Managing project contexts: Interaction history as a resource

Victor Kaptelinin
Department of Informatics
Umeå University
901 87 Umeå, Sweden
+46 90 786 5927
vklinin@informatik.umu.se

ABSTRACT

The paper deals with exploiting the potential of interaction histories for managing multiple project contexts in both traditional and smart environments. Mapping interaction histories to specific projects is proposed as a way to make interaction histories a useful resource for supporting continuous, coordinated work on a set of projects over time and distributing resources across contexts and devices. The proposed approach is illustrated with a simple example of using project-specific interaction histories for synchronizing work between a personal computer and a mobile device. Implications of the proposed approach to design of smart environments are discussed.

Keywords

Interaction history, project context, distributed work

INTRODUCTION

Even the modestly smart environments of today, featuring, for instance, automatic doors or sensor-based lighting, may cause problems for people in the environments by imposing excessive constraints, creating uncertainty, and misinterpreting user intentions. If car doors unlock automatically when the owner is approaching, how can one check if the locks work properly?

When environments become more "intelligent," the risks of causing mismatches between user's and system's models of interaction are likely to increase. Development of new interaction techniques capable of minimizing such risks is considered a key issue in design of smart environments [1].

This paper argues that exploiting interaction histories in smart environments can be facilitated by allowing the users themselves indicate (implicitly or explicitly) what their goals are. More specifically, it is suggested that providing support for selecting the currently active project – a relatively long-term sequence of tasks, subordinated to a higher-level goal, distributed in time and place, and often interrupted – can help utilize information contained in interaction histories and provide support to people acting in smart environments. The analysis in the paper is based on experience of employing interaction histories in a traditional desktop environment. Capitalizing on this experience, the paper makes an attempt to address issues related to smart environments.

The rest of the paper is organized as follows. The next section identifies the need to cope with the enormous volume of data that can potentially be included in interaction histories, in both traditional and smart environments. After that a number of possible ways to make interaction history data more manageable are discussed, including mapping events in interaction history to user's projects. Then a simple example of utilizing project-related interaction histories to support work distributed between several computing devices is presented. The paper concludes with a reflection on the implications of the proposed approach for creating smart environments.

INTERACTION HISTORIES: LIMITED YET ABUNDANT

Preserving and examining the traces human activities leave in the physical world may require a considerable effort. By contrast, traces left in virtual environments allow for relatively effortless storage and analysis. Given enough memory space and processing power, information technologies can record user inputs (or other external inputs), system events, and store them in the form of automatically created interaction histories. The potential of interaction histories for supporting the user was recognized by researchers and practitioners quite early. In the field of Human-Computer Interaction (HCI) interaction histories have been an important research issue for over a decade. For instance, a panel organized at the CHI'94 Conference [10] identified main functions of interaction histories in interactive systems and formulated an agenda for future studies in that area. In software development interaction histories have been practically employed, in one way or another, in a wide range of computer applications and systems [14].

Arguably, however, both research and practical applications of interaction histories are still in their infancy. For the most part, researchers and practitioners focussed so far on relatively simple and obvious uses of interaction histories. The list of issues waiting to be properly addressed, indicated in this workshop's call for papers [13], testifies that interaction histories remain a largely untapped resource in HCI.

There are at least two reasons why interaction histories have been difficult to study and use in traditional HCI. First, the possibilities for collecting informative interacting histories are rather limited. Recording low-level events, such as keystrokes or mouse clicks, is a relatively simple task but inferring user actions, -- and objects employed in the actions, -- from the low-level evens is often problematic. Some programs, such as Microsoft Office ® applications, generate higher-level events and thus support collection of informative interaction histories. However, many programs do not provide such support. In addition, very few systems automatically capture user actions in the physical world, such as, talking to a colleague during lunch or placing a carbon copy of a document in a physical folder. Therefore, an important part of users' everyday activities is not represented in interaction histories.

Second, even though interaction histories are limited, they can be excessively large. According to our experience [8, 9], recording interaction histories generates volumes of data, which makes it impossible for users to keep track of unprocessed histories. To make use of interaction histories users have to rely on representations produced by the system. Currently, little is known about how to present interaction histories to the user so that they are helpful rather than confusing.

Therefore, interaction histories are at the same time limited and abundant. Moving from traditional computer use to smart environments alleviates the first of these problems. Sensor technologies open up radically new possibilities for capturing human interaction with the world.

However, the second problem – abundance – is likely to get worse. The sheer amount of data generated by smart environments can be overwhelming. Even the most advanced storage devices can be insufficient for storing all that data. Therefore, the question of how much the system should remember remains open [15].

The volume and diversity of data in smart environments also present a problem for analysis of the data. The fact that the data is analyzed automatically does not by itself eliminate the problem. If people who create or otherwise control technology have a vague or unrealistic idea of how interaction history data can support interaction in principle, no processing power can rectify that.

BREAKING DOWN THE FLOW

There are two main ways to reduce the complexity of a recorded interaction history and make it more manageable. The first way is to summarize the information contained in the history, for instance, with tables, charts, or timelines, displaying the frequency, aggregate time spent, or distribution of certain types of actions or certain objects. Such representations of an interaction history in general could be useful, for instance, for reflection or accounting. Summarized representations can also be used automatically. For instance, if it is established that at a particular time people form lines at a certain ATM, at that time a sign could display information pointing out to other available ATMs in the area, while at other times the same sign can display different information.

The second way to make information contained in an interaction history more practically useful is to process the information and transform it into a form relevant to the task at hand (cf. [8]. Analysis of literature reveals several strategies employed to relate interaction histories to user tasks: (a) identifying patterns of co-occurring objects, (b) selecting a sub-set of history on the basis of formal criteria, (b) mapping to objects, and (d) mapping to projects.

Identifying patterns of co-occurring objects includes selecting an object, such as an email address [5.7] and detecting other objects that appeared in an interaction history concurrently with the selected one. The structure of the associations created, for instance, by applying cluster analysis techniques, can be visualised as a configuration of nodes linked to the selected object and to each other. This type of analysis opens up a possibility for a user to find objects relevant to the task at hand by following their links to other objects. The user can start with an available object, browse through its links (if necessary, selecting an associated object and exploring, in turn, its links, etc.) and eventually find relevant resources.

Selecting a part of interaction history on the basis of formal criteria is similar to using the "Find" function: the user can select a time period, type and name of objects, and so forth, to create a smaller-scale, more manageable subset of an interaction history. For instance, the user can single out events that took place last week, which involved using documents with "ECHISE" in their names. An example of selecting a subset of interaction history is creating a sub-stream in the Lifestreams system [6].

Mapping to objects is linking events in interaction histories to specific objects. It allows the user to see the history of actions with an object by simply selecting the object. This approach was employed, for instance, in design of educational technologies [14]. The history of actions carried out by a student with an object in a simulation environment can be viewed by other students, and thus support communication, reflection, and mutual learning, Mapping to projects is linking events in interaction history to user's projects. We define projects as *higher-level, longer-term tasks*. Mapping events to projects allows the user to filter out irrelevant parts of interaction history and focuses only on relevant events when working on a project.

The rationale behind this approach is supporting users in managing projects. Since projects are carried out to attain higher-level goals, they are relatively independent from concrete information technologies. For instance, one can invite guests to a party via email, IM, SMS, phone, postcards, or face-to-face communication.

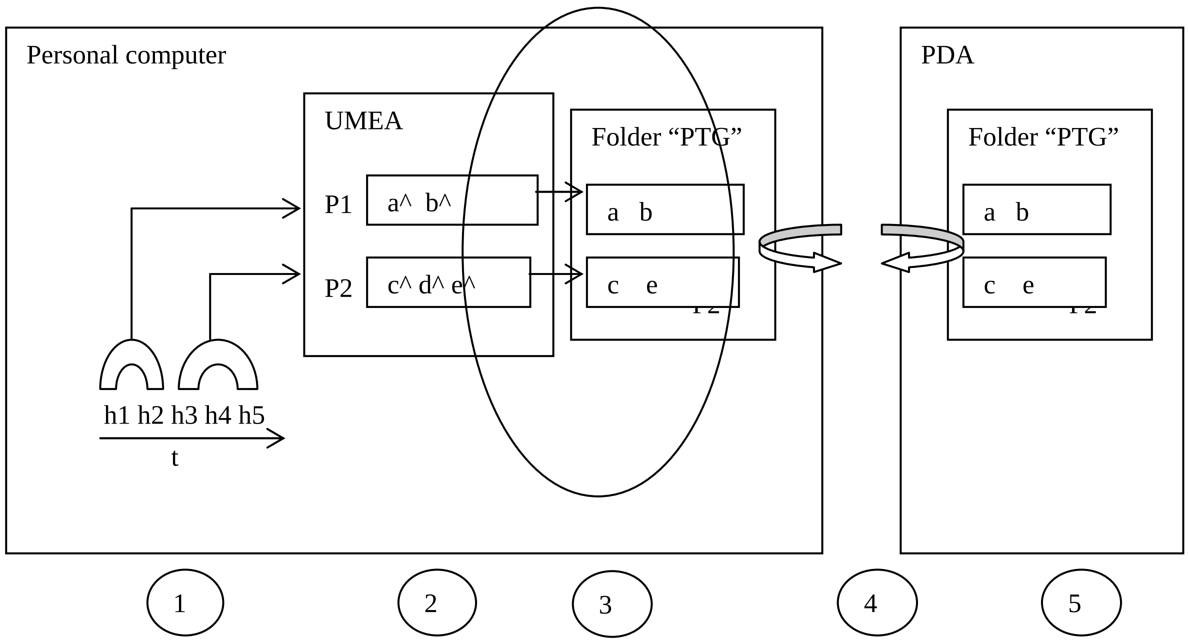

Fig. 1. "Packing for a trip": Copying files to PDA

Since projects are longer-term tasks, they are typically carried out in several sessions, distributed over time and intertwined with periods of work on other tasks. Therefore, working on a project requires: (a) ongoing coordination, making decisions about when to work on what project, (b) maintaining the continuity of working on a project despite pauses and breaks, and (c) integrating activities performed with various tools within one project. Empirical studies of computer users indicate that these problems are real.

Detailed, micro-level studies of the everyday use of information technology [2, 4, 9] revealed that people are constantly switching between different tasks. According to Czerwinski et al [4] "returned-to tasks," that is, tasks that tend to be resumed after an interruption, have a special status in the structure of user work. The study "… demonstrated that returned-to projects were more complex, on average, than short-term activities. These key projects were significantly lengthier in duration, required significantly more documents, were interrupted more, and experienced more revisits by the user after interludes." [4, p. 179]. At the same time, it was found that "… the reinstatement of complex, long-term projects was poorly supported by current software systems." [4, p. 175].

Mapping interaction histories to projects opens up possibilities to "stitch" separate sessions of working on a project into a coherent sequence of actions leading to the overarching aim of the project.

EXAMPLE: MANAGING TECHNOLOGICAL DISTRIBUTION OF WORK

Mapping interaction histories to projects can help maintain project coherence not only over time but also across various computing devices used within a project. Let us consider a simple example illustrating this claim.

In previous papers we presented a system named UMEA (User-Monitoring Environment for Activities) [11]. The system allows the user to define a set of projects and select one of the projects as active. The system monitors user actions and resources used within the active project, and automatically compiles project-related lists of resources. Entries in the calendar, notes, and "to do" lists are automatically linked to the active project, too. Therefore, when the user returns to a project by selecting it as active, the user gets convenient access to resources necessary for working on a project. At the same time, the user makes it possible for the system to update project workspace. An empirical evaluation of the UMEA system demonstrated that it helped users in managing their projects. At the same time, the evaluation identified possibilities for further improvement.

As a result of the empirical evaluation, the UMEA system was re-designed. One of the features added to the new version of the system (not yet reported) was intended to support distribution of work between several devices, for instance, between a desktop computer and a mobile device, such as a laptop, PDA, or smartphone. Mobile devices allow working on some tasks, such as reading and editing documents, when a regular personal computer is not available. However, limited memory space available on PDAs may make it impossible for the user to store all resources the he or she might possibly need. To support users in dealing with these problems the following feature was added.

Files related to different projects are automatically placed in different sub-folders.

The "PTG" folder serves for synchronization between the personal computer and the PDA (4). When the personal computer is synchronized with the PDA, using a standard synchronization feature of existing PDAs, resources selected by the user and stored in the "PTG" folder are copied to PDA's memory (5). If the user continues working on a project and creates new files or new versions of old files, these resources will be copied to the personal computer during the next synchronization session, again, using the standard functionality of existing handheld devices.

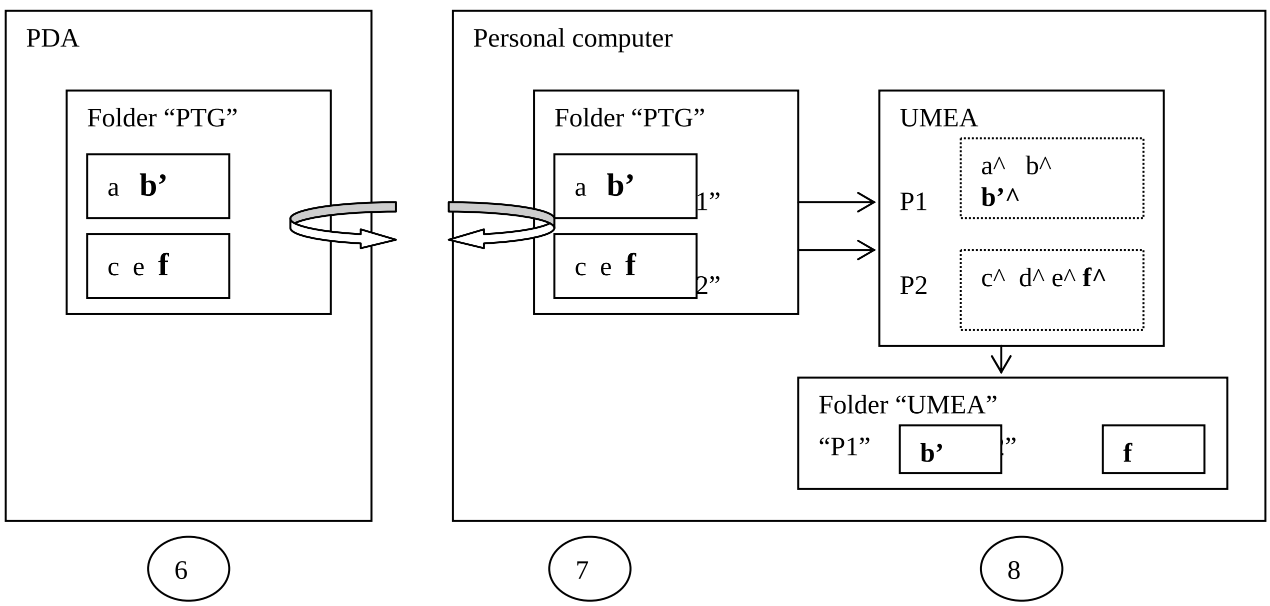

Fig. 2. "Unpacking": Adding new files from PDA to personal computer

The feature is schematically illustrated with Fig. 1. Via monitoring user actions (1) the system creates an interaction history, recorded as a sequence of events (h1-h5), where each event is an action carried out with a file (such as opening or saving). Events h1 and h2 are linked to project P1, while events h3-h5 are linked to project P2. By identifying files indicated in event descriptions the system links files **a** and **b** to project P1 and files **c, d**, and **e** to project P2 (2). The user can open the files from within the UMEA system by selecting a link to a file. The user does not need to know where a file is located. If the user wants to copy necessary files to a mobile device, he or she issues the "Project to go" command (3). The system displays a dialog window. The user browses through the files and indicates, which of them should be copied to the "PTG" folder (or any other folder selected by the user). Therefore, even though project-related files can be distributed all over the file system, the user can easily copy them to one folder.

In the next version of the UMEA system the "Project to go" feature is expected to be further advanced. Functioning of the prospective feature is shown in Fig. 2. New project-related files or new versions of existing files, created or copied by the user when working on the PDA (6) are added to a personal computer during next synchronization (7). These files are detected by the UMEA system and added to lists of files of their respective projects (8). In addition, the files are copied to appropriate project folders to make sure that they are not lost in the future.

The "Project to go" feature illustrates how the UMEA system uses interaction histories to distribute resources between a personal computer and a PDA.

To illustrate how a similar approach can be employed in a smart environment, let us consider the following imaginary scenario. When the user works on a project, the environment keeps track of using both virtual and physical resources. The system can display, for instance, the list of books and papers related to the project, their locations, when they were used last time, and so forth. When the user prepares for a meeting, lists of project-related resources can help decide which papers should be taken to the meeting and where to find them. When several projects are discussed during the meeting and new documents are distributed to the participants, the smart conference room keeps track of which documents are used within which project. This information is transferred to the user's personal work environment (for instance, the user can download it to his or her PDA) and when the user comes back from the meeting with a bunch of papers, these papers are automatically added as new resources to their respective projects.

CONCLUSIONS

Interaction histories remain to be a largely unexplored resource in human-computer interaction. This paper discusses one particular approach to using interaction histories, that is, mapping interaction histories to projects to support managing multiple project contexts.

More specifically, the paper draws on the experience of employing interaction histories in traditional desktop environments and argues that a promising way to design smart environments is:

(a) letting people choose what they want to do rather than inferring user intentions from available data, and

(b) making sure, in a non-obtrusive way, that relevant resources are "ready to hand" when the user needs them.

According to Streitz and Nixon, a key issue in designing smart environments is "When does the system (or the infrastructure) try to predict the user's intentions and when are the users presented with choices?" [15]. The analysis in this paper allows to formulate two tentative guidelines addressing this issue. First, an articulation of user's intentions should preferably be a "by product" of attaining a meaningful goal. For instance, a user of the UMEA system may make a project active just to get an access to project resources. A by-product of that is making it possible for the system to map user actions to the project.

Second, even in cases when users intentions are inferred, the user should be able to control the system. In the second version of the UMEA system users can *link* resources to projects. Selecting a linked resource automatically makes the corresponding project active. In this case user's intention to switch to another project is inferred by the system. But it is the user, who determines how the system works. For instance, the user can unlink the resource. An elegant combination of system inference and user control is described by Cypher [3]. His Eager system suggests the next action when it recognizes a repetitive activity. However, it does not constrain the user. The user can continue working as usual and when he or she feels confident that the intention is recognized correctly, the user can let the system finish the task.

Of course, further work is needed to establish how/if the approach and guidelines presented in this paper can be applied in design of smart environments.

REFERENCES

1. Coutaz, J., Crowly, J. L., Dobson, S., Garlan, D. Context is key. *Communications ACM*, 48 (3), 2005.
2. Cypher, A. The structure of users' activities. In: D. Norman, S. Draper. (eds.) *User Centered System Design*, *New Perspectives on Human-Computer Interaction*. Lawrence Erlbaum Associates, 1986.
3. Cypher, Allen. "Eager: Programming Repetitive Tasks by Example". In Proceedings of CHI, 1991 (New Orleans, April 28 - May 2). ACM, New York, 1991.
4. Czerwinski, M., Horvitz, E., Wilhite, S. A diary study of task switching and interruptions. *Proceedings of the CHI 2004 Conference on Human Factors in Computing Systems* (Vienna, Austria, April 2004).
5. Farnham, S. Personal Map: Automatically modelling the user's online social network. *Position paper at the CSCW 2002 workshop "Redesigning email for the 21st Century"* (New Orleans, Louisiana, November 2002), 2002.
6. Fertig, S., Freeman, E., Gelernter, D. "Finding and Reminding" Reconsidered. *SIGCHI Bulletin*, v. 28, January 1996.
7. Fisher, D., Dourish, P. Social and temporal structures in everyday collaboration. *Proceedings of the CHI 2004 2004 Conference on Human Factors in Computing Systems* (Vienna, Austria, April 2004). ACM Press, 2004.
8. Fischer, G. Articulating the task at hand and making information relevant to it. Human-Computer Interaction, 16(2-4), 2001.
9. Gonzales, V., Mark, G. "Constant, constant, multi-tasking craziness": Managing multiple working spheres. *Proceedings of the CHI 2004 Conference on Human Factors in Computing Systems* (Vienna, Austria, April 2004). ACM Press, 2004.
10. Hill, W., Terveen, L. New uses and abuses of interaction history: Help form the research agenda. *Proceedings of the CHI'94 Conference on Human Factors in Computing Systems*, 1994.
11. Kaptelinin, V. UMEA: Translating interaction histories into project contexts. *Proceedings of the CHI 2003 Conference on Human Factors in Computing Systems* (Ft. Lauderdale, Florida, April 2003). ACM Press, 2003.

12. Nardi, B. (ed.) *Context and consciousness: Activity theory and human-computer interaction.* Cambridge, Mass.: The MIT Press, 1996.

13. Prante, T., Meyers, B., Fitzpatrick, G. Harvel, L. D. *1st International Workshop on Exploiting Context Histories in Smart Environments. Call for papers.* 2005.

14. Shneiderman, B. (1999) *The Future of History*. Available at: http://www.cs.umd.edu/hcil/about/events /history-workshop/slides/Shneiderman/ index.htm

15. Streitz, N., Nixon, P. The disappearing computer: Introduction. *Communications ACM*, 48(3), 2005.

Author biography

Victor Kaptelinin is a Professor in the Department of Informatics, Umeå University. His current work deals with activity theory as a theoretical framework in Human-Computer Interaction, mobile technologies in education, and design of integrated digital work environments.