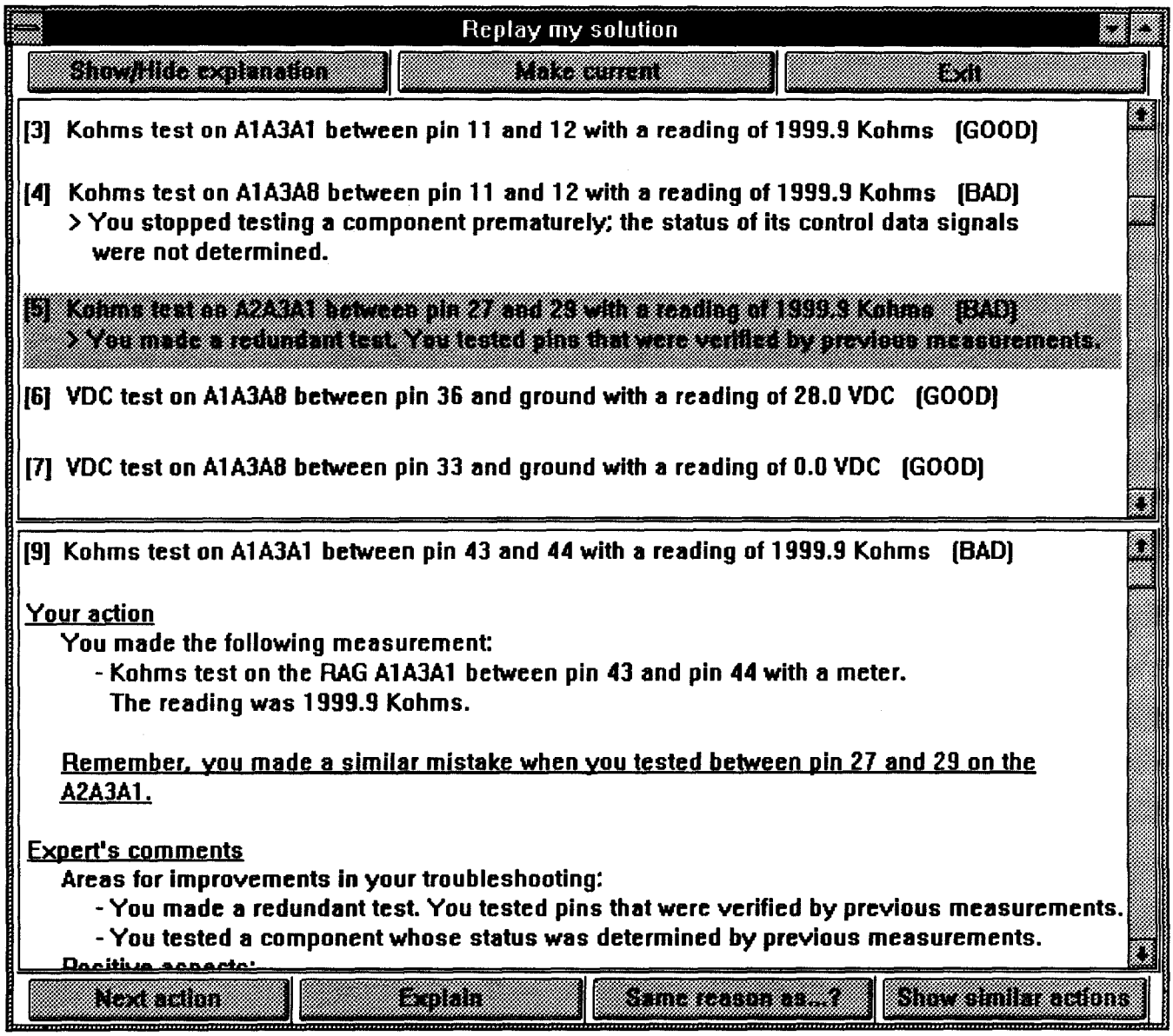

Figure 1: Screen dump of our interface

tutorial interaction. This interface has been built on top of the existing SHERLOCK system.

In the sections that follow, we describe our interface in more detail. In particular, we discuss

- the use of the dialogue history by the system, which can refer to a relevant prior explanation when the opportunity arises;
- the use of the dialogue history by the user, who can inspect, query and manipulate the dialogue history.

REFERRING TO THE DIALOGUE HISTORY

The system makes use of the explanations stored in the dialogue history to supplement the current explanation (see lower pane of Figure 1) by referring to a relevant prior explanation whenever it is possible to do so (see upper pane of Figure 1). In order to do this, we use a case-based reasoning algorithm developed by Rosenblum and Moore [18]. Given a current action and a sequence of previous actions, the algorithm computes a partial ordering of previous actions based on the similarity of each prior action to the current action. It bases similarity decisions on a set of facets that SHERLOCK employs to evaluate student actions. These facets were derived from a cognitive task analysis aimed at identifying the factors that expert avionics tutors use in assessing student's troubleshooting actions [15]. Examples of facets are: *premature swapping of a component, Ohms test which should have been Voltage test, redundant test.* Each action performed by the student is characterized by a set of such facets. Using this algorithm, the system can determine which, if any, prior action(s) are similar to the current action. When more than one action is similar to the current action,

the algorithm ranks the actions based on the number of facets they share with the current action.

Initial results using this algorithm are quite promising. In a corpus of 8 student-tutor protocols involving 154 student actions, the algorithm correctly selected as most similar the same prior explanations referred to by the human tutor. In 3 cases, the algorithm suggested a similarity not used by the tutor. However, when presented with these similarities, our expert tutor judged them as correct and stated that explanations that explicitly pointed out these similarities would have been pedagogically useful. In all other cases in which the human tutor did not make reference to a previous explanation as part of an answer, the algorithm reported that no prior action was similar.

Using this algorithm, our system can locate the prior action that is the most similar to the current action and highlight the prior explanation that critiques the similar action in the visualization of the dialogue history. Note that the prior action is an instance of the same type of problem-solving error that the student has made in the current action.

The screen dump in Figure 1 shows such an example. The current action being explained is *Kohms test on the component A1A3A1 between pin 43 and 44 with a reading of 1999.9 Kohms* (lower pane). Part of the text which does not appear in the Figure explains why this is a redundant test (the output of the A1A3A8 is the input of the A1A3A1):

> *Previous actions have shown that the output signal at pin 11 and pin 12 of the RAG's A1A3A8 is bad. Assuming the circuit has only one fault, the signal path from pin 43 to pin 11 is not suspect, the signal path from pin 44 to pin 12 is not suspect.*
>
> *Since the input signal is bad, the problem is not in the A1A3A1.*

SHERLOCK has assessed this action as COULD BE IMPROVED, mainly because this is a redundant test. In addition to an explanation of this assessment, our system provides a reference to a previous action, which was assessed as bad for precisely the same reason as the current one. The explanation that critiques this prior action is shown in the upper pane and highlighted. We believe that referring to and displaying this prior explanation helps students to create a conceptual link between these two actions and allows them to form an appropriate abstraction, thus providing students with a better understanding not only of the current explanation, but also of the previous one, and the general notion of when a testing action is redundant.

NAVIGATING THE DIALOGUE HISTORY

In our interface, the dialogue history is displayed in the upper pane. This pane essentially acts as a window to the dialogue history, allowing the student to access any portion of the dialogue history by means of a scroll bar. As shown in Figure 1, the dialogue history records each action the student has replayed, and a brief summary of the explanation SHERLOCK provided for each action. We chose to display short summaries of explanations, rather than entire explanations so that more than one or two actions would fit in the dialogue history window. By providing more of the dialogue history in this abbreviated form, students are also able to see more of the context surrounding each action, and scroll through the dialogue history more quickly to find the portion they are interested in. In addition, by giving only a brief summary of the explanation associated with the prior problem-solving situation, we hope to encourage students to "self explain", i.e., to attempt to reconstruct the details of the prior situation, the reason the prior situation was suboptimal, and to compare the prior situation to the current situation. The "self-explanation effect" has been shown to greatly improve learning [3]. As described below, we also provide a facility that allows the user to go back to any previous action and ask that the system display the entire explanation associated with that action.

Let us now describe and justify the different options that are offered to the student for accessing the dialogue history.

Going Back to a Previous Action

The original version of Reflective Follow-Up in SHERLOCK allows users to display previous actions only by going through them in sequential order. Using our interface to the dialogue history, the student can go back to any previous action without stepping through all intervening actions.[1] To view the explanation associated with any prior action, the student simply selects this action in the upper pane (it then becomes highlighted) and clicks on the MAKE CURRENT button. The previous action is then displayed in the lower pane and the student can look at it again, ask for a comparison between this action and a previous one, or request that all similar actions be displayed.

For example, in Figure 1, if the user wants to see the entire previous similar explanation that the system is referring to, he can click on the MAKE CURRENT button. The 5th action, already selected, will then be displayed in the lower pane. In addition to other information pertaining to the 5th action, the explanation will contain the following text:

> *Previous actions have shown that the output signal at pin 11 and pin 12 of the RAG's A1A3A1 is bad. Assuming the circuit has only one fault, the signal path from pin 27 to pin 23 is not suspect, the signal path from pin 29 to pin 24 is not suspect.*
>
> *Since the input signal is bad, the problem is not in the A2A3A1.*

Displaying all the Previous Similar Cases

The SHOW SIMILAR ACTIONS button allows students to display in the upper pane all of the prior explanations describing actions that are deemed most similar to the current action.

[1] Note that both our system and the original SHERLOCK system do not actually roll back the state of the simulation to a prior state when the student asks to view the explanation related to a prior action. That is, we maintain only a record of previous actions, their facets and the corresponding explanations. While simple in theory, restoring the simulation to a previous state would require storing snapshots of the entire simulation state after each action, or reinitializing the simulation and recomputing the effects of each action. Both alternatives are in practice infeasible because of the large amounts of space or time that would be required.

This facility is intended to aid students in forming generalizations. Reminding students of all the instances of a principle and stating this principle should promote better understanding of the principle and when it can be applied than repeatedly stating the principle as each instance of it arises. For example, visualizing all of the previous instances of actions that were assessed as redundant tests should help the student form a general idea of what it means to be an redundant test.

Querying the Dialogue History

As noted above, it is important to allow the student to ask questions about previous problem-solving steps. In the current version of our interface, we allow students to ask for a comparison between the current action and a previous action that they have selected from the dialogue history.

In order to allow such a comparison, we have augmented the representation of the facets that SHERLOCK uses to assess students' actions. In the original system, facets were represented as a simple list, with no relationship between facets. In our system, the facets are represented in a generalization hierarchy in which leaf nodes represent the original facets and internal nodes represent generalizations of facets. For example, the two facets *premature swapping of a component because the input was not determined* and *premature swapping of a component because the output was not determined* are both linked to the more general concept *premature swapping*, which is itself linked to *premature action*, etc. To establish a comparison between the current action and the one selected by the student, we use a spreading activation algorithm to find the most specific node(s) that generalizes both actions. Given the two sets of facets characterizing the current action and the one selected by the student, our algorithm finds the closest common ancestor(s). It then uses the features stored at each such ancestor to form a natural language comparison expressing this generalization. For example, it can say: "*Well... in both cases you swapped a component prematurely*" or "*Absolutely right, in both cases, you made a measurement on a component off the circuit path*".

As noted in [7], learning is also based on a process called *organization*, which permits students to divide sets of information into general subsets. The facility we provide follows this idea of generalizing from specific cases to allow students to organize what they have learned.

Customizing the Dialogue History

In order to allow the student to make the representation of the dialogue history as concise as possible, and to control which explanations appear in the upper pane, the interface allows students to indicate that the summary of an explanation for any action should not be displayed. Selecting an action in the dialogue history and clicking on the SHOW/HIDE EXPLANATION button will hide the explanation corresponding to an action if it is currently displayed, and will show the corresponding explanation if it is currently hidden.

EVALUATION OF THE INTERFACE

As noted above, the design of our interface was based on phenomena we observed in naturally occurring instructional dialogues, as well as results from psychology that show that relating new knowledge to what has already been acquired by the student is an effective teaching strategy. However, to determine whether the interface we have designed is useful, we have begun conducting experiments with users. We defined three ways of evaluating the usefulness of the interface:

- Do the subjects use the interface?
- Do the subjects perceive the interface as useful?
- Does use of a system with our interface improve learning?

Thus far, we have conducted 10 experiments, lasting approximately 45 minutes each. For various reasons, we have not yet been able to evaluate the interface with Air Force avionics technicians. Instead, the subjects used in our evaluation were computer science students who had some familiarity with the goals of the SHERLOCK system as well as electronic troubleshooting. For this study, the subjects solved relatively simple troubleshooting problems, since their lack of background knowledge in the avionics domain made it impossible for them to tackle the harder problems available in SHERLOCK. But we believe that this should not affect evaluation based on our first two criteria. However, as we discuss below, it did prevent us from evaluating whether learning is improved with our interface.

Each problem-solving session consisted of approximately twelve actions. On the average, the system highlighted a previous relevant situation more than once per session. For purposes of the evaluation experiments we designed the interface to record all subject actions, e.g., mouse clicks, scrolling, etc. This allows us to monitor the frequency of usage of various features of the system. In 9 out of the 10 experiments, people used the scroll bar to look at previous explanations in the dialogue history. In addition, the two other functionalities of our interface that were used most frequently (from 2 to 3 times per session) were the capability to go back to a previous action and view its associated explanation, and ability to have the system provide a comparison between the current action and a previous one. The customization of the dialogue history, which allows students to make it more concise by hiding a particular explanation, was almost never used. We expect that this was due to the fact that the dialogue history did not become very long because the problems we picked for the experiments required a relatively small number of steps to solve the problem.

These experiments indicate that people use the tool. We also believe that they found it useful. Evidence for this claim comes from interviews we conducted with each subject immediately after the experiments. When we began our evaluation, we had also intended these experiments to help us determine whether students' learning is improved as a result of using our interface. We planned to assign subjects randomly to two groups. The first one would have performed the experiment with our dialogue history interface and the second group would have used a version of our system without the interface. SHERLOCK provides scoring software that keeps track of student's performance on each problem, and this can be used to assess performance change as the students solve a series of problems. However,

we have found that it is virtually impossible to conduct these experiments with subjects that are not Air Force avionics technicians. The task SHERLOCK teaches is very complex, and assumes familiarity with routine aspects of the avionics troubleshooting task. Using non-Air Force personnel would require that subjects spend prohibitively many hours learning background knowledge with the system in order to solve enough problems to demonstrate an effect on learning. We plan to conduct such an experiment with Air Force personnel in the future.

CONCLUSION

In this paper, we described an interface for tutorial interactions which is based on the idea that tutorial explanations are more effective if they are related to what has been conveyed previously. The dialogue history, which contains the previous interactions between the system and the user is represented on the screen and is used by both system and student. The system supplements the current explanation with a display of relevant prior explanation(s) when they exist. The user is allowed to inspect and query the dialogue history. Results from a preliminary experiment using the system indicate that the users do use the capabilities provided in the interface and that they find these features useful.

REFERENCES

1. Carenini, G., Ponzi, M. and Stock, O. Combining Natural Language and Hypermedia as New Means for Information Access. In *Proceedings of the Fifth European Conference on Cognitive Ergonomics* (Urbino, Italy), 1990.

2. Carenini, G. and Moore, J.D. Generating explanations in context. In *Proceedings of the International Workshop on Intelligent User Interfaces* (Orlando, USA), January 1993.

3. Chi, M.T.H., Bassok, M., Lewis, M., Reimann, P. and Glaser R. Self-Explanations: How Students Study and Use Examples in Learning to Solve Problems. *Cognitive Science 13*, 2, (1989), 145–182.

4. Cohen, P.R. The Role of Natural Language in a Multimodal Interface. In *Proceedings of Fifth Annual Symposium on User Interface Software and Technology*, (Monterey, CA), 1992, pp. 143–149.

5. Collins, A. and Brown, J.S. The computer as a tool for learning through reflection. In *Learning Issues for Intelligent Tutoring Systems*, Mandl & Lesgold (eds), Springer Verlag, New-York, 1988, pp. 1–18.

6. Dahlbäck, N., Jönsson, A. and Ahrenberg L. Wizard of Oz studies – why and how. In *Proceedings of the international workshop on Intelligent User Interfaces*, (Orlando, Florida, January), 1993, pp. 193–200.

7. Gagné, E.D., Yekovich, C.W. and Yekovich F.R. *The cognitive psychology of school learning*, Harper Collins, 1993.

8. Glynn S.M. Explaining science concepts: a teaching-with-analogies model. In *The Psychology of Learning Science*, Glynn, Yeany, Britton (eds), Hillsdale, NJ: Lawrence Erlbaum Associates, 1991, pp. 219–240.

9. Hayes-Roth, B. and Thorndyke, P.W. Integration of knowledge from text. *Journal of Verbal Learning and Verbal Behavior 18*, 1979, pp. 91–108.

10. Kurlander, D. and Feiner S. A Visual Language for Browsing, Undoing, and Redoing Graphical Interface Commands. In *Visual Languages and Visual Programming*, S.K. Chang (ed.), Plenum Press, New York, NY, 1990, pp. 257–275.

11. Lambert, L. and Carberry, S. A Tripartite Plan-Based Model of Dialogue. In *Proceedings of the Twenty-Ninth Annual Meeting of the Association for Computational Linguistics*, (Berkeley, CA, June), 1991, pp. 47–54.

12. Lesgold, A., Lajoie, S., Bunzo, M. and Eggan G. Sherlock: A Coached Practice Environment for an Electronics Troubleshooting Job. In *Computer Assisted Instruction and Intelligent Tutoring Systems: Shared Goals and Complementary Approaches*, Larkin, Jill, Chabay, Ruth (eds), Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1992.

13. Moore, J. D. What Makes Human Explanations Effective? In *Proceedings of the Fifteenth Annual Conference of the Cognitive Science Society*, Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1993.

14. Moore, J.D. and Swartout W.R. Pointing: a way toward explanation dialogue. In *Proceedings of AAAI'90*, pp. 457–464.

15. Pokorny, R. and Gott, S. The evaluation of a real-world instructional system: Using technical experts as raters. Technical report, Armstrong Laboratories, Brooks Air Force Base, 1990.

16. Ramshaw, L. A Three-Level Model for Plan Exploration. In *Proceedings of the Twenty-Ninth Annual Meeting of the Association for Computational Linguistics*, (Berkeley, CA, June), 1991, pp. 39–46.

17. Reiser, B.J., Friedmann, P., Gevins, J., Kimberg, D.Y., Ranney, M. and Romero A. A Graphical Programming Language Interface for an Intelligent Lisp Tutor. In *Proceedings of the ACM CHI'88 Conference on Human Factors in Computing Systems*, 1988, pp. 39–44.

18. Rosenblum, J.A. and Moore J.D. Participating in instructional dialogues: finding and exploiting relevant prior explanations. in *Proceedings of the World Conference on AI in Education*, (Edinburgh), 1993.