User Activity Histories

Etienne Pelaprat
etienne@hci.ucsd.edu

R. Benjamin Shapiro
bshapiro@hci.ucsd.edu

Cognitive Science, University of California, San Diego

Abstract

Current software interfaces fail to incorporate historical data from user interaction into their design. While some systems exhibit a minimalist use of history in the form of undo and redo, selective menu items, and other static elements, there has been a lack of use of history in the dynamic elements of interaction. We propose a more widespread use of historical data from user-software interaction to augment the desktop and application environment. We believe the use of historical data can improve the user's experience at many different levels. Our approach begins by assuming that everything the user is doing on the desktop is important to them, and that it will be important again in the future.

Keywords: Groupware, Human Factors, Workflow Management, Graphical User Interfaces

1 Introduction

In the real world there exists a rich and visual history of interaction between people and physical objects. Books contain annotations in their margins, clothes become faded, door knobs lose their original color and texture – these are all signs of use that we perceive as evidence of interaction. There is a "history of use" for all physical objects in the world, and we make this history useful for us. The history of an object can develop incidentally and deliberately. An annotated page will help us re-capture a train of thought, a faded shirt is a favorite shirt, and the discolored and worn part of a door knob indicates where it is best to push or grasp to open the door. Our research applies the "history of use" metaphor to digital objects. An interaction with a digital object is not an ephemeral conversation, but rather a process that changes both the user and the digital object, and we believe it is important that the digital object reflect that change or take advantage of that interaction. Furthermore, we see great promise in applying historical data of interaction to dynamic elements of the desktop environment.

2 Interaction History

Interaction history research has been approached in many different ways. Kurlander and Feiner were some of the first to extend undo and redo from what they called *spatial browsing* and *temporal browsing* to a graphical display [3]. This approach was restricted, however, to a particular application which relied heavily on a single graphical interface. Nevertheless, they were able to show that their graphical augmentation of classical undo and redo increased user efficiency. Will Hill and Jim Hollan extended the history of use metaphor to digital objects on the computer [1]. They applied the metaphor to a modified version of the text-based graphical editor Emacs and developed Read Wear and Edit Wear [2]. Read Wear and Edit Wear provided a visual representation of history of use for files edited in Emacs. While this approach was still limited to a particular application, the history described here was more permanent than simple undo and redo. Users could leave the application, even quit it, and return to it days later having the history of previous use still be there. Read Wear and Edit Wear was a particularly interesting implementation of history because it included in its representation who had modified or read a file. The representation of wear was distributed across a system of people, rather than an individual. This is an important idea for the use of history which we will investigate and develop later in our research.

There has been research in building and representing histories of user interaction with the web [4]. Alex Wexelblat's research was primarily interested in how to best represent and share web trails, or routes of travel through the web, rather than the interaction with the web pages themselves. The interaction was limited to merely viewing the web page. This approach to history research is important, however, because it remained independent of a particular application setting. The web provides a setting which is very diverse – a rich and unpredictable mix of graphics and text. Wexelblat's approach was removed from a particular setting and instead took a more top-down view of interaction history.

2.1 Focus of History

Past approaches to recording history have had to compromise between focused, accurate history of interaction in a particular setting and minimally descriptive, sparsely detailed history of interaction with the entire system. Read Wear and Edit Wear, as well as graphical undos, was limited to one application setting. However, they provided a contextually sound history of interaction within that setting. Conversely, Wexelblat's work allowed for a very comprehensive description of the overall interaction, but sacrificed the content of the interaction of the user with the interface. Consequently, it has been difficult to research the impact activity history could have on an entire system like the desktop environment. The challenge we are faced with is how to develop a method of gathering historical data that is rich in details as well as comprehensive to the entire system, where there is no compromise. Our approach to gathering history works from a bottom up as well as top down view.

Our bottom up approach begins where the interaction begins on computers: at the system level. It is possible, on linux systems, to run the system utility "strace" (*s-trace*) such that it intercepts system function calls and their arguments when running X Windows, a graphical windowing system.. One of the benefits of using strace to gather history data is that it runs seamlessly in the background. A user can record their history with no added cost. Furthermore, strace will also trace any children spawn from a traced application. With strace, we are able to create a rich record of history about all the work done while in X Windows. Unfortunately, the raw data is cryptic and not immediately useful when trying to get a sense of the interactions of the user and the desktop.

The goal of the top down approach is to support the data we gather from the bottom up approach. To do this, we have modified the graphical user interface (GUI) libraries of a major X Windows Distribution. This affords us information we could not directly extract from the strace data. For example, active and passive windows, scrolling, menu items and their actions, and more. This extra information is set in a specific context, but will help us build a more accurate history of interaction.

2.2 Inferring Structure and History

Certainly, every example stated above is inferring interaction history from the data they collect. It is also the case that those systems engaged in relatively shallow inferences, and did not stray far from the raw data they collected. Our approach performs a set of inferences in cycles. Each makes inferences about a higher level of structure of history from the inferences formed by the previous cycle.

A first cycle analyses the raw time stamped *strace* data and extracts several bits of information. We use this data about keyboard and mouse input to determine what application the user was interacting with when. Any type of file operation can be observed and tied to the application performing it. A second cycle uses this data to make inferences about semantically meaningful requests, such as a ”save file” command. After individual application level data is collected and processed, we aggregate input across multiple applications to find higher-level actions that span multiple applications. Finally, we integrate these actions and try to detect abstract task-level activities, like writing a research paper while finding information on the web. Once these tasks are known, we might mine a users task history to find common patterns, like writing code in the morning and debugging in the afternoon.

Our approach to collecting and representing user interaction history allows us to look at the entire working system and the types of structures the user has created. Furthermore, because we store low and high level data, any part of a representation presented to the user can be decomposed into lower-order parts, providing high, task-level representation of behavior, and all of the rich details that constitute it.

3 Applications of History

We believe a description of structure and history described at many different levels can have interesting and useful applications in the desktop setting. Just as objects in the real world benefit from a history of interaction with the user, so too can the digital world.

3.1 Digital Diaries

We believe historical data can be accumulated over time and used as a diary of activity on the computer. Furthermore, it will be interesting to develop different kinds of representations of the data, given the application. For example, we envision a diary system whereby a user can look back in time and get a sense of their progress on a certain project. They can also annotate their past work and share it with other folks. We are unsure about what kinds of interactions users would have with diaries of their work because we have not tested them on real users, but anticipate many positive and useful interactions.

The nature of the diary will also be driven by how it represents the data. We feel finding the best representations will be mediated by the nature of the diary. A work diary may focus on representing work patterns, such as files used by a program, when they were used and how. Conversely, a personal diary may try to represent your leisure time. For example, if you're writing e-mails to friends or chatting with them online, the representation my reflect how much you are communicating with people, or perhaps even the nature of the communication. Both diaries, work and personal, will represent history from the same source differently, and it will be interesting to see how those develop.

3.2 Routine Driven Interfaces

We are developing speculative, routine based interfaces. These are interfaces that take advantage of historical data when constructing menus of high-order, task-level options. Users' sequences of input actions are very seldom random. Rather, they correspond to well linked chains of activity that represent organized scripts of behavior.

This scripting occurs at several levels. Users follow scripts within applications, for example always opening an ”address book” after starting a new mail message. Scripts may be data-driven: a user might always open the same spreadsheet when writing to a particular contact. And scripts may be temporally bound and attached to regular, daily activities. A user might have a consistent morning routine that consists of logging-in to the computer, immediately checking email, viewing several favorite web sites, and then diving into a word processor (a reasonable routine for a secretary). In addition to this sequence, the user might have a habit of always checking mail at noon, just before leaving for lunch.

Most modern GUIs have some sort of task launch menu always easily accessible to the user (e.g. MS Windows' *Start Menu* and KDE's *KMenu* application launcher). These task launchers try to provide easy access to all applications available on the system. However, none of these systems has provided a means of filtering the set of options presented to match the user's activity patterns. At best, they construct a small subset of the most recently used applications and display those choices on the first menu presented to the user. We try to extend this filtering to take advantage of knowledge of script-level potential activities. We will construct menus populated with choices that are most likely to be helpful in the context where they are presented (and, of course, allow the user to easily revert to a broader range of options).

4 Conclusion

There exists a need for historical data on the desktop. Current software interfaces lack the history of interaction which is so prominent in the real physical world. We propose to elaborate on past work in this area of research not by simply imitating the real world on the digital desktop, but by also taking it to the next level. The digital world affords us many benefits not present in the real world which can augment the use of history. We have described two examples, digital diaries and routine driven interfaces, which take advantage of the rich interaction between the user and the desktop. We have also shown that the use of history in these examples is augmented by its digital nature.

References

[1] HILL, W., AND HOLLAN, J. History-enriched digital objects. In *Proceedings of Computers, Freedom, and Privacy* (1993).

[2] HILL, W., HOLLAN, J., WROBLEWSKI, AND MCCANDLES. Edit wear and read wear. In *CHI'92 Conference on Human Factors in Computing Systems* (1993).

[3] KURLANDER, D., AND FEINER, S. Editable graphical histories. In *IEEE Workshop on Visual Languages* (1988).

[4] WEXELBLAT, A. History-based tools for navigation. In *Proceedings of Hawai'i International Conference On System Sciences* (1999).