The Design of History Mechanisms and their Use in Collaborative Educational Simulations

Catherine Plaisant, Anne Rose, Gary Rubloff*, Richard Salter†, Ben Shneiderman†*

Human-Computer Interaction Laboratory, Institute for Advanced Computer Studies,
Center for Engineered Learning Systems, *Institute for Systems Research, †Department of Computer Science
University of Maryland, College Park, MD 20742
http://www.cs.umd.edu/hcil
rms@cs.umd.edu

Abstract

Reviewing past events has been useful in many domains. Videotapes and flight data recorders provide invaluable technological help to sports coaches or aviation engineers. Similarly, providing learners with a readable recording of their actions may help them monitor their behavior, reflect on their progress, and experiment with revisions of their experiences. It may also facilitate active collaboration among dispersed learning communities. Learning histories can help students and professionals make more effective use of digital library searching, word processing tasks, computer-assisted design tools, electronic performance support systems, and web navigation.

This paper describes the design space and discusses the challenges of implementing learning histories. It presents guidelines for creating effective implementations, and the design tradeoffs between sparse and dense history records. The paper also presents a first implementation of learning histories for a simulation-based engineering learning environment called SimPLE (Simulated Processes in a Learning Environment) for the case of a semiconductor fabrication module, and reports on early user evaluation of learning histories implemented within SimPLE.

Keywords: HCI, scaffolding, simulation

1 Introduction

Football players review videotapes of their most recent game to search for examples of which plays worked well, or how their defense was bridged. Defensive and offensive coaches can analyze strategies and offer guidance about tactics. Trainers can extract exemplary plays for future instruction sessions and advisors, recruiters, or retired players can add their comments. A more dramatic review of activities takes place when air safety inspectors examine the flight data recorders following a plane crash. They too want to know what went right and what went wrong so that designers and pilots can learn to prevent disasters. Here again careful analyses and multiple commentaries from diverse experts can be applied.

Videotape and flight data recorders provide valuable technological assists in reviewing the past. The capacity to easily record activities could also benefit learners who use computers. An entire session could be captured, so that peers or mentors could analyze a student's work with simulators, educational software, or design tools. We conjecture that *learning histories* are useful because they encourage meta-cognitive processes, encouraging students to monitor their behavior and reflect on their progress (Carroll, Beyerlein, Ford, and Apple, 1996; Guzdial, Kolodner, Hmelo, Narayanan, Carlso, Rappin, Hubscher, Turns, and Newstetter, 1996).

Users of most computer systems have experienced the disorientation of not knowing how they got into their current state and therefore not knowing what to do next, or worse still, not knowing how to get out of it. Increasingly, graphical user interfaces offer UNDO capabilities so that users can walk backwards and retrace their steps, but this modest functionality could be greatly expanded with many benefits. Capturing and manipulating the history of user actions seems especially promising as an aid in learning environments.

Giving students access to a record of their actions will allow them to review their previous steps and:

- understand what they did
- correct an earlier step and replay the entire history
- save histories for later review and consultation with peers and mentors
- search histories for individual steps or sequences
- send annotated histories by email to peers and mentors
- combine histories into a collaborative project
- compose demonstrations or tutorials that guide other students through complex procedures
- create macros that repeat specified sequences for exploratory research

Histories can benefit the learner directly in a wide range of domains. They can help students and professionals improve their skills with digital library searching, word processing tasks, computer-assisted design tools, electronic performance support systems, and web navigation (Greenberg and Witten, 1988; Hill and Hollan, 1993; Wexelblatt and Maes, 1999). For example, one study of World-Wide Web users showed that 58% of all URLs had been previously visited (Tauscher and Greenberg, 1997), and therefore web navigation could benefit greatly from powerful history tools. Unfortunately, while some version of web navigation history is already included in most browsers, these are generally limited and ephemeral (although improved versions are under commercial development). Even in this simple case, design controversies abound because the strategy for producing a compact, meaningful list (in a linear, tree, or network format) is not apparent. Novel approaches to web visitation histories have been proposed, and many have been demonstrated to be beneficial (Wexelblat and Maes, 1997; Hightower, Ring, Helfman, Bederson, and Hollan, 1998; Greenberg and Cockburn, 1999).

Histories also facilitate collaborative learning, which typically involves students reviewing each other's work, or building on each other's contribution. When the students' productions consist of texts or drawings it is relatively easy to use traditional technologies such as email or bulletin boards to disseminate the products to be reviewed. But when the project being reviewed involves complex time-dependent sequences of actions (e.g. learning to conduct a physics experiment or an emergency procedure), it becomes difficult to review the work of others without re-running the sequence of actions. In this context a rich history record can be invaluable.

Learning histories seem especially potent for engineering simulations where the sequencing and timing of actions can have a dramatic effect on the simulated world, for example a fabrication or chemical process. The use of learning histories in engineering simulations will be the main focus in the rest of this paper. Our approach consists of making the record of actions available to the learner. It differs from the approach of traditional-computer assisted learning environments, which monitor and analyze user actions with the goal of categorizing user behavior and prescribing lessons or exercises. But instructor and student can respectively employ these traditional and nontraditional approaches to make use of history data for complementary purposes.

In this paper we describe the design space and challenges of implementing learning histories. We describe our first implementation of learning histories for a semiconductor fabrication module, implemented using a simulation-based engineering learning environment called SimPLE (Simulated Processes in a Learning Environment) developed at the University of Maryland. Finally we report on early user evaluation of the prototype.

2 Goals for research on learning histories

A learning-history mechanism must record sufficient information about user actions and system state to enable review of previous actions and their impact. To accomplish this requires an efficient internal representation of user actions accompanied by a well-designed user interface to present this information (Shneiderman, 1998). But even this level of functionality requires careful design, proper software engineering, and meaningful representations in order for users to be able to see the relationship between their actions and the simulation outcomes.

The desirable features of a robust learning history mechanism include: *first-class objects*, *adjustable replay speed*, *annotation*, *revision*, *search* and *macros*.

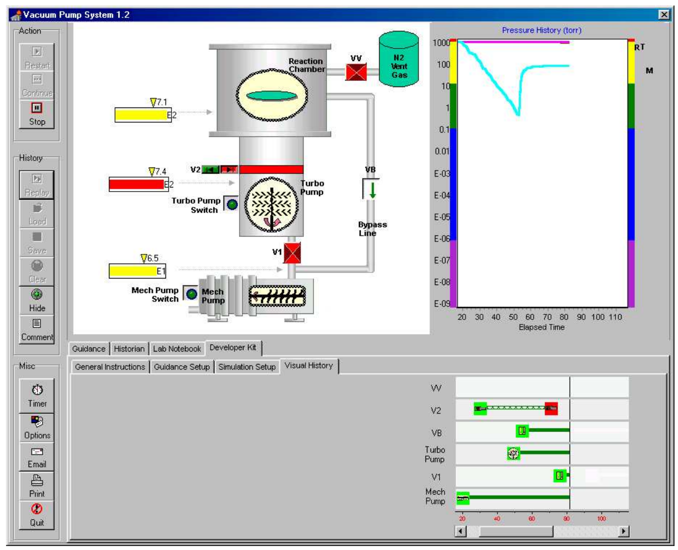


Figure 1: History recording. While users manipulate the controls of the simulation (top left section), actions are recorded (bottom right section). Comments can be added during or after the recording (see below).

2.1 First-class objects

A proper software engineering approach would promote learning histories as *first-class objects*; i.e. persistent data structures that exist independently of the processing environment in which they were created. Such histories could be saved, sent by email to peers and mentors (Figure 2), posted to a website, edited, extracted, combined, and searched. Instructors, for example, could challenge classes to submit action sequences that accomplished goals rapidly, with fewer steps, or with fewer resources. While the logs of traditional command-line systems often provide sufficient information to implement these services, generalizing to interactive GUI environments is considerably more difficult.

2.2 Adjustable replay speed

Just as videotapes can be replayed, stopped, or viewed with stop-action control, learning histories should be able to be replayed at normal speed, but also faster or slower. Stepping through critical sequences or going backward one step at a time would increase the benefits. The capacity to see the summary of a long sequence of actions and then zoom in on details also seems useful.

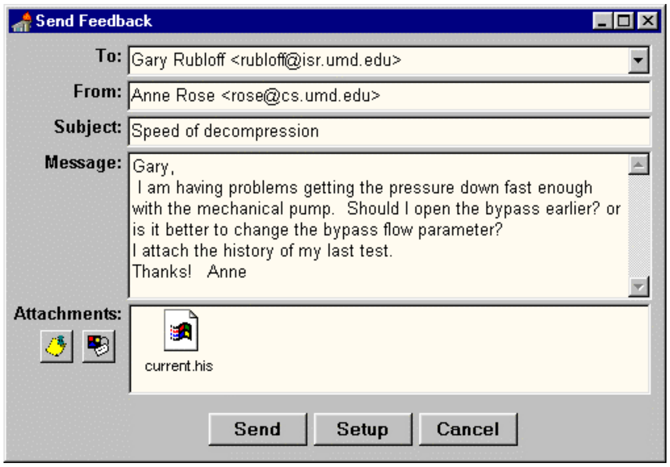


Figure 2: Mailing of history with question to the peer or mentor.

2.3 Annotation

Annotation is a critical component in communicating with others about the history. The learning environment might generate automatic annotations. When key events or accomplishments happen, a user might highlight or comment on actions taken, posing or answering questions, or relating them to other elements of the learning environment (e.g. referring to a valve before the valve is closed, and highlighting a graph to show the effect of the valve closing on pressure.) The annotated history can then be made available to others (Figure 2). Histories can also be complemented by audio recordings (e.g. recording users' comments describing a problem and the steps that led to it), or enriched by providing access to related tools (encyclopedias, web, calculators, use of guidance materials).

An annotation mechanism would also allow instructors to create demonstrations and tutorials conveniently. By carrying out a set of actions, the instructor could demonstrate common tasks, emergency operations, or even common mistakes. Instructor annotations can augment the history at key points in time. During student replay of the session each annotation will make a timely appearance, providing guidance for student comprehension, or directing the student's eye to particular events.

Depending on the group interaction style, annotation privileges may vary. Identifying students and permitting them to add comments seems a minimum requirement. Instructors may be able to edit the history to hide unwanted detail, while students may be prohibited from doing so.

2.4 Revision

While replays are informative, revisions to the history would enable the repair of problems and the capacity to try out alternatives conveniently. Sometimes revision to a single action can yield success, but more complex revisions must be anticipated. Revision is complex, because modifying one part of a history may invalidate later actions or produce surprising results. The degree of revision permitted will depend on the application and the analysis of the history.

2.5 Search

Search facilities could enable students to find an action, outcome, or pattern within a lengthy simulation sequence, or across several such sequences. Similarly, search facilities could enable instructors to find which of their students had used or misused a feature, achieved or failed to achieve a goal, and what patterns of work were most common. Specifying the searches is a non-trivial problem as the complexity of the history grows; consider, for example, specifying patterns that capture state relationships across large numbers of heterogeneous controls.

2.6 Macros

A more advanced learning history system would include the capacity to create and execute macros (Kurlander and Feiner, 1992). A user could carry out an action sequence and then add iteration and conditionals to create non-trivial programs to allow numerous simulations to be carried out rapidly. In a semiconductor manufacturing application, a student could vary the temperature or pressure over a large range and automatically run hundreds of simulations to see the impact on yield, resources, or time to complete. Macro facilities also have rich possibilities for further research, such as modularity and argument-passing strategies. Of course, additional features mean increased learning time, so a simplified strategy seems wise.

2.7 Extension to Experienced Learners

It should be noted, and emphasized, that the features of a learning historian can be beneficial to practitioners and experts as well, facilitating collaborative design, analysis and optimization. And since modeling and simulation lies at the heart of systems engineering design, features such as the macro capability in a historian could accelerate and/or automate complex sequences of design iterations for optimization.

3 Learning history design issues

Design tradeoffs abound in developing tools for learning histories. For example, sparse histories are easier to review and manipulate, but may contain too little information to be useful. A complete record of all events that occur during a session certainly would provide a sufficient database for any history function, but may be too large to be practical. Culling out minor actions, timestamps, and outcomes may or may not improve the situation. Can controls be designed to fine-tune the level of history capture to suit the application? Note that by capturing more information, the burden of screen management, zooming, panning, and filtering grows, but the additional benefit may be small.

Keeping track of histories of user actions is easy in command-line interfaces, such as the Unix C-shell, but it becomes more complex in graphical user interfaces (GUIs). History keeping is still more complex in simulation environments, where capturing the state of the process is difficult (Kurlander and Feiner, 1993; Cole and Tooker, 1996; Jones and Schneider, 1996; Nahvi, 1997). Device-level recording is simple (mouse click locations and keystrokes), however interface-level recording and task-level interpretation is strongly preferred because these are the meaningful components of interaction. In our vacuum pump simulation (Lu, Oveissi, Eckard, and Rubloff, 1996; Rose, Eckard, and Rubloff, 1998), it is hardly helpful to record that there was a mouse click at location (150,345) and then at (427,611); the appropriate level is to record that a valve was opened and a pump started. Adding the time stamp is easy, but recording the state of the simulation at that time can pose problems, unless the design accommodates efficient history recording. There is also a distinction to be drawn between deterministic and stochastic simulations: in the former, a limited number of user events will cause precisely the same system response, so the history model may be much more compact.

As user-interface designers and software engineers we are concerned about generalization across applications and scaling up to larger simulations. Can history data structures or visual representations be generalized across applications? Will control panels for replay, revision, and search be useful across many applications so user learning will not have to be repeated?

3.1 Benefits

Assuming we can build appropriate learning history facilities, then what benefits might we expect? We have three primary hypotheses:

- Students who can review and reconsider their work will have more productive learning experiences.
- Students who can send their work to peers and mentors for comment will explore more freely and ask for advice more often.
- Instructors who can monitor student histories will be more aware of student problems and learning module deficiencies.

These can be expanded into more specific hypotheses tied to specific application domains, varied student populations, and diverse educational situations.

4. Initial Implementation of Learning History

We implemented a learning history mechanism in the context of an application framework for constructing simulation-based learning environments called SimPLE (Simulated Processes in a Learning Environment, Rose et. al., 1998). Our long-term goal is to develop a learning history mechanism for this application framework that will allow all modules developed in SimPLE to generate and use learning histories. Our anticipated use for this initial history system was twofold: a) to provide the instructor with a demo-based tutorial composer (Munro, 1996) built on annotated histories; and b) to facilitate communication between instructor and student via recorded histories of student simulation runs. This initial implementation was designed with these goals in mind.

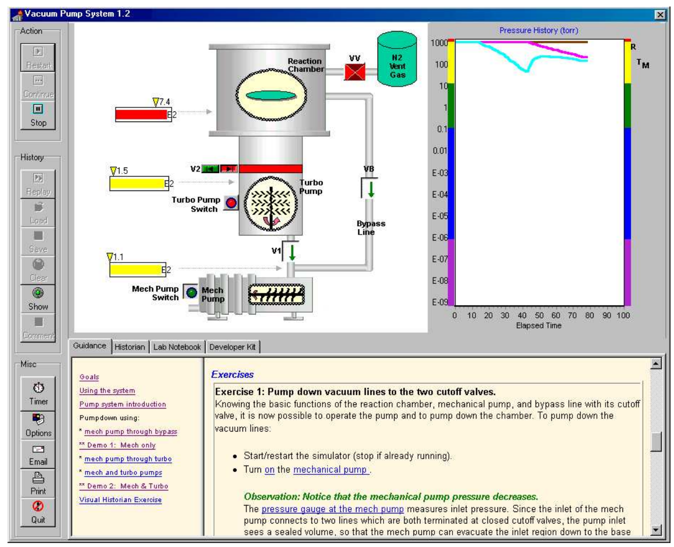


Figure 3: VacSim, an operational SimPLE module for learning the basic principles of vacuum pump technology as needed for semiconductor manufacturing.

4.1 SimPLE

Modules developed with SimPLE use dynamic simulations and visualizations to represent realistic time-dependent behavior and are coupled with guidance material. The software architecture enables independent contributions from developers representing educational content (e.g., domain knowledge in the form of simulation models and guidance materials) and software development (e.g., user interface). It provides a user interface template and accompanying software aids to reduce the software development effort. The simulation uses VisSim™, v. 3.0 and the front-end uses Delphi™,v. 4.0.

SimPLE modules have primarily dealt with process control simulations involving sequences of operations and continuous deterministic physical processes. Our main focus has been semiconductor manufacturing (Lu et al., 1996), but other applications include a simulation of the hydrology of the Nile River (Levy and Baecher, 1998) and traffic flow after diversion from major arteries (Plaisant, Tarnoff, Keswani, and Rose, 1999).

4.2 Learning history implementation

We implemented an initial learning history mechanism in the SimPLE simulation *VacSim* (see Figure 3). The VacSim module is designed to teach the basic principles of vacuum pump technology as needed for semiconductor manufacturing. The simulation emulates the configuration used to evacuate a reaction chamber; it operates with two pumps (mechanical and turbo) and four valves. Each input control has two states: on/off for the pumps and open/closed for the valves. Output data consists of total pressure measurements for the two pump chambers and the reaction chamber. The module teaches the procedure for producing extreme low pressure in the reaction chamber, through a sequence of actions that turn pumps on and off and control vacuum channels by opening and closing valves.

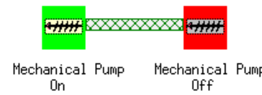
A learning history for this domain need only consider high-level time stamped events:

- **Active events**: actions on the simulation (e.g. open valve, turn on pump, etc.) or annotations (written comments or audio notes);
- **Reactive events**: messages (e.g. errors) or status reports (e.g. stable pressure reached, 50% of desired vacuum level reached);
- **Modeling events**: user modifications of simulation parameters.

In the minimal history model for a deterministic simulation (which is the case for all SimPLE simulations to date), there is no need to save the reactive events since they are generated when the experiment is re-run. This model, however, does not support general movement in time through the recorded simulation. For non-deterministic simulations, and to gain greater functionality, a history model may require a greater portion of the entire state of the simulation to be recorded, and this performance challenge will be explored later in the project. To simply reproduce a given session of the VacSim module (i.e. the reactive events) it is sufficient to record the state changes and parameter changes over time of the six input controls.

4.3 The Visual Historian

The *visual historian* (Figure 4) provides the means by which a user can interact with the history of a SimPLE session. It shows the parallel between control state and pressure values, making apparent the consequences of pump/valve actions on chamber pressures. In the bottom display of this figure, events changing the state of each of the six controls are pictured symbolically using green and red icons, for the *pump on*/*valve open* and *pump off*/*valve closed* types of events, respectively. Green lines show the period during which a valve is open or a pump is operating; the green line becomes crosshatched once the valve is closed or the pump is turned off:



The top display in Figure 4 shows pressure values at corresponding points in time. The two visual historian displays make it is easy to discern the current state of the system and the periods during which each control was "on", and to correlate control events with their effects on chamber pressures.

4.4 Historian features

The visual historian's annotation mechanism permits students and instructors to communicate about a particular session. Notes can be marked as questions or comments, displaying a different icon in each case. The student may add questions for the instructor as to why certain events occurred. The instructor can further annotate the student's history with responses to these questions. Alternatively, the instructor can build a tutorial demo out of a particularly instructive history by adding annotations to provide guidance during the replay.

Session histories can be replayed immediately, edited, or saved for later recall and replay. Once a history has been recorded, the visual historian functions as a *history editor*: Icons representing simulation events may be dragged left or right, causing their corresponding events to occur earlier or later, respectively, during the replay. Annotations may also be edited and repositioned. This feature proves invaluable during tutorial writing, as it allows the author to fine tune tutorial actions before the tutorial is saved.

4.5 Historian operation

During replay, a thumbnail of the session being replayed is displayed and correlated with corresponding outputs (Figures 1 and 5). Each annotation presents the text in a dialog box, which pauses the simulation until the dialog box is dismissed (Figure 5). In a subsequent implementation these "break-notes" will be supplemented by "side-notes", which will appear for a time without pausing the run.

New notes can be dropped on the timeline during the replay by pressing the "Add note" button. We assume that most learners will read or add notes during the replay, or afterward upon examination of the run's history. But advanced learners and instructors preparing demos and tutorials can use the visual historian to control more precisely the timing of the notes. The visual historian also seems the right place to control the speed and other options of the replay such as the highlighting on elements of the simulation (e.g. in the current implementation, when a control changes state during a history replay, it flashes brightly to attract attention).

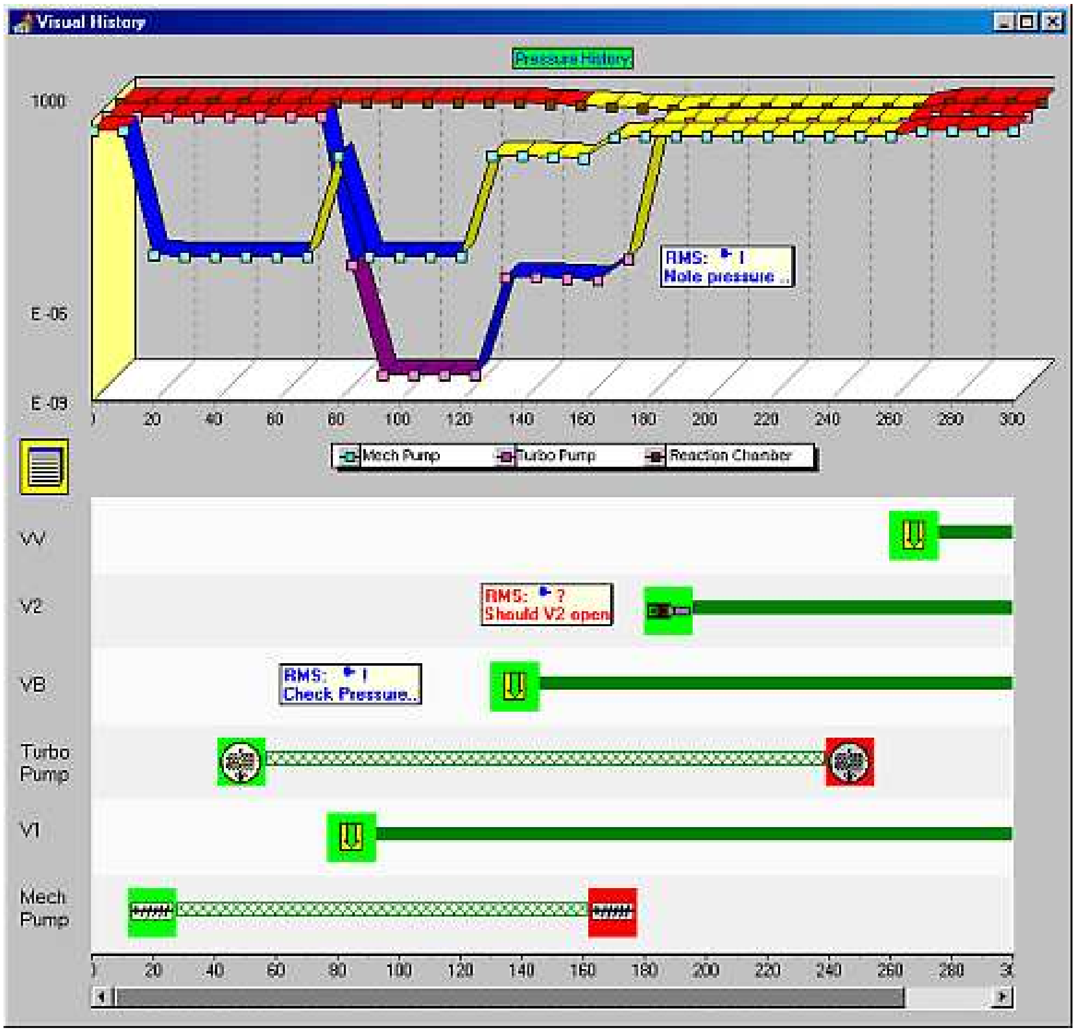


Figure 4: The visual historian with annotations by user with initials "RMS": The bottom half indicates user actions, while the top half shows corresponding pressure values.

Histories produced by the SimPLE visual historian can be saved with a name, sent by email (as in Figure 2), and loaded in the historian and replayed. A replay produces the same sequence of events, embellished by annotation break-notes and flashing control state changes. Thus to make a demonstration, an instructor simply records his sequence of actions, reviews it by replaying the history, fine tunes the event sequence by editing the history, adds notes, and sends it to the class.

While learners experience the demo, they can pause the replay, experiment on their own and resume the demo. They can add questions and re-send to the instructor. On their own, learners can start their own exploration, create their own complete simulation runs and record them using the historian, and then ask peers for comments, reviews or tests. The annotation tool is simple enough for first time users to feel confident using it, and powerful enough for advanced users or instructors designing complex tutorials. Adequately simple default options that do not burden novice users, and control panels that give users control over the level of details of the recording and the richness of the history are essential to a well-designed environment.

5. Pilot study

In order to test our early design and prototype, we ran an informative evaluation with five users. First, an instructor running VacSim prepared a tutorial containing two demos constructed using the historian. In one of these, the instructor used his expertise to operate the controls so that the optimal pressure level was reached in the reaction chamber (the goal of the exercise). In the other, he intentionally brought about a less than optimal outcome. Subsequently the instructor carefully examined the visual history (Figure 4) to recognize key events that required comment, and added text (as annotations to that history) to embellish the presentation.

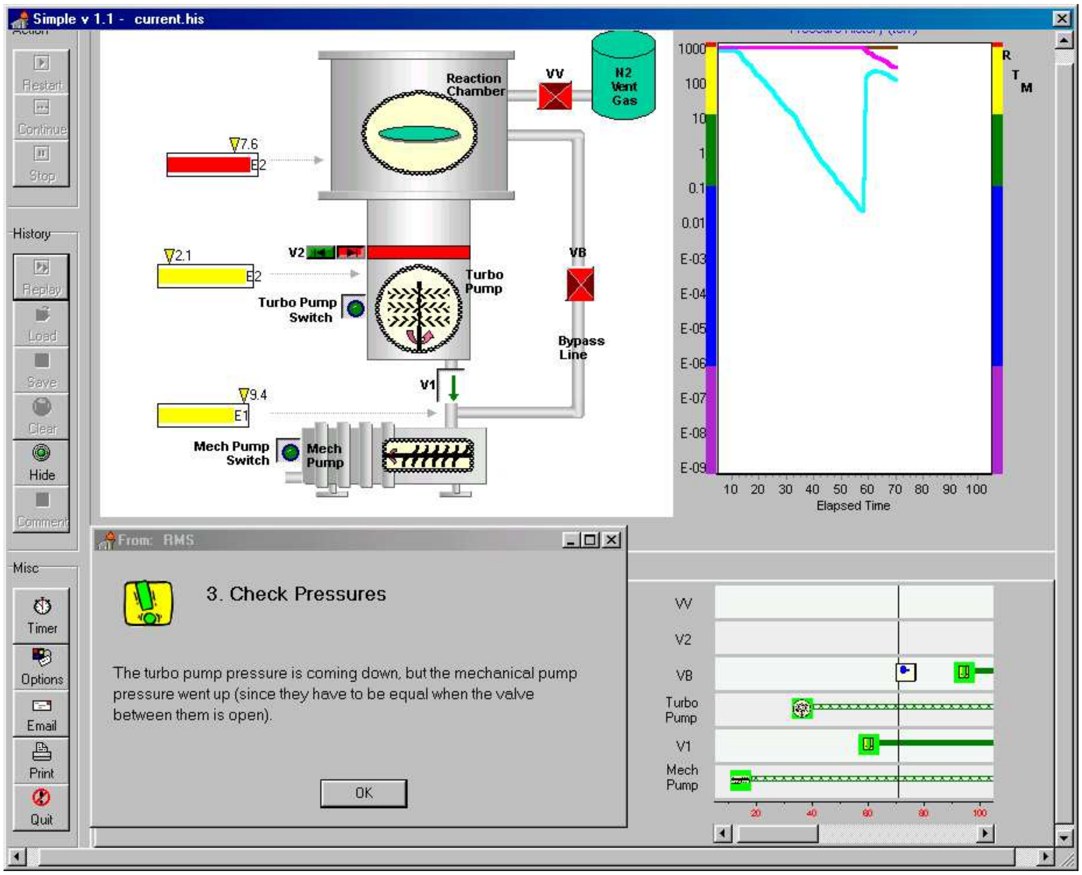


Figure 5: History Playback

Because the SimPLE modules are still being refined and appropriate classroom populations are not always available, we chose to use a mix of undergraduate and graduate students as our initial testers of this material. None of the users had used the learning histories before. They came from different backgrounds, and none of them knew anything about vacuum pumps and their use in semi-conductor manufacturing, making them learners of *both* the content and the history tool.

On the first round of review we sat with the five users and answered general questions about the system, while recording their comments and usability problems. We then asked each user to work through the short tutorial on vacuum pumps prepared by the instructor that includes the history-based demos. At the conclusion of his or her session each user was asked to email general comments and include at least one annotated history. Users were asked not to discuss the topic of the experiment "live" but only by email. At the end of the experiment we interviewed each user independently.

5.1 Results

We received an email with an attached history file from each of the five participants. Each email contained numerous comments (an average of 7 comments per participant), and in four of the five cases the attached history also contained annotations from the participant commenting on specifics of his or her simulation run. In one case, a participant's question was answered by returning his history to him with new annotations from the instructor. All of the participants found that the material produced by the history module embellished the presentation ("The demo was [definitely] useful. The notion of the historian is really neat. While I might be tempted to change how some of the tutorial material is arranged and organized… the demos seem fine... I had no trouble running or creating them..."). The most common request was to extend the history-keeping technology to allow more freedom of movement within the replay ("How can I back up in the demo?" "How can I back up history to a given point in my own simulation?" "It might be nice to have a fast forward function." "I would've liked to be able to jump to certain points on the history, rather than just being able to play it from the beginning."). A similar suggestion was made regarding support for allowing changes in the initial simulation settings prior to playback. ("[The] demo suggested to change settings ... I was under the impression that it was suggesting to change the settings in the history playback (which I was told is not supported at this time). I think it would be a good idea to allow this, although it may get complicated to figure out if you are editing a history file or an original simulation.")

Other usability factors mentioned more than once included: 1) clearer demarcation between live simulation and history replay ("[I] would suggest [a] more obvious indicator that you are in 'history playback mode'"); 2) real estate ("Why not put all of the history stuff on the main window?"); and 3) timing ("… the time between events on the demos should be shortened.").

6. Conclusions and future directions

From this first study we conclude that the students found the history mechanism valuable to their learning sessions, and would be eager to experience an even more powerful history keeping environment that would permit both selective replay and interaction. This portends a deeper investigation of tradeoffs between dataset size and capability in history models, but one which could prove useful in the simulation setting by providing added functionality.

The initial SimPLE visual history system described above will be generalized so that it can be included in the SimPLE framework. This will require the development of a set of components that can be configured to record simulation histories in a diverse set of domains. Currently these include semiconductor manufacturing, the hydrology of the Nile River basin, and highway traffic management. The broad spectrum of data and control patterns in these application areas will help us test the generality and completeness of our historian-building components.

We are encouraged by this initial effort that histories have great potential as tools in collaborative learning. We hope that future work will produce a general methodology for providing instructors and learners with the capabilities described in this paper as a natural component of their educational experience.

Acknowledgements

The authors greatly appreciate the assistance of Julia Lawall, Harry Hochheiser, Steven Salter, Maya Venkatraman and Zachary Friedman in this research and in the preparation of this manuscript.

References

Canizares, C. A. (1997). Advantages and disadvantages of using various computer tools in electrical engineering courses, *IEEE Trans. on Education 40*, 3, 166-171

Carroll, S., Beyerlein, S., Ford, M., and Apple, D. (1996). The Learning Assessment Journal as a tool for structured reflection in process education, *Proc. Frontiers in Education'96*, IEEE, 310-313.

Cole, R. and Tooker, S. (1996). Physics To Go: Web-based tutorials for CoLoS physics simulations, *Proc. Frontiers in Education'96*, IEEE, 681-683.

Davidson, N. and Worsham, T. (1992). *Enhancing Thinking Through Cooperative Learning*, New York: Teachers College Press.

Edelson, D. C., Pea, R. D., and Gomez, L. M. (1996). Computer support for learning though complex problem solving, *Communications of the ACM 39*, 4, 43-45.

Greenberg, S. and Witten, I. H. (1988). How users repeat their actions on computers: principles for design of history mechanisms, *Proc. CHI'88: Human Factors in Computing Systems*, 1988, 171-178.

Greenberg, S. and Cockburn, A (1999). Getting back to back: alternate behaviors for a web browser's back button. *Proceedings of the 5th Annual Human Factors and the Web Conference*, Held at NIST, Gaithersburg, Maryland, USA, June 3.

Guzdial, M., Kolodner, J., Hmelo, C., Narayanan, H., Carlso, D., Rappin, N., Hubscher, R., Turns, J., and Newstetter, W. (1996). The collaboratory notebook, *Communications of the ACM 39*, 4, 32-33.

Hightower, R. R., Ring, L., Helfman, J., Bederson, B. B., & Hollan, J. D. (1998). Graphical multiscale web histories: A study of PadPrints, *Proceedings of ACM Conference on Hypertext (Hypertext 98)*, ACM Press, 58-65.

Hill, W. and Hollan, J. (1993), History-enriched digital objects, *Proceedings of Computers, Freedom and Privacy (CFP'93)*, available from http://www.cpsr.org/dox/conferences/cfp93/hill-hollan.html

Jones, P. M. and Schneider, K. J. (1996). Learning environment for magnetic resonance spectroscopy (LEMRS): Supporting apprenticeship learning in operational environments, *Journal of Educational Multimedia and Hypermedia 5*, 2, 151-177.

Kurlander, D. and Feiner, S. (1992). A history-based macro by example system, *Proc. User Interface Software and Technology '92*, ACM, New York, 99-106.

Kurlander, D. and Feiner, S. (1993). A history of editable graphical histories, In Cypher, A. (Editor), *Watch What I Do: Programming by Demonstration*, MIT Press, Cambridge, MA, 405-412.

Levy, G. and Baecher, G.B. (1998). NileSim: Learning by Experimenting, J*ournal of Water Resource Systems and Management*, ASCE, in press.

Lu, G. B., Oveissi, M., Eckard, D., and Rubloff, G. (1996). Education in semiconductor manufacturing processes through physically-based dynamic simulation, *Proc. Frontiers in Education'96*, IEEE, 250-253.

Munro, A., Johnson, M.C., Pizzini, Q.A., Surmon, D.S., and Wogulis, J.L. (June 1996), A Tool for Building Simulation-Based Learning Environments, in Simulation-Based Learning Technology Workshop Proceedings, ITS'96, Montreal, Qubec, Canada.

Nahvi, M. (1997). Dynamics of student-computer interaction in a simulation environment: Reflections on curricular issues, *Proc. Frontiers in Education'97*, IEEE.

Plaisant, C., Tarnoff, P., Keswani, S., and Rose, A. (October 1998). Understanding transportation management systems performance with a simulation-based learning environment, University of Maryland Department of Computer Science Technical Report CS-TR-3947, UMIACS-TR-98-60, ISR-TR-98-59, College Park, MD.

Rose, A., Eckard, D., Rubloff, G.W. (May 1998). An application framework for creating simulation-based learning environments, University of Maryland Department of Computer Science Technical Report CS-TR-3907, UMIACS-TR-98-32, College Park, MD.

Shneiderman, B. (1998). *Designing the User Interface: Strategies for Effective Human-Computer Interaction: Third Edition*, Reading, MA: Addison-Wesley Publ. Co.

Tauscher, L. and Greenberg, S. (1997), How people revisit web pages: Empirical findings and implications for the design of history systems, *International Journal of Human-Computer Studies*, Academic Press.

Wexelblatt, A. and Maes, P. (1997), Visualizing histories for Web browsing, *RIAO'97: Computer-Assisted Information Retrieval on the Internet,* Montreal.

Wexelblatt, A. and Maes, P. (1999), Footprints: history-rich tools for information foraging, *CHI'99: Human Factors in Computing Systems,* Pittsburgh PA, 270-277.