Improving Design with Artifact History

Brent Neal Reeves
TwinBear Research
6138 Gale Drive
Boulder, CO, 80309
brentr@cs.colorado.edu

Abstract

History tools play an important part in supporting human computer interaction. Most research in history tools has focussed on user interaction histories. In contrast, this paper presents a theoretical framework for artifact history and describes a computer based design environment which implements embedded artifact history. The most promising area for history tools is in collaborative design, helping users to understand others' as well as one's own previous work.

Introduction

History tools serve several needs in human computer interaction. Lee [1992] argues that they can help alleviate problems in human computer interaction such as variability from user to user and session to session. User activities contain repetition, which motivates a way to provide support for easily reusing previous commands (GreenbergWitten, 1988). Studies of the use of history have focussed on history as a domain-independent tool for user support (Lee, 1992; Barnes, Bovey, 1986).

Though repetition plays a part in human-computer interaction, it is only a small part. History tools have far greater potential than command-line recall. There is much potential for computer systems to serve as external memory aids in restoring the context surrounding past design decisions (Suchman, 1987; Anderson, 1985; Reder, 1982). In the context of collaborative design, it is not enough to provide *user* history, there should be *artifact* history. Reeves [1994] evaluated these issues with respect to collaborative design; the focus in this paper is on the individual's perspective of artifact history.

After reviewing related work and arguing in detail for artifact history, this paper will describe INDY, a design environment for local-area network design. User studies raised several interesting issues related to embedded artifact history.

Motivation

Context and Reminding

Research in human memory has shown that people are prone to recall by inferring "what is plausible given what they can remember" (Anderson, 1985). Memory performance improves the more closely the current context matches the past physical, emotional and internal context. Much of recall involves plausible inference rather than exact recall (Reder, 1982).

Each designer on a project team understands only a portion of the overall design artifact. As large projects evolve over time it will become increasingly important for computer based design environments to help capture the evolution of an artifact and not just its current state. The history serves to remind designers of how the artifact came to be and what the context was when certain decisions were made.

The challenge for computer-based environments is to make that context available which is most useful to reminding. Since there is such a variety of aspects by which people index things, this is a difficult task. For example *circumstantial indexing* describes how people associate seemingly unrelated events in order to aid recall (Bolt, 1984). The promise of embedded history is to include the dimension of time in the indexing possibilities provided by a computer-based environment.

Design Rationale

Though design rationale appears to have great promise (Kunz, Rittel, 1970), there have been few recorded successes (Yakemovic, Conklin, 1990). The designers must perceive a benefit for the extra cost of documenting their reasoning (Reeves, Shipman, 1992). From the individual designer's perspective, nothing is welcome that adds to the responsibility without clear benefits, or with benefits only for others. The benefit of history related to design rationale is that in domains such as network design, which involve two-dimensional sketches and graphical representations, designers can often deduce rationale by seeing the process of how something came to be (ChenDietterichUllman, 1991). Artifact history is thus a way to capture some aspects of design rationale at no cost to the user.

In the network domain, a logical map of the current network hides many tradeoffs and compromises that were made in the past, yet which still affect current decisions. Having the history of the evolution clarifies some of the tacit knowledge that is represented in the static logical map. Also, artifact history allows the individual designer to benefit from others' work without requiring extra work on their part.

Related Work

Human Computer Interaction

Computer tools which incorporate history are based on the observation of how humans use the past to solve current problems (Lee, 1992). The common use of a history tool is to reuse and possibly modify a history item to save keystrokes and/or mouse strokes (Linxi, Habermann, 1988). History serves users by allowing reuse of previous interaction.

Lee [1992a] provides a summary of recent work in history tools and identifies seven uses of history:

1. reuse: reusing scripts directly or with modification
2. inter-referential I/O: referring to previously displayed information in current interaction
3. error recovery: undoing mistakes
4. navigation: finding out one's current place, and path taken
5. reminding: aiding recall and providing visual cues to past events
6. user modelling: responding to differences in individual users
7. user interface adaptation

Studies of the use of history have focussed on history as a domain- and application-system independent tool for user support (Lee, 1992; Barnes, Bovey, 1986). User activities contain repetition, which motivates a way to provide support for easily reusing previous commands (GreenbergWitten, 1988). The area most researched and most supported by computer-systems is *reuse*. According to Linxi and Habermann [1986], the most common use of a history tool is to reuse and possibly modify a history item to save keystrokes and/or mouse strokes. Though undoubtedly useful in simple tasks such as UNIX command line submission, focusing on reuse of an operating system command overlooks how people use computer-based environments to carry out complex design projects.

Though these interactions with an operating system are important and tools such as TCSH (Ellis, Greer, Placeway, Zachariassen, 1987) are a useful addition to operating system interface, the outcome of a multi-year design project hardly depends on whether the designer types *lpr project.ps* or reuses a previous *lpr* command. Whether it is command-line recall, or macro recording, the support is still barely above the keystroke level. Few would argue the usefulness of macros, but higher levels of support are needed for collaborative design.

The taxonomy above assumes that history is user-specific. However in collaborative design, an additional perspective is helpful, that of artifact-specific history. The focus shifts from the individual to the shared artifact.

Embedding History in Artifacts

Researchers have looked at the physics of real-world objects to inform the design of computational media (Hill et al., 1992). For example, as auto parts manuals become worn, they provide visual and tactile cues to guide further use. In the same way that physical wear and tear can be a resource, computational media should embed the history of an artifact in the artifact so that it can serve as guide to further use. This history should be embedded in the design artifact in such a way that the system can leverage the contents of both the artifact and the history. Cross reference from history event to location of the object in the artifact (and back) should be easy.

One example of this in INDY (described in more detail in the next section), was that deleted items would not disappear entirely, but would retain a grayed-out or ghosted image on the screen. It was visible enough to notice (and thus to remind the user that there had been something there), but not so prominent as to be a major distraction.

Though *user* interaction histories have proven useful (Lee, 1992), a system that supports design should take the *artifact* perspective. This means that the history is in terms of all changes made to the artifact, rather than several histories stored by user. This provides continuity of design projects as people come and go and the project evolves. The shared artifacts as well as the interaction is kept together to support collaborative design.

Suchman [1987] argues that human computer interaction is constrained by the tendency of the human's knowledge to be context-specific or *situated*. In long-term design projects designers must collaborate across space and time. This means that knowledgeable domain workers will need tools to help remember the relevant issues and see the progress of the various projects.

In a computer based design environment, history tools can provide the context of previous work, rather than just its current state. Understanding "how" something came to be often answers many "why" questions, especially in design tasks that involve graphics, such as CAD (Sukaviriya, 1990). Local and Wide Area Network Design is an area in which logical and physical diagrams play a central role in communicating crucial information. Static diagrams hide much information relevant to modifying the current network, such as how certain tradeoffs and compromises came to be.

Summary

Related work on history tools thus illustrates three goals. First, the level of support has been too low. Users need more than command-line recall. Second, the history should not be separated from the artifact, but embedded. There are aspects of physical wear that should find representations in computer based media. Third, the history should be about an artifact, not just an individual user. Collaborative design relies heavily on the use of shared artifacts and the design history of an artifact goes beyond any one user's interaction history.

INDY: History Support for Collaboration

INDY is a system which supports the collaborative design of computer networks. It builds on design environments (Fischer, 1992) by providing a design medium in which the design as well as the communication about the design is stored. This section describes the domain of computer network design and summarizes a long-term study of users of INDY.

Network Design Domain

A local area network links devices such as workstations, file servers, and printers using cables of different lengths and types. Networks can be viewed at different levels of abstraction, such as connectors and cables, work groups, and entire subnets. Beyond the subnets are building-wide backbones and eventually extremely large nets such as arpa- and internet.

Rarely are networks designed from the ground up. There is usually too much investment in the old infrastructure and adding functionality is a process of evaluating difficult tradeoffs. It seems to be worse than software in that most of the work is done after the initial design is allegedly over. Just like in a large software project it is crucial to have a chief architect who oversees the project from start to finish, so too does a turnover in systems administration make it difficult to make good decisions about tradeoffs. There is rarely one "right" way to build a complex network at the level of, say, the Engineering Center at the University of Colorado at Boulder (CU). Thus personal experience and bias plays a large role in redesigning existing networks.

This domain is useful to study from the perspective of this research, because the context of a design decision is so important in a rapidly changing domain. There are many ways to support given requirements and problem framing plays an important part. Network design is typical of a wide range of other design domains. Though analysis tools exist, it is not possible to automate network design. Each solution illustrates tradeoffs between many goals such as cost and reliability.

Figure 1 shows how to trace the history of any design unit. A list of changes is shown, and the corresponding history transactions are highlighted in the history pane (in this case the complete artifact history is long and only one history entry of the specified design unit (Workstation-259) is visible in the right pane.) Support is also provided for finding recent work (Figure 1 shows that user "haleden" made changes on 1/28/93).

One can select any of the history transactions and the artifact will be "rolled back" to that point in time. That is, the diagram will look like it did in the past; additions will be removed and items that have been deleted will appear once again. The artifact is rolled back one change at a time, so that each move, create, or delete is seen. In some cases a unit is affected which is not currently on the visible page, in this case, the window is automatically scrolled to show the affected unit.

Network designers made much use of this feature as they each have primary responsibility for only portions of the network, yet their work often overlaps. They can restore the artifact to the state in which they last edited the artifact and are able to better understand the rationale and implication of any new modifications. As a reflexive tool, the previous context is restored to enhance recognition of one's own work. The tool thus acknowledges and supports designers' situated cognition.

Users were involved with INDY in two ways. First, they helped in the design process by evaluating early prototypes. Second, they participated in a long term study of collaborative design. This study involved two designers taking turns designing and evaluating each other's work, followed by others analyzing the resulting design. It was conducted in a participatory design fashion and user suggestions guided the implementation. The purpose of the long term study was to simulate the process of several project members collaborating asynchronously and to evaluate the role played by embedded communication and embedded history.

All participants in this study were experienced in network design, and all five were employed in various roles of network design at the time of the study. Table 1 shows that experience ranged from 4 to 12 years. This included experience in systems administration and local and wide-area network design.

Setup and Tasks

The study took place at of the University of Colorado at Boulder. After a representation of the current computer network was made in INDY, two participants were given the tasks of adding workstations to this network and evaluating each other's work. This collaboration took place over a period of 5 months. User names have been changed to preserve anonymity. The following sequence was chosen to closely mirror current events and provide opportunities for studying collaborative design:

This author entered parts of the office tower (OT) and classroom wing (CR) backbones and the labs which are attached into INDY. To give a flavor the "size" of the task, the history list shows that 56 transactions were used to setup the initial network configuration. That is, 56 design changes such as create, move, connect were used to seed the environment in which the users tasks were then carried out. This author entered two main subnets in the engineering center at CU. Then Joshua and Luke took turns, each a couple of weeks apart, in making modifications to the net. At the end several other network designers evaluated the artifact.

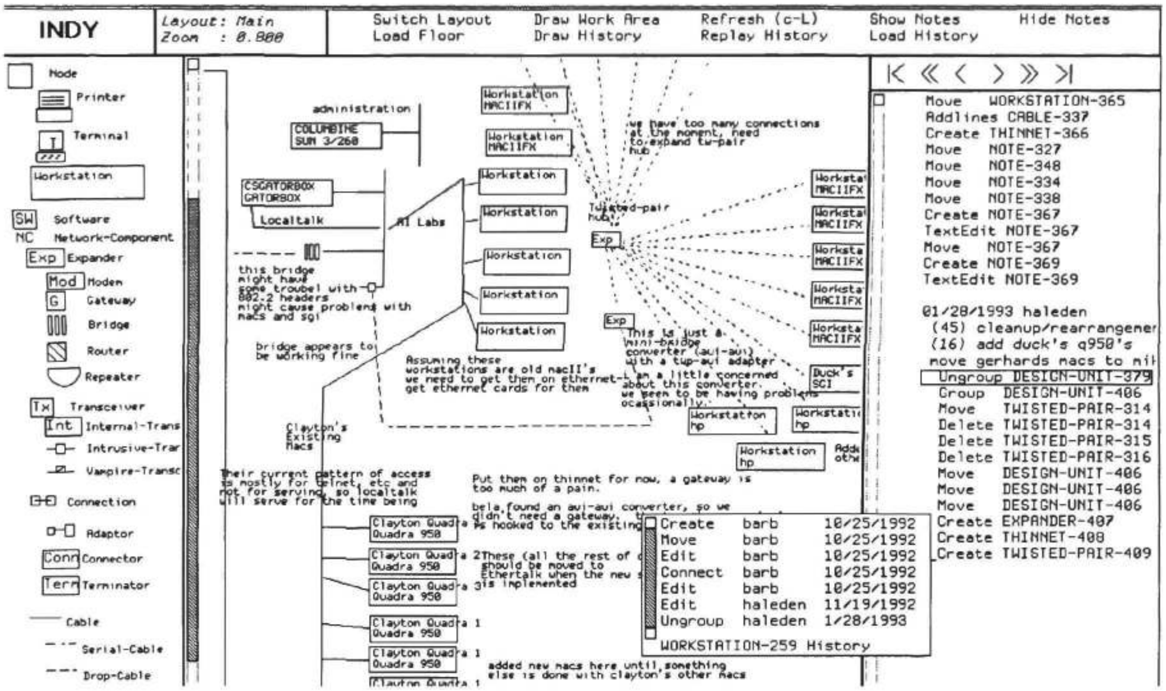

Figure 1: INDY: a network design tool with embedded history

The major panes in this screen image (from left to right) are the Palette, the Work Space, and the History. The history has a cassette-deck (upper right) as well as direct-manipulation interface. The history can be queried according to users, design units, and dates. The pop-up window (lower right) shows the trace of a workstation, which was created on 10/25/92.

Table 1: Users' experience and background

User	Network Design Experience
Joshua	7 years
Luke	10 years
Ben	12 years
Micah	4 years
Pat	6 years

Tasks were chosen to coincide with recent and proposed machine purchases and changes in the Computer Science department. The tasks were close enough to the real situation to cause designers to take into account the real situation in carrying out the tasks. Five of the sessions were videotaped and protocol analyses showed an intermixing of references to the INDY representation and the CU network in resolving design decisions and evaluating the network layouts.

Results and Discussion

After the initial creation of the artifact with 56 transactions, participants used another 249 transactions to accomplish the tasks during the study.

Artifact changes. Designers did not hesitate to make changes to the artifact. This was due in part to the explicit directions to "make any changes you want to." However there was no hesitation, or indication of discomfort in carrying out that assignment. It was not possible to tell how the absence of a history mechanism would have affected this, i.e. whether the willingness to change other's work was due to the fact that the changes could easily be undone, or whether the willingness to change something was related to the individual's confidence that the changes were indeed *correct*.

Perception of history. In their use of the history mechanisms, it seems that designers were not thinking of it in the sense of Hollan and Stornetta [1992], i.e. archived *communication*, but rather as an archived *trace of changes*. Nevertheless one designer mentioned the time dimension of communication:

> I can see you wanting to have the notes basically, well the notes are time based, you have a time dimension. And so you might have conflicting notes attached to the same place or same object... And so you might want to see the entire sequence of notes associated with something. You might want to see the last note associated with it.

Reminding. Several suggestions were made relative to history access and presentation. Users requested access to the history *by user*. This arose in a reflexive CSCW use, namely the need to find where one had done work oneself.

History list organization. Designers wanted to group sets of transactions and describe them so that they did not have to look at a long list of changes as remember what they accomplished. To provide a feel for the "size" of a group, the number of transactions is shown. For example, the right pane of Figure 1 shows that it took 16 transactions to "add duck's q950's" (Apple Quadra 950s).

Embedded history. The integration of history and artifact was well exploited. The following quotes from the transcript motivate INDY's history access mechanisms:

> I don't remember exactly where I was last time. Is there a way to determine where I left off?
>
> Sometimes it'd be useful to know who made this change in the design.
>
> Another thing that might be useful would be for me to have some way to say "Where was that created and who created it?"
>
> So is there any way to find out who made these notes?
>
> Is there a way to find the history of an item?
>
> But that's incorrect... you see here, history's useful, because I want to go back to the context of when... I started more or less...

Language of design. A history mechanism also surfaces the issue of the "language of design", because the history must serve the user in understanding previous work (Schoen, 1987). The language used to describe actions should match closely the language used to do design. If one were to search the history for "change" one would not see "delete" followed by "create." The more closely aligned the history transaction language is with the "language of doing", the better the system will be at providing services to augment design.

Conclusion

History tools at the level of command line recall are certainly useful, but more research is needed at higher levels of problem solving. The research direction taken here indirectly motivates strong emphasis on task analysis, because the history is best represented in terms of tasks the user wants to accomplish (and later remember).

Situated cognition research on the importance of context motivates history tools that support the collaborative design of complex artifacts (Lave, 1988). They must be task specific and integrated with the design environment. The payoff is that the system can then augment human designers by providing important aspects of the context of previous work to stimulate recognition.

Individuals benefit in two ways from artifact history. First, it acts as a tool to aid in reminding. One can restore the context in which previous decisions were made and so better understand some of the tacit knowledge that affected the decisions. Second, it allows a user to learn from others' interaction without extra work on the others' part.

This research has also exposed limitations of history tools. This approach works best in domains where graphics play a central role. In these domains it is easy to represent as well as visualize changes. These changes carry much semantic meaning to experts so little else must be tracked except the graphical changes. In other domains it is not clear how to represent or visualize a change. For example software engineering has long had history tools (version control systems), yet there are no good tools for representing or visualizing changes.

Acknowledgements

The members of the HCC group at the University of Colorado helped in the development of the conceptual framework that this work is based on. The members of the Computer Operations Committee (especially Barb Dyker and Hal Eden) participated in the design and evaluation of INDY. This research was supported in part by grant No. IRI-9015441 from the National Science Foundation and a fellowship from NYNEX.

References

J.R. Anderson (1985). *Cognitive Psychology and Its Implications (2nd Edition).* New York: W.H. Freeman and Co.

D.J. Barnes, J.D. Bovey (1986). Managing command submission in a multiple-window environment. *Software Engineering Journal, 1*(5), 177-184.

R.A. Bolt (1984). *The Human Interface.* Belmont, CA: Lifetime Learning Publications.

A. Chen, T.G. Dietterick, D.G. Ullman (1991). A Computer-Based Design History Tool. *Proceedings of the 1991 NSF Design and Manufactering Systems Conference*, 985-994. Austin, Texas: SME (Society of Manufactering Engineers).

M. Ellis, K. Greer, P. Placeway, R. Zachariassen (1987). *TCSH - C shell with filename completion and command line editing.* Unpublished. (UNIX Programmer's Manual (revised U of T, previous revisions from Fairchild, HP Labs., and OSU IRCC)).

G. Fischer (1992). Domain-Oriented Design Environments. *Proceedings of the 7th Annual Knowledge-Based Software Engineering (KBSE-92) Conference (McLean, VA)*, 204-213. Los Alamitos, CA: IEEE Computer Society Press.

S. Greenberg, I.H. Witten (1988). How users repeat their actions on computers: Principles for design of history mechanisms. *Proceedings of ACM CHI'88 Conference on Human Factors in Computing Systems*, 171-178. New York: ACM.

W.C. Hill, J.D. Hollan, D. Wroblewski, T. McCandless (1992). Edit Wear and Read Wear. *Human Factors in Computing Systems, CHI'92 Conference Proceedings (Monterrey, CA)*, 3-9. ACM.

W. Kunz, H.W.J. Rittel (1970). *Issues as Elements of Information Systems* (Working Paper 131). Berkeley, CA: Center for Planning and Development Research, University of California.

J. Lave (1988). *Cognition in Practice.* Cambridge, UK: Cambridge University Press.

A. Lee (1992). *Investigations into History Tools for user Support.* Unpublished doctoral dissertation, Toronto, Canada: University of Toronto.

C. Linxi, A.N. Habermann (1988). *A history mechanism and undo/redo/reuse support in ALOE* (Technical Report CMU-CS-86-148). Pittsburgh, PA,: Department of Computer Science: CMU.

L.M. Reder (1982). Plausability judgment versus fact retrieval: Alternative strategies for sentence verification. *Psychological Review, 89*, 250-280.

B.N. Reeves (1994). The Role of Artifact History in Collaborative Design. *Submitted to the Conference on Computer-Supported Cooperative Work (CSCW'94)*, .

B.N. Reeves, F. Shipman (1992). Making it Easy for Designers to Provide Design Rationale. *Working Notes of the AAAI 1992 Workshop on Design Rationale Capture and Use*, 227-233. San Jose, CA: AAAI.

D.A. Schoen (1987). *Educating the Reflective Practitioner*. San Francisco, CA: Jossey-Bass Publishers.

L.A. Suchman (1987). *Plans and Situated Actions*. Cambridge, UK: Cambridge University Press.

Sukaviriya (1990). Coupling A UI Framework with Automatic Generation of Context-Sensitive Animated Help. *Third Annual Symposium on User Interface Software and Technology*, 152-166. New York: ACM Press.

K.C. Yakemovic, E.J. Conklin (1990). Report of a Development Project Use of an Issue-Based Information System. *Proceedings of the Conference on Computer-Supported Cooperative Work (CSCW'90)*, 105-118.