Using Stories to Enhance and Simplify Computer Simulations for Teaching

Richard G. Feifer
Thomas R. Hinrichs
The Institute for the Learning Sciences
Northwestern University
1890 Maple Ave
Evanston, IL 60201
feifer@ils.nwu.edu
hinrichs@ils.nwu.edu

Abstract

Computer-based simulations are a valuable teaching tool because they permit a learner to explore a phenomenon on his own and to learn from his mistakes. Two factors, however, limit the use of computer simulations in teaching: good simulations are hard to build and learners can flounder with just a simulation.

We have built HeRMiT[1], a case-based tutor that integrates a simulation with a library of videotaped stories. The stories make up for any lack of depth or fidelity in the simulation by facilitating the generalization and application of underlying principles.

Introduction

The best way to learn how to do something is to try to do it and learn by your mistakes. Computer simulations allow you to make mistakes in situations that would normally be dangerous or expensive (i.e., flying a plane or disarming a bomb). Two factors, however, limit the use of computer simulations in teaching:

1. good simulations are hard to build;
2. learners can flounder with just a simulation.

Building a good simulation in any but the most trivial domains is difficult. Allowing the user a wide range of actions and simulating the results of any combination of those actions requires a complete model of the domain.

Even if we could build a simulation with sufficient fidelity, it is difficult to learn from mistakes without some guidance. Failure only shows you what not to do. It is often not obvious why you failed, in what range of circumstances the actions you took would lead to failure and what you should have done instead.

One solution to the floundering problem is the addition of a computer-based coach to watch over the learner's shoulder and advise (Burton & Brown, 1979; Goldstein, 1979). The coach, however, also requires a good model of the domain, plus a model of the learner. Even if we could model a learner's misunderstandings in a simulation, we still have the problem of generating a dialog to ameliorate those misunderstandings.

We have built HeRMiT (Human Resource Management Tutor), a case-based tutor (Feifer & Soclof, 1991; Schank, 1991) that teaches without true fidelity in a simulation, without a learner model, and without computer generated responses by adding human stories (Bell & Feifer, 1992). Instead of the computer generating instruction, good story tellers, experts in the domain, tell their stories on video tape. These stories are then indexed to the kinds of failure for which they are relevant.

These stories make teaching through simulations more practical in three ways:

1. the simulation need only provide a context and motivation for the story, the stories make up for any lack of depth or fidelity in the simulation;
2. it is easier to index failures than to model the learner sufficiently to provide intelligent coaching;
3. it is easier to show a video than to generate instruction, and more compelling to the learner;

Supplementing Simulations

In a case-based tutor we begin with a simulation of the task we want the learner to accomplish. The

[1]HeRMiT was built in cooperation with the Professional Education Division of Andersen Consulting.

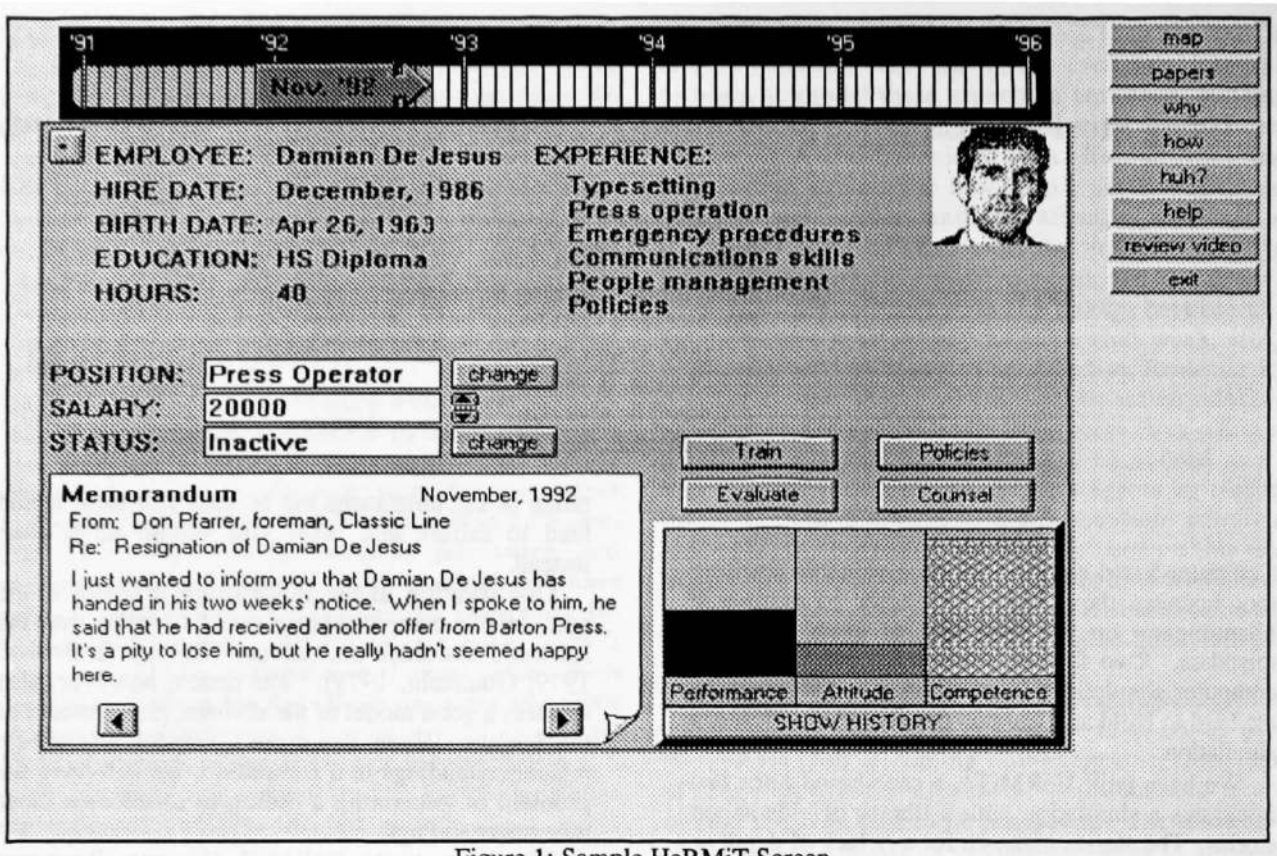

Figure 1: Sample HeRMiT Screen

interface to the simulation provides the learner with a range of actions that would be possible in the situation. The simulation provides immediate feedback: what might happen in the real world if the learner took the same actions?

HeRMiT's goal is to teach the basic issues involved in managing human resources, and to convince the learner that there is a connection between the manner in which a company manages its human resources and the company's bottom line. To accomplish this goal the learner is asked to manage the human resource function of a simulated company (figure 1).

The *Position, Salary, Training, Evaluate, Counsel,* and *Status* buttons allow the learner to take actions that would be available to a Human Resource Manager in a company. The bar charts in the lower right indicate the results of any of those actions. In addition there are meters on the main screen that reflect the company's overall productivity and morale as a result of actions taken on individual employees.

Simulating Human Resource Management

To be a useful teaching device, the HeRMiT simulation had to be broad enough to illustrate basic principles, such as the 'Peter Principle' (promoting someone to their own level of incompetence) and the 'Hawthorne Effect' (employees tend to be more motivated when management pays attention to them).

It was also critical to capture individual differences among employees. Human resource management is not formulaic — there are no cookbook solutions to human problems. To emphasize the importance of paying attention to the individual, we represent employees as differing in their levels of dedication, aptitude, ambition, experience, and education. These factors determine how quickly a new employee learns his job, how effectively he performs, and how rapidly his expectations of salary and position will grow. Accounting for individual differences is especially critical when making hiring, promotion, and salary decisions.

The simulation also had to be deep enough to emphasize the nature of fundamental tradeoffs in personnel management. For example, paying high salaries may help employee morale, but it will eat into corporate profits. Likewise, promoting people when they are ready may be best for the individual, but it may also lead to top-heavy management and large payroll costs. HeRMiT forces the student to balance these considerations.

Despite these complexities, the simulation is not particularly sophisticated. We were able to restrict the range of phenomena it covers (the breadth of the simulation) by restricting the variety of its inputs and

outputs. The inputs to the simulation are the discrete actions that the student can take: changing an employee's salary or position, training the employee, evaluating, counseling, firing and hiring an employee. The outputs of the simulation are the values of a few parameters: the attitude, competence, and performance of each employee and the overall morale and productivity of the company. These few parameters still permit the simulation to be open ended because the student controls the timing and magnitude of his actions.

Learning From Failure

What happens once the learner's actions have led the simulation to a negative outcome? In Figure 1, for example, the memo in the lower left informs the learner that a valued employee has just resigned. The resignation is not a good thing, indicating that the learner in some way mis-handled this simulated employee. The learner is wondering: 'Why did he quit?' or 'What could I have done to avoid it?' In this particular case the employee quit because she had become overqualified for her current position and the learner did not promote him.

Relying on the Simulation

Ideally, the learner forms a hypothesis for the cause of the negative outcome and tests it in the simulation. Through trial and error the learner can find a correct path. Ideal here refers to both the learner and the simulation. The ideal learner has the motivation and the skills to form and test hypotheses. The ideal simulation provides accurate feedback for the full range and combination of learner actions.

Unfortunately HeRMiT's simulation is not ideal. One of the problems in developing social simulations such as HeRMiT is the availability and precision of theories of human behavior. Existing models of motivation and performance tend to be descriptive and statistical in nature and are often defined in terms of influence systems. To translate this into a generative model we had to specify behaviors more precisely than theory would warrant.

One possible solution to this problem is to use a 'non-deterministic' simulation in which the behavior of an individual obeys a probability distribution. For example, in our domain this would mean that an employee might or might not quit in a given situation. For this approach to be effective, however, the learner would have to generalize this behavior by running the simulation many times. Since we expect the learner to run the simulation only a few times, we felt that the complexity and unpredictability of a non-deterministic simulation would be unnecessary and undesirable.

Our solution was to build a deterministic simulation by translating qualitative influences into cumulative, decaying, and one-shot effects and scaling those effects to demonstrate (and somtimes exaggerate) the human resource management principles. For example, we model the Hawthorne Effect as a short-term boost in attitude after any positive management action (e.g., a raise, promotion, or evaluation) and a gradual, cumulative decline in attitude that begins after the employee has been ignored for too long (where the time of onset and rate of decline depend on characteristics of the individual employee). While this oversimplifies the real phenomenon, more precise theories are unavailable, and this model is sufficient to communicate the ideas we wish to teach.

Using Stories

If the simulation were to stand on its own, its shortcomings would be more readily apparent. However, because the simulation is one component of a case-based tutor, it successfully provides a motivating context for learning the theory of human resource management, a responsive environment for making H.R. decisions, and an opportunity to make mistakes and fail with impunity.

Once a learner has failed, he is thinking about the context of that failure. He is motivated to learn anything that might help to avoid this failure in the future. There is also a good chance that he will store any new information appropriately in his long term memory, such that he will be able to recall it in relevant future contexts. Such failures provide the teaching system an opportunity to tell good stories. Thus we do not correct a learner when he makes a mistake. Rather, we wait for the mistake to lead to a recognized failure and for the learner to attempt to diagnose the failure on his own. Stories in HeRMiT are told by charismatic storytellers, who lived through similar disasters. Each story contains a description of the first warning signs, a dramatic description of how the problem led to some horrible outcome and the lessons learned from the situation.

The learner viewing the screen in Figure 1 should be wondering what he did wrong. While still viewing the personnel folder, he can look for clues and may form a hypothesis. Once he closes the folder, the learner has satisfied the above criteria, and HeRMiT presents a story. In this case the story is about a mid-level manager who was due for a promotion. The company, however, needed her in her present position because of some special skills that she had. They explained the situation to her, and thought that she understood and was being a good sport. Six months later she quit to accept a higher position with a competitor and is currently making life miserable for her original company.

The story provides the learner with at least one more example, this one in the real world, of the type of failure they encountered in the simulation. To use the story the learner must generalize their failure to a class of failures. Instead of thinking about the mistake of "Not promoting someone who has been doing the same job for 4 years," they are encouraged to think about the more general issue of "Failing to meet an employee's expectation of growth within a job."

Since the failure described in a story will not perfectly match the failure that led to the learner's negative outcome, the story does not provide an "answer" to the learner's problem. Thus the learner must derive the general principles in order to adapt the warning signs or solutions indicated in a story to the simulation.

Indexing Stories to Failures

An effective case-based teaching system must tell the right story at the right time. Knowing the right time to tell a story turns out to be easy in HeRMiT because failures are only manifested in a few ways: when the company's productivity hits zero, when overall morale drops below a threshold, or when a good employee quits. The difficulty is in knowing the right story that will best explain the learner's error.

Picking the right story is difficult for two reasons. First, reconstructing the events that led to a failure is beyond the ability of HeRMiT's simulator. Second, failures often have multiple causes in this domain.

Tracing back from a failure to its causes is hard partly because there are no backward links from the outputs of the simulation to the inputs, and partly because the simulation does not automatically record its history of events. Even if it did, it would still be difficult to determine how far back to trace and how best to explain the failure in meaningful terms. For example, if an employee were to quit, is it because his attitude was low or because he was hired and not properly trained for the job?

Therefore, rather than trying to trace back from a failure to its causes, we record the learner's mistakes as they are made and before a failure actually occurs. To do this, HeRMiT maintains a list of typical Human Resource problems for which it has video stories, such as training too little and too late, underpaying, promoting too soon, promoting too far, and evaluating too infrequently. Each type of mistake is recognized by a 'demon' that checks whether its conditions are satisfied after each simulated month. When a demon recognizes a mistake, it adds it to a list of the learner's mistakes that could lead to failure. If a failure later occurs, HeRMiT considers any mistake that could have led to the particular failure to be a possible cause.

Unlike traditional intelligent tutoring systems, there is no learner modelling. HeRMiT models only causality within the domain, not between the learner's cognitions and the domain. We do not try to blame the failure on some belief the learner might have, which would be very difficult. Rather we blame the failure on an action we know the learner took.

The list of possible causes is used to choose an appropriate story to tell. The difficulty here is that there may be many factors that contribute to a failure. For example, one way the simulated company could fail would be if the learner hired the wrong applicant, neglected to train him, paid little attention while the employee's attitude declined, and then tried to correct the mistake by overpaying everybody. In this case, the most recent mistake is probably not the most critical.

Instead of choosing the most recent mistake, we rank the types of mistakes roughly in order of their importance from a Human Resources perspective. HeRMiT then selects the most critical mistake and presents its corresponding video story. The story doesn't necessarily explain 'the' cause of the failure, but suggests how one of the learner's mistakes may have contributed to the failure. This approach of indexing by mistakes is simple but effective for domains with stereotypical classes of mistakes and well defined failure modes.

Conclusion

We have built a tutor that combines two knowledge elements:

First, there is a simplified simulation of the human resource function of a small company. Users have reported that it is fun to use. They enjoy playing with the simulation, trying out different actions, seeing how long they can stay afloat.

Second, there are 40 minutes of indexed stories about actual human resource management disasters. People enjoy listening to the stories. Some people have even sat through all 40 minutes, from beginning to end.

As engaging as each of these elements is, however, neither standing alone would effectively teach the principles of human resource management. It is only by combining them and giving the computer the ability to find the right story for a failure, that we have a cognitively sound tutoring environment.

The first learner test of HeRMiT was conducted in April of 1992 with five subjects. Four subjects had no previous exposure to human resource management. After using HeRMit all learners expressed a new appreciation of the importance of human rescource management to the health of a company and were able to demonstrate an

understanding of the basic issues (through written and oral debriefing). At least as important, all subjects and the 30 learners who informally used the program reported that it was engaging.

Acknowledgements

This research was supported in part by the Defense Advanced Research Projects Agency, monitored by the Air Force Office of Scientific Research under contract F49620-88-C-0058 and the Office of Naval Research under contract N00014-90-J-4117, by the Office of Naval Research under contract N00014-J-1987, and by the Air Force Office of Scientific Research under contract AFOSR-89-0493. 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, an Institute Partner, and from IBM.

In addition to the authors, the following people are involved in the development of HeRMiT: Larry Langelier, Donna Fritzsche, Wayne Schneider, Michael Korcuska, Joshua Tsui, Leena Nanda, Cheryl Jindra, Matt Greising, Alan Nowakowski, and Don Jastrebsky. Special thanks to the HeRMiT storytellers: Bill Braemer and Tim Coan.

References

Bell, B. L., & Feifer, R. G. (1992). Intelligent tutoring with dumb software. In *Proceedings of the International Conference on Intelligent Tutoring Systems*, . Montreal:

Burton, R., & Brown, J. S. (1979). An investigation of computer coaching for informal learning activities. *International Journal of Man-Machine Studies*, 11, 51-77.

Feifer, R. G., & Soclof, M. S. (1991). Knowledge-based tutoring systems: Changing the focus from learner modelling to teaching. In *Proceedings of the International Conference on the Learning Sciences*, (pp. 151-157). Northwestern University, Evanston, IL: Association for the Advancement of Computing in Education.

Goldstein, I. P. (1979). The genetic graph: A representation for the evolution of procedural knowledge. *International Journal of Man-Machine Studies*, 11, 51-77.

Schank, R. C. (1991). *Case-based teaching: Four experiences in educational software design* (Technical Report No. 7). The Institute for the Learning Sciences, Northwestern University, Evanston IL.