

AI, Multimedia, and Education

Roger C. Schank

Northwestern University
The Institute for the Learning Sciences
1890 Maple Avenue
Evanston, Illinois 60201 USA

Abstract

As large, multimedia databases, interactive digital video, and high-bandwidth fiber optic networks become a reality, a key challenge is harnessing this emerging technology to effect real changes in how people access information, particularly in educational applications. Meeting this challenge will require methods for organizing, retrieving, and presenting, in a way that people find useful and meaningful, information represented in a wide variety of formats. Artificial intelligence, particularly such areas as knowledge representation, natural language understanding, vision, and learning, will play a central role in addressing these problems.

1 Introduction

For years, the dominant educational paradigm in computer-based instruction has been what can best be described as the "page-turning" architecture (Schank, 1990). Computer programs that were intended to be educational invariably asked students to do little more than "press button for next page." The reasons for this are simple enough: Designers of educational software have generally been short on computer science expertise, and have been trying to implement ideas of "programmed learning" that predate modern computers.

A far better paradigm for educational software, indeed the best piece of educational software that has ever been produced, can be found in something that is often not even recognized as being a piece of educational software: the flight simulator. The idea behind the flight simulator—learning by doing—is an old one, certainly in practical terms, and has been the subject of explicit theoretical discussion at least as far back as Dewey. And although the flight

simulator predates the modern conception of multimedia computing, it is in fact multimedia in the common sense of that term. Even though it is not a completely formed piece of educational software, in that it lacks some key elements to be described below, the flight simulator can serve as a model of the kind of simulation-based instruction computers are capable of providing. The key point concerns passivity versus activity: Good educational software is active, not passive. Students ought to be *doing* something, not watching something. Multimedia computing offers a serious prospect for improving simulation-based instruction, but to date it has generally been far too passive in actual implementation.

Asking a student to "press button for next page" is a bad idea whether the page is text or video. As an example, consider the current-day remote-controlled television. In its weakest possible sense, such a television is a multimedia computer. The computer aspect in this case, if any, is actually rather trivial, since it merely allows the user to push buttons for new channels. The major drawback of television, viewed as a multimedia computer, is that the central database is *user-controlled* but not *user-directed*: Video is supplied by networks which do not respond to user commands. The overall purpose of the system, at best, is to enable access to many streams of continuous information and entertainment.

The transition to true multimedia computing requires vastly increasing flexibility at both the provider and user ends. Users must be given more interesting commands than just "change channel," and we must make the providers of video responsive to these new user commands. Further, and most significant in terms of what is often missing in multimedia computing, we must have a purpose, and that purpose must be the essence of the computer program that is in control of the situation.

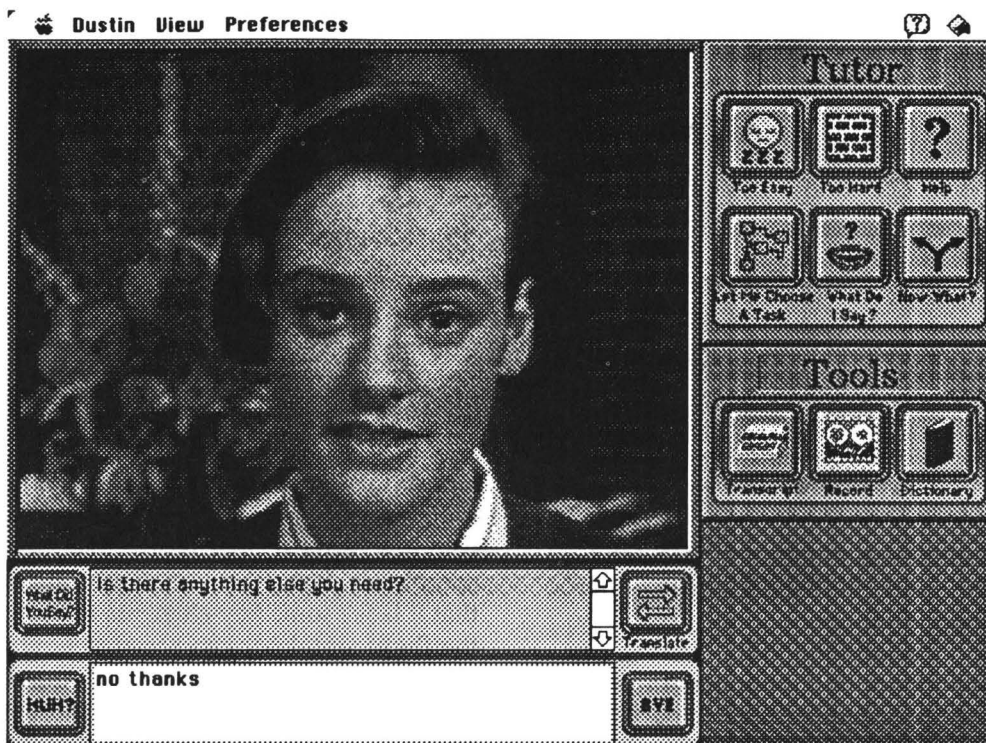


Figure 1. Dustin.

If the purpose is education and training, then the overall architecture of the multimedia system should be as follows (Schank, 1990): First, the system should incorporate a simulation of the task to be learned. Second, it should include an on-demand video database that can supply relevant video as needed. Third, controls must be given to the student user that allow him or her to control of the process. We shall consider each of these elements one at a time, in reverse order.¹

2 Button Controls

At Northwestern University's Institute for the Learning Sciences (ILS), we have been experimenting with a set of buttons that would enable users of educational software systems to ask questions, make requests, or communicate something about their mental state, and thereby to be in control of their own educational process. We are currently using the following set of nine buttons (Jona, Bell, and Birnbaum, 1991):

- Why
- How
- Huh
- What Next
- Big Picture
- History
- No Way
- Too Hard
- Boring

The purpose of these buttons is to allow a student to stop whatever is going on and find out what he or she needs to know. Providing the student with such a button set shifts control of the learning environment from the computer to the learner. This shift in control imposes strong demands on what needs to be in the database of the teaching program. In a video-based multimedia environment, in particular, it is necessary to have anticipated the questions that might naturally arise (and when and where), and to have videotaped and catalogued the answers to these questions.

Figures 1 and 2 present examples of two screens from programs built at ILS that use variants of this button set. **Dustin** (figure 1) is a program that teaches English as a foreign language (Ohmaye, 1992); **Victor** (figure 2) teaches basic telephone answering skills

¹ In what follows, all of my examples will be drawn from work being carried out at the Institute for the Learning Sciences. Many other projects, being carried out by my colleagues on the panel and elsewhere, could certainly be used to exemplify the same principles.

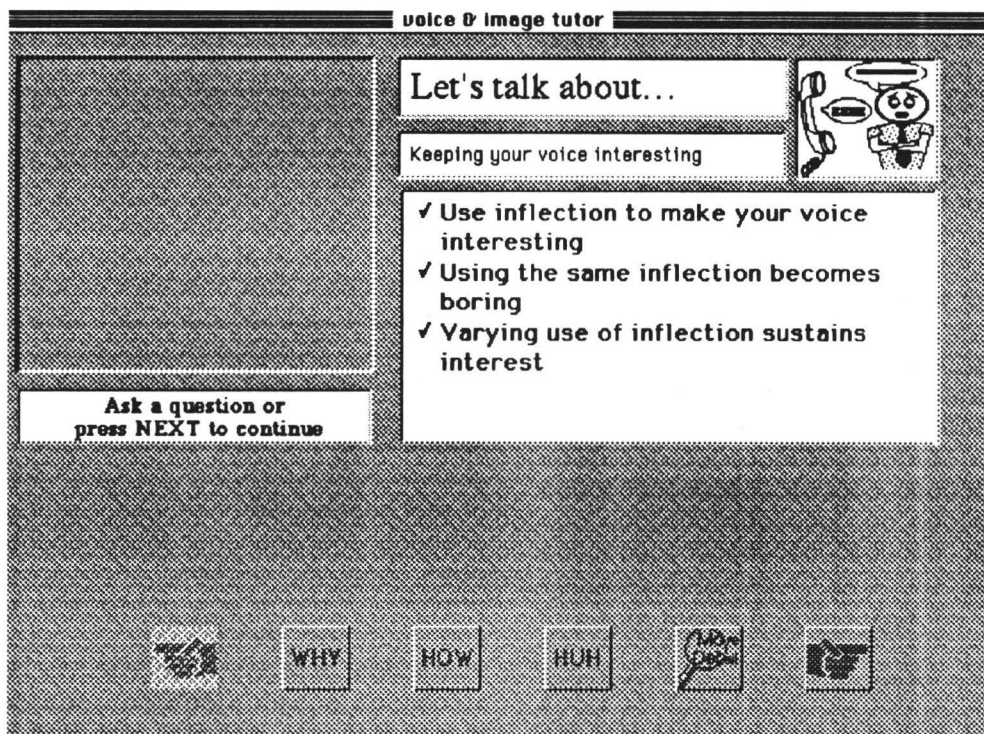


Figure 2. Victor.

(Feifer and Soclof, 1991). Both programs make considerable use of multimedia, mixing video, audio, and written text. More importantly for any multimedia program that purports to teach its user something, these programs provide active, participatory, environments.

Dustin allows users to converse with video images taken from real-life situations. For instance, one exercise in learning to speak English involves checking into a hotel. In Dustin, the user plays the role of a guest and interacts with the computer via keyboard and mouse. The system parses user inputs into requests for the available video that can most appropriately serve as a response to what has been typed. The buttons help the user by providing suggestions about what to say to the system, help in understanding what was said by the system, by allowing the user to change the task, and so on. In Dustin, the student is in control of the conversation in which he or she is participating.

The same is true of Victor, but an additional feature is present as well. Students perform various tasks and the computer relates their performance in the task to a video that explains how to improve their performance. At the point where the video pops up, various buttons activate as well, allowing the student to engage in a dialogue with the instructor by asking

Why, How, Huh, and so on, thus calling up additional video.

3 The Indexing Problem

Both Dustin and Victor take seriously the concept of interactivity, but doing that raises a central technical problem, namely the indexing problem, which must also be seriously addressed. The on-demand video database that is a defining characteristic of any strong multimedia program consists of a large set of video clips, probably only a minute or two in length, each telling a story. The ultimate goal is to allow a user to see and hear what he or she wants, when he or she wants, without necessarily knowing the informational content that is desired. When a person engages another person in conversation, he or she does not know what is in that person's memory, nor is it expected that a menu of ideas about which that person can talk will be provided. Further, he or she does not expect to be told that, at any time, more information can be requested, or the dialogue can be terminated. People expect more than this from one another and they will expect more from computers.

A number of approaches exist that would potentially enable humans to ask anything they want of a machine. First, and most obvious, is

equipping machines with some capability to understand natural language. This is an extremely difficult problem involving many issues, but the point I wish to emphasize here is that, if we keep our eye on the ultimate goal of facilitating effective communication, then the aspect of this problem that is most important is memory organization and indexing (Schank, 1982). Another way of putting this is that even if a computer understands what has been said to it, it cannot respond unless it has something to say.² We can, in principle, provide a multimedia computer with many very interesting things to say, by providing it with video clips of the most interesting people in the world saying what they have to say. But, after we have hundreds of thousands or millions of one- or two-minute clips that relate in various ways to previous clips, e.g., contradict what has been said in other clips, provide examples of what others were talking about, and so on, a very serious problem remains: finding the right clip at the right time.

Roughly speaking, there are two indexing problems. The first is the *absolute indexing problem*. A single video clip can be about many different subjects, making many different points. The "point of the story" is, in effect, the *name* of the story. To retrieve a story by absolute indexing, we must be able to ask for it by name and have the computer translate our request into an index that it can employ in memory search. The absolute indexing problem is extremely complicated. It requires builders of multimedia systems of any depth to have looked seriously at the issues of representation of knowledge—issues which have been a central focus of artificial intelligence research for some time.

The *relative indexing problem* is simpler and closer to being resolved (see, e.g., Ferguson *et al.*, 1992). The issue here is how to index video clips so that they relate to program states, in particular, to actions taken by a user, to buttons pressed by a user, and to clips just presented to the user. Options must be presented dynamically to the student, with the machine anticipating the possible questions that a student might need answered at any given moment and providing easy access to material that addresses those questions.

² Whether indeed it could even be said to "understand" in any sense without having some sort of response is debatable.

4 The Task

All of the above functionalities must be provided within the framework of an all-encompassing task that both motivates the student and guides the teaching (Bransford *et al.*, 1989; Collins *et al.*, 1989; Schank, 1992). The flight simulator would be a great model for this, if it were augmented by providing access to video clips of teachers commenting on what the student had just done and answering questions that arose. Furthermore, video of flight problems, their outcomes, and their causes, could be indexed in terms relevant to situations that arise in the simulator so that they could be discussed when the student encountered similar situations. Simulations of real business situations for corporate training, or activities like traveling or producing a television news show for children, allow for real teaching, provide a sense of accomplishment for the student, and motivate the student to pay attention to instruction, because it is needed to help him or her accomplish a goal. Complete systems such as these represent the real potential of multimedia computing in education.

In sum, the following observations about teaching and learning appear to me to be central guides to the production of high-quality educational software:

1. Students learn best doing, in the context of motivating tasks that are not easily accomplished.
2. Recognizing what to do is much easier than recalling what to do. Multiple choice must be eliminated.
3. Most interesting questions have no agreed-upon answers. Software should encourage students to explore a variety of possible answers.
4. Students respond to well-told stories. Software must have good stories to tell.
5. Computers should not be in control of the educational process. Students should have the power to determine what is coming next.

5 Concluding Remarks

As large, multimedia databases, interactive digital video, and high-bandwidth fiber optic networks become a reality, a key challenge is harnessing this emerging technology to effect real changes in how

people access information, particularly in educational applications. Meeting this challenge will require methods for organizing, retrieving, and presenting, in a way that people find useful and meaningful, information represented in a wide variety of formats. Artificial intelligence, particularly such areas as knowledge representation, natural language understanding, vision, and learning, will play a central role in addressing these problems.

Researchers in the three areas mentioned in the title—AI, multimedia, and education—are beginning to address the issues that emerge at their intersection. Some of the most important questions include:

1. When and how can multimedia presentation be used to significantly enhance the acquisition of knowledge, information, and skills by human beings, or to foster better communication between them?
2. How can large multimedia databases be organized to facilitate retrieval of relevant information, either by people or by computer? How can we build systems to automatically determine what sort of information might be relevant to a user's needs?
3. How can interactions with multimedia systems be structured to reproduce the advantages of a one-on-one conversation with a human expert—or to go beyond them?
4. How can we provide tools to content experts that will allow them to craft multimedia systems that are both cognitively and computationally plausible?

These questions are particularly timely for a number of reasons. Externally to AI, the hardware and software technology for multimedia systems has progressed rapidly in recent years, and even more rapid progress is to be expected in the near future. This technology has now reached the point where we can begin to see some of what it may be possible to do with it in the near future—and where the challenges and problems outlined above become real and pressing issues.

Considerations of the current state of AI also make the areas of multimedia and education particularly attractive as a source of problems and applications. It has been clear for some time that the construction of autonomous intelligent systems can no longer be the sole, perhaps even the principal, goal of artificial

intelligence if it is to continue to flourish as a field.³ Multimedia and education offer wonderful opportunities for AI research, both in terms of the interesting scientific and technical challenges raised, and the prospects for contributing to the solution of important real-world problems.

Finally, I wish to thank my colleagues on the panel—Glorianna Davenport (MIT Media Laboratory), Patrizia Ghislandi (Universita degli Studi di Milano, CTU Centro di Tecnologie per l'Apprendimento), James Spohrer (Apple Computer), and Kathleen Wilson (Bank Street College)—for sharing their perspectives on these issues.

Acknowledgments

This work was supported in part by the Advanced Research Projects Agency, monitored by the Office of Naval Research under contracts N00014-91-J-4092 and N00014-90-J-4117. The Institute for the Learning Sciences was established in 1989 with the support of Andersen Consulting, part of The Arthur Andersen Worldwide Organization. The Institute receives additional support from Ameritech and North West Water Plc, Institute Partners, and from IBM.

References

- Bransford, J., Sherwood, R., Kinzer, C., Hasselbring, T., and Williams, S. 1990. Anchored instruction: Why we need it and how technology can help. In D. Nix and R. Spiro (Eds.), *Cognition, Education, and Multimedia: Exploring Ideas in High Technology*, Erlbaum, Hillsdale, NJ.
- Collins, A., Brown, J., and Newman, S. 1989. Cognitive apprenticeship: Teaching the craft of reading, writing, and mathematics. In L. Resnick (Ed.), *Knowing, learning, and instruction: Essays in honor of Robert Glaser*, pp. 453-494, Erlbaum, Hillsdale, NJ.
- Feifer, R., and Soclof, M. 1991. Knowledge-based tutoring systems: Changing the focus from learner modeling to teaching. *Proceedings of the 1991 International Conference on the Learning Sciences*, Evanston, IL, pp. 151-157.

³ Schank (1991); this point has also been argued, from substantially different viewpoints than my own, by Winograd and Flores (1985), and Fischer (1987), among others.

Ferguson, W., Bareiss, R., Birnbaum, L., and Osgood, R. ASK systems: An approach to the realization of story-based teachers. *Journal of the Learning Sciences*, vol. 2, pp. 95-134.

Fischer, G. 1987. A critic for Lisp. *Proceedings of the Tenth IJCAI*, Milan, Italy, pp. 177-184.

Jona, M., Bell, B., and Birnbaum, L. 1991. Button Theory: A taxonomy of student-teacher communication for interface design in computer-based learning environments. *Proceedings of the Thirteenth Cognitive Science Conference*, Chicago, IL, pp. 765-769.

Ohmaye, E. 1992. Simulation-based language learning: An architecture and a multimedia authoring tool. Technical report no. 30, Northwestern University, The Institute for the Learning Sciences, Evanston, IL.

Schank, R. 1982. *Dynamic Memory: A Theory of Reminding and Learning in Computers and People*. Cambridge University Press, Cambridge, England.

Schank, R. 1990. Teaching architectures. Technical report no. 3, Northwestern University, The Institute for the Learning Sciences, Evanston, IL.

Schank, R. 1991. Where's the AI? *AI Magazine*, vol. 12, no. 4, pp. 38-49.

Schank, R. 1992. Goal-based scenarios. Technical report no. 36, Northwestern University, The Institute for the Learning Sciences, Evanston, IL.

Winograd, T., and Flores, F. 1986. *Understanding Computers and Cognition: A New Foundation for Design*. Ablex, Norwood, NJ.