UMEA: Translating Interaction Histories into Project Contexts

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

ABSTRACT
Virtual environments based on the desktop metaphor provide limited support for creating and managing project-specific work contexts. The paper discusses existing approaches to supporting higher-level user activities and presents a system named UMEA (User-Monitoring Environment for Activities). The design of the system is informed by activity theory. The system: (a) organizes resources into project-related pools consisting of documents, folders, URLs, and contacts, (b) monitors user activities, (c) automatically adds new resources to pools associated with active projects, and (d) provides personal information management tools linked to individual projects. An empirical evaluation of the system is reported.

Keywords: interaction history, activity theory

INTRODUCTION
To carry out a higher-level task (or *project*) the user typically has to set up and manage a project-specific work context, that is, organize necessary resources to make them readily available when working on the project [11,17,18]. Standard virtual work environments provide little support for managing project contexts, especially when projects span several applications and involve various types of information objects.

A key idea behind the desktop metaphor underlying current operating systems is a distinction between the file system, i.e., a hierarchically organized long-term storage of potentially useful information, and the desktop, i.e., a workspace where documents and applications necessary to accomplish a task can be placed to make them easily accessible [28]. However, studies of the actual use of desktop environments, such as Mac OS or Microsoft Windows, have shown that the desktop, for a variety of reasons, is not being used as a space for integrating project-related resources [1,13,20]. A common method to set up a project context is to create a project folder. This method, however, has serious disadvantages. First, some files can be related to several projects at the same time. Second, more importantly, files are not the only type of information objects that may be needed to work on a project.

For instance, to prepare and submit a conference paper one may need a word processor to write the paper, an email program to follow announcements and updates, and a web browser to upload the paper. The paper can be located as a text file in a folder named "Conferences", email messages can be stored in a mailbox also named "Conferences", and the conference website bookmark can be placed into a bookmark "folder" named, once again, "Conferences". In other words, users may have several hierarchical systems within their virtual work environments - for example, a file system, a mailbox system, and a bookmark system, -- which contain thematically related items but are independent from each other. To complete projects that utilize several types of information objects users have to locate appropriate resources in each of these systems separately and find a way to coordinate them.

Existing systems provide little support for integrating multiple information hierarchies [see also 6]. To address this problem the paper (a) proposes an approach to providing low-overhead support for integrating various types of project-specific information, (b) presents a system developed on the basis of this approach, and (c) reports a preliminary empirical evaluation of the system.

APPROACHES TO SUPPORTING HIGHER-LEVEL USER ACTIVITIES
Current approaches to supporting higher-level user activities can be divided into four main categories: personal information management systems, dedicated project spaces, communication-based virtual environments, and non-hierarchical information space architectures.

Personal information management (PIM) systems, such as Microsoft Outlook, Palm Desktop, or ACT!, provide electronic versions of traditional organizer tools: calendars, address books, To Do lists, and notepads. They allow the user to define tasks at any level of abstraction. In particular, the user can describe a higher-level task irrespective of applications and systems, which are going to be used to accomplish the task. A limitation of PIM tools is that management of activities is separated from management of work objects, such as files and applications, necessary to carry out these activities. When a task is described with a PIM system, such a description does not support finding and arranging resources for this particular task.

Dedicated project spaces are spatially defined subsets of a virtual work environment, which provide contexts for individual projects or types of tasks. Creating a project folder, as mentioned above, is the most basic way to create a dedicated project space. The ROOMS system [11] allows the user to set up specialized workspaces ("rooms") containing resources necessary to carry out different types of activities. The basic approach underlying the ROOMS system, that is, allocating virtual spaces to individual projects, where users can place tools and materials and thereby build special-purpose work environments, has been implemented in a number of more recent systems. The Task Gallery system [24] provides users with task windows displayed as canvas hanging on the walls of a virtual 3D hallway. Another example is Manufaktur, a collaborative 3D workspace intended to support design professionals in creating and maintaining the context of a project [26]. The Kimura system [17, 29] compliments a computer monitor displaying a project space (that is, a virtual desktop) with peripheral projection displays, which show automatically generated visualizations of all user's workspaces. These visualizations help the user overview, recall, monitor, and select a workspace. Selecting a visualization on a peripheral display switches the computer monitor to the corresponding workspace, and the user finds the workspace in essentially the same state in which it was left.

The main problem with dedicated project spaces is overhead. The user has to set up an environment for a project, arrange necessary resources, and regularly update them. When the project is finished, the user may need to clean up the space. Overhead is probably the main reason why many systems for creating dedicated project spaces have not become widely used. An exception is window managers for X, such as fvwm, which support multiple workspaces. However, these workspaces are intended for various types of activity, such as handling email, reading news, or word processing, rather than individual projects [33].

Another common problem with dedicated project spaces is that they typically do not support certain types of information objects. For instance, if the user sets up a folder for a project, they can get an easy access to files but not email messages or To Do lists related to the project.

The rationale behind *communication-based virtual work environments,* such as ContactMap [21] and TaskMaster [2], is twofold. First, email has evolved from an application into a habitat, where people carry out various tasks [6]. Second, there is typically a mapping between communication patterns and patterns of user activities in general [6, 21]. Communication-based systems organize project-related resources around contacts (ContactMap) or communication threads (TaskMaster), which makes it possible to automatically link new messages and attachments to already existing projects. For instance, an attachment file received from a member of a project team can be automatically saved in an appropriate folder.

However, communication patterns do not always coincide with projects. The same team can work on a number of projects, and the same project can include a number of communication threads, which means that the user may still need to define projects explicitly. Besides, if information related to a project is not sent or received via email, it should be manually linked to the project. Therefore, even though communication-based virtual work environments appear to facilitate carrying out certain tasks, they might also create new forms of overhead.

The systems discussed so far are extensions of the traditional architecture of desktop environments, based on the hierarchical file system. Radically different approaches to creating project contexts were employed in *non-hierarchical architectures of information spaces:* Lifestreams [9,10] and Presto [6].

The Lifestreams system organizes documents according to their chronological order, which supports simple and intuitive strategies of finding documents in electronic workspaces. People often associate events with certain time; so knowing the approximate time of an event can help identify the fragment of the Lifestreams sequence, which contains the target document. The system also has the advantage of combining document space management with management of individual and group activities. For instance, a draft paper to be discussed by a group of co-authors can be placed at a certain time in the future, which can facilitate either access to necessary resources or coordination of group work. The user can create "sub-streams" of documents by entering selection criteria

Presto [6] is a system developed within the framework of the Xerox PARC Placeless Documents project. The system provides an infrastructure that allows for flexible and dynamic generation of collections of documents by defining appropriate sets of attributes.

Both Lifestreams and Presto are intended to support creating complex information structures that match certain criteria and can include various types of resources. These structures are dynamically generated as a result of computation conducted on the whole collection of available information. Both systems can in principle be employed to create project-specific workspaces containing various types of information. However, to create a project context the user has to define formal criteria for selecting relevant information, which in many cases can be difficult or even impossible to do.

Users, especially in the beginning of a project, often have a vague idea of which attributes are important. The set of criteria, according to which information should be considered relevant, may develop over the course of working on a project. Besides, these criteria are likely to be implicit. Even if the criteria are clear and explicit, using them could require an excessive effort. For instance, when working on a project the user might come across a potentially useful email address. It is typically much easier to simply add the address to information related to the project than to find selection criteria, according to which the system will automatically generate a "sub-stream" or a document collection that would include the address. Therefore, a potential problem with non-hierarchical architectures is a lack of support for situated and opportunistic work practices and tacit knowledge of the user.

CREATING PROJECT CONTEXTS THROUGH INTERACTION HISTORIES

The work reported in this paper has been informed by activity theory. Activity theory is an approach in psychology and social sciences, which deals with purposeful interactions of active subjects with the objective world [19]. These interactions, or activities, are understood as social, hierarchically organized, developing, and mediated by tools. During the last decade there has been a growing interest in activity theory as a potential theoretical framework for Human-Computer Interaction (HCI) [3, 14, 19].

The space limitations of this paper do not allow for a detailed discussion of the implications of activity theory for design of systems supporting higher-level user activities. The main relevant points can be briefly summarized as follows. First, the meaning of various objects that constitute environments, both physical and virtual, is determined by the context of activity, that is, by the relation of the objects to subject's motives and goals. Therefore, virtual work environments should help the user organize resources around their meaningful goals. To support coordination of various levels of activity a system should integrate higher-level representations of goals (such as representations provided by PIM systems) with resources needed to accomplish these goals (such as applications or documents). Besides, in everyday life individual's focus of attention is constantly switching between different conscious goals and corresponding contexts [4, 8]. Switching contexts is seldom quick and effortless. A system might make it faster and easier. Second, since human activities are situated, determined by their physical and social contexts, a system should provide support for a wide variety of actual work practices. Third, a system, as a mediational artifact, should be as transparent as possible to allow the user to focus on meaningful goals rather than interaction with technology.

Therefore, systems supporting higher-level user activities should meet the following requirements: (1) integrate personal information management, communication, and management of tools and materials, (2) capitalize upon actual work practices, (3) and minimize overhead and make the benefits of creating project environments apparent to the user.

The approach described in this paper (see also [13,15,16]) is an attempt to meet the above requirements by providing a system, which: (a) makes it possible for the user to directly indicate a higher-level task, that is, a project, (b) monitors user activities and tracks resources used when carrying out the project, and (c) automatically organizes and updates these resources to make them easily available to the user when he or she resumes working on the project.

This approach builds upon HCI research and development related to interaction histories [5,12,23,27,31]. It proposes a novel strategy for creating and managing project-specific work contexts. To the best of my knowledge, the only reported attempt to combine support of higher-level activities with monitoring the user is the Kimura system, mentioned above [17,29]. However, the use of interaction histories in Kimura, that is, generating awareness cues, is different from the use of interaction histories within the approach presented here, that is, facilitating access to specific resources.

THE UMEA SYSTEM

General architecture

A system named UMEA (User-Monitoring Environment for Activities) was developed on the basis of the approach described above[1].

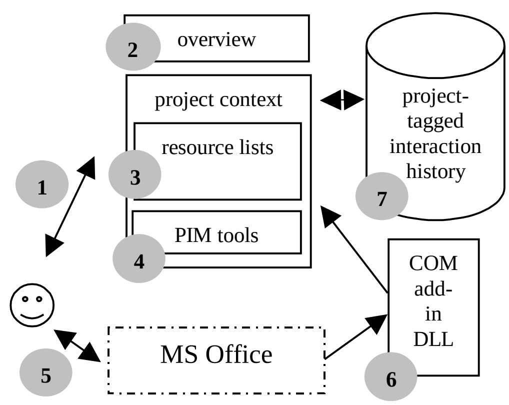

Figure 1. The general architecture of the system

The UMEA system is an application running under Microsoft Windows. The application is written in Microsoft Visual C++ 6.0 using DISCo (www.disco.ru) GUI library. The general architecture of the system is shown in Figure 1.

The system can be either in the foreground mode (1) or the background mode (5). In the foreground mode the system presents users with an overview of their projects (2). The

[1] A brief technical description of the application is given in [16].

user can select a project and then open a resource (a document, a folder, a web page, or a contact) by choosing it on a pop-up menu displaying a project-specific list of resources (3). The user can also set up a new project. Besides, a number of personal information management (PIM) tools (4) is provided to the user.

When the system runs in the background (5), it receives Microsoft Office 2000 events, such as opening a web page, printing a file, or sending an email. The events are received through a COM add-in DLL implemented as an IDTExtensibility2 object (6). When the system receives an event, the event is tagged to the currently active project and saved in a Microsoft Access database (7). If the event is associated with a new resource, that is, a resource that has not yet been used within the currently active project, this resource is added to the appropriate list of project-specific resources (3).

The main technical limitation of the current version is that interaction histories can only contain events received from Microsoft Office applications. This problem is somewhat alleviated by making it possible for the user to directly open any of the folders previously opened within the project, so that the user can get access to project-related files, even if the files are not created by Office applications.

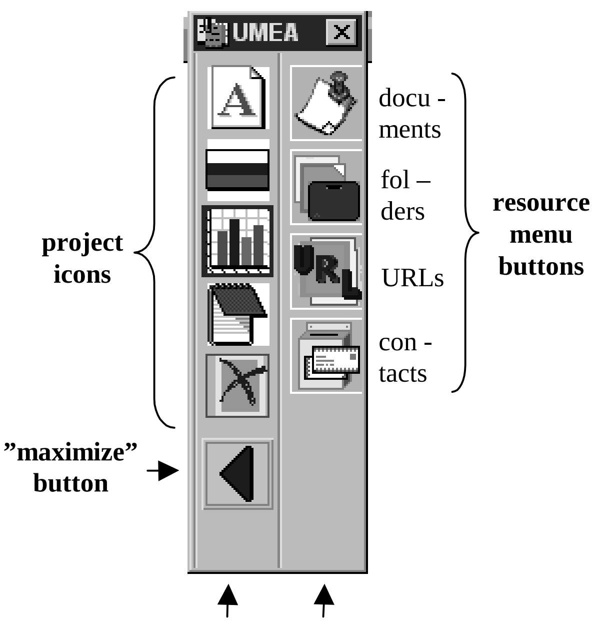

Figure 2. The minimized overview

User interface

The user can switch between three different views of the system: the minimized overview, the maximized overview, and project windows. The minimized overview (Fig. 2) consists of two vertical panels: (a) the project panel displaying project icons, and (b) the resource panel, displaying buttons, corresponding to four types of project-specific resources: documents, folders, URLs, and contacts (email addresses).

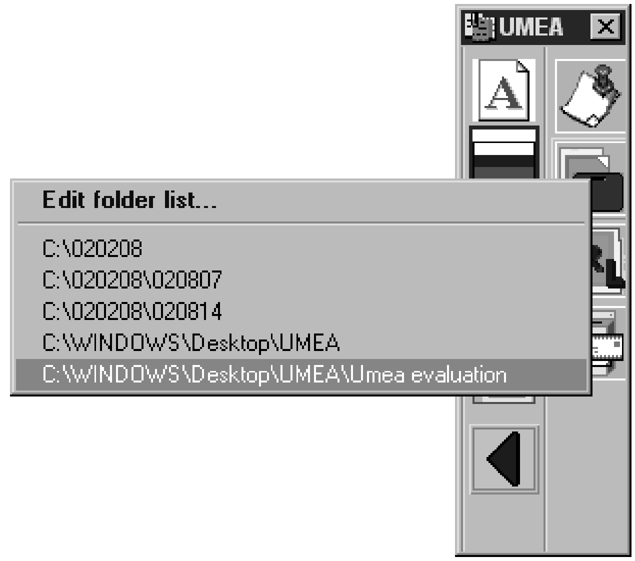

Figure 3. The minimized overview with an open pop-up menu containing a list of folders.

By selecting a project on the project panel the user determines, which lists of resources can be accessed through buttons on the resource panel. When a project is selected, clicking on one of the resource menu buttons opens a corresponding list of resources related to that project (Fig. 3). Choosing an item on the list opens a resource, that is, a document, a folder, a web page, or a new email message addressed to the selected contact.

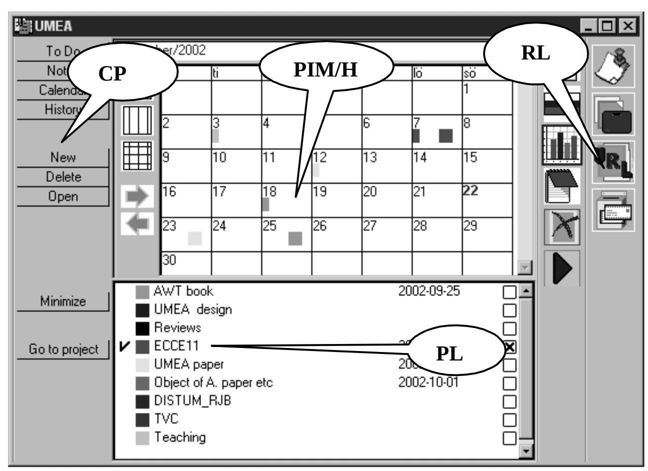

Figure 4. The maximized overview
(PL – projects list, PIM/H – PIM/ History area displaying the calendar, CP – control panel, RL – resource lists)

The user can edit lists of resources by manually adding or deleting items. The user can also assign ranks to items on a list. Items are organized into groups according to their ranks, and groups with higher ranks are displayed higher on the list.

The maximized overview (Fig. 4) is an extended version of the minimized overview. In addition to the project panel and the resource panel it displays: (a) a complete list of projects, (b) a PIM/ History area displaying PIM tools and the interaction history of the active project, and (c) a control panel. PIM entries are linked to projects. Information associated with a certain date, such as a project deadline, is automatically displayed in the calendar as a verbal description (the day view) or a bar of the color of its respective project (the month view and the week view).

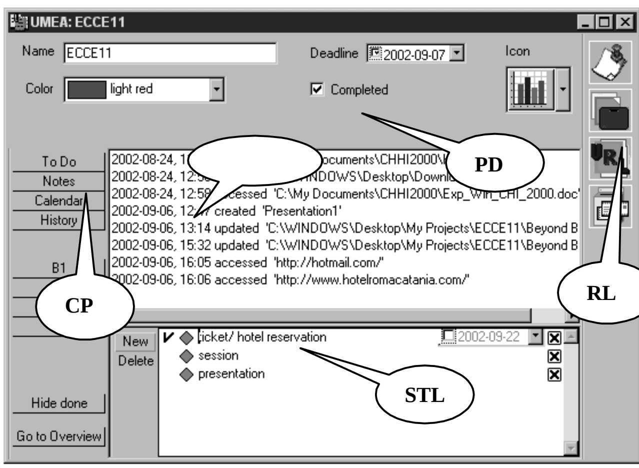

Figure 5. A project window
(PD – project details area, STL – sub-task list, PIM/H – PIM/ History area displaying history, CP – control panel, RL – resource lists)

The history (Fig. 5) is a project-specific interaction log, which contains timestamped events describing objects (such as documents or folders) and actions (such as printing or accessing). The history can be manually edited by the user. For instance, the user can select a sequence of events in project's interaction history and delete the sequence or re-assign it to another project.

From the maximized overview the user can open project windows of selected projects (Fig. 5). Project windows look similar to the maximized overview but only display information related to one project. When a project window is open the user can edit project attributes (the name, the associated color, the icon, and the deadline) or decompose the project into subtasks. A project window opens automatically when the user creates a new project.

Project-specific histories and resource lists are continuously updated in the same way regardless of whether an overview window or a project view window is displayed.

Sample use scenario

A possible scenario of using the system is as follows. The user launches UMEA and opens the maximized overview displaying the list all ongoing projects and a calendar. Then the user decides, which project he or she is going to work on. Making that decision involves browsing the list of projects and checking approaching deadlines in the calendar, as well as checking notes and To Do lists. To view notes and To Do entries associated with a project the user clicks on the name of the project and selects an appropriate PIM tool. Occasionally the user opens lists of documents related to a project to understand the status of a project. When the user eventually decides that the selected project is the one he or she is going to focus on, the user immediately continues working on the project. The files, folders, URLs, and contacts previously used within the project are easily accessible. The user opens some of these resources by selecting them on appropriate resource lists. The user edits and saves documents, sends emails, and browses the web. New file names, email addresses, and URLs are automatically added to respective lists of resources related to the project. The user also creates and edits notes, To Do lists, and calendar entries, which are automatically linked to the project. When the user wants to switch to another project, he or she clicks on the name of that project on the project list and get an immediate access to a new set of resources.

The user is motivated to re-enter project environments to get an access to project resources. At the same time, by re-entering a project context the user makes it possible for the system to continue creating a project-specific interaction history and develop an even more elaborated project context.

If the user discovers that a wrong project name has been selected and a part of the interaction history of project A is in fact related to project B, the user re-assigns that part of the history to project B. The resources used in that sequence of events are deleted from resource lists of project A and added to resource lists of project B.

Design considerations: A retrospective account

The design of the UMEA system is motivated by an attempt to meet three requirements formulated earlier in the paper.

Integrating personal information management, communication, and management of tools and materials. The UMEA system provides an integrated access to documents, PIM tools, and communication facilities. The user can check a project-specific To Do list, open relevant files and web pages, and send email messages to people on the project-specific contact list, all from the same project context.

Capitalizing upon the actual work practices of the users. The system does not prescribe the user how they should do their job. Users can carry out their tasks however they wish, without being constrained by the system. The system monitors whatever the user is doing and updates project contexts on the basis of this monitoring. Therefore, the system appears to be compatible with a wide range of work practices.

Minimizing overhead and making the benefits of creating project environments apparent to the user. The user is expected to start enjoying some benefits of the system as soon as a new project is created and given a meaningful name. Adding more information about the project, such as selecting an icon or a color, setting a deadline, or describing subtasks, opens up new possibilities for managing projects. Therefore, the user can gradually learn more and more features of the system and correspondingly obtain more benefits.

First experiences with the system

Currently empirical evidence about the actual use of the system comes from two sources. The first source is the experience of the author, who has been using the system on an everyday basis for several months and for whom the system became the main tool for personal project management.

The second source is an empirical study, in which a group of eight users evaluated the system over the period of two to six weeks. The group consisted of eight native Swedish speakers, from 21 to 51 years old, and included undergraduate and graduate students, university teachers, a professional programmer, and a secretary. The participants were asked to try the system and provide comments on its advantages and disadvantages; they were free to stop using the system at any time. Three ("local") participants were given a 10-15 minutes introduction into how to use the system, while the other five ("remote") participants received system files electronically with a very basic explanation. They had to learn the system mostly by trial and error. Questionnaires were administered before and after the trial period. The pre-questionnaire was oriented to subjects' practices related to managing their virtual workspaces. The post-questionnaire focused on individual use patterns, general assessment of both the system and the underlying approach, and participants' opinions about specific advantages and disadvantages of the system.

All participants but one positively evaluated the underlying approach. One participant was generally uncomfortable about dividing his activities into separate projects. As to the current version of the system, most participants (6 out of 8) found it useful. However, they were more positive about the general approach than about its current implementation. On the scale from "-2" ("totally useless") to "+2" ("very useful") the mean value assigned to the general approach was +1.7, while the mean value assigned to the actual system was +0.8.

The advantages of the system, mentioned by the participants, included: (a) an access to various types of resources related to a project "from within one place", (b) an overview of ongoing projects, (c) a possibility to instantly switch back and forth between projects, and (d) the help provided by the system in recalling the context of a project, which made it easier to resume working on the project after a break.

Two main problems with the system were reported. First, there was a need to manually clean up resource lists and/or interaction histories from time to time to delete irrelevant items. The problem was mostly caused by system's automatic unselective inclusion of all resources used within a project to resource lists, even if the resources were not really important. In some cases extraneous items appeared on the lists because users were actually working on projects different from the ones they selected.

Second, some participants experienced difficulties with understanding the user interface and the functionality of the system. These difficulties were repeatedly mentioned by the "remote" participants. However, none of the "local" participants reported this problem. It appears the current version of the system can be learnt in a short time, but not by trial and error.

CONCLUSIONS AND PROSPECTS FOR FUTURE WORK

First experiences with the system indicate that it addresses a real need of users for a low-overhead integration of various types of information around higher-level, meaningful goals. The current version of the system appears to be practically useful, at least to some users. At the same time, there is a considerable potential for development of both the system and the underlying approach.

The most evident problem that needs to be solved to make the system more usable is minimizing the effort necessary to clean up lists of resources, that is, to get rid of constantly accumulating irrelevant items. A set of heuristics have been developed to avoid excessive cluttering of the lists.

"Background" resources. If the user regularly performs background tasks, such as checking the news on the web, he or she can inadvertently add irrelevant resources, such as a news website URL, to several foreground projects. A possible solution to this problem is to set up a list of "background" resources, which can be recognized by the system as resources that should not be linked to any of user's projects.

Automatic ranking of resources. Interaction histories contain numerous clues on the relative importance of the resources used within a project. If the user saves successive versions of a document within a project, it means the document is probably essential to the project. On the other hand, if the user opens a document only once and closes it immediately after opening, the document is perhaps not very important and could even be opened by mistake. If the system automatically sorts out resources by assigning them ranks, the most important resources can be displayed higher on the list to make them more accessible. Less important resources can be automatically moved down the list or hidden.

Automatic switching between project contexts. Many resources are used exclusively in one project or can be modified only within a specific project. These resources can be automatically or manually linked to their respective projects. Using these resources can be used by the system as an indicator that the user is working on a certain project, even if the user does not explicitly indicate switching to that project. For instance, if the user is working on a document in one project context and then saves the latest version of the document in another context, the system can infer that the active project is indicated by mistake. The system can automatically switch to the correct project or, alternatively, display a warning message and let the user select the active project.

These heuristics are expected to be implemented in the next version of the system. Because of substantial individual differences between users the heuristics need to be implemented so that the user could select options, such as criteria of filtering out irrelevant information or conditions for switching between projects, which fit his or her preferences and work practices.

Future work will also have to deal with two problems, which present a challenge not only to the current version of the system, but to the very idea of automatically translating interaction histories into project contexts, as well. The first is the case of multi-purpose activities, such as checking email, surfing the web, or attending a meeting with a long list of items on the agenda. Constant switching from one topic (or project) to another is a characteristic feature of such activities. If a person has to indicate each time that he or she is changing contexts, the whole activity could be disorganized. To prevent that users should be able to postpone or avoid mapping their actions to specific projects until after a multi-purpose activity session. The most promising way to support users in doing so appears to be an automatic analysis of the content of email messages, web pages, etc., and presenting the user with suggestions on how specific fragments of multi-purpose activities can be mapped to different projects.

Besides, a serious limitation of most computer systems is their inability to monitor user behaviors in the physical world, which are critically important for many real-life tasks and projects. However, an increasing number of systems allow for monitoring various types of activities beyond human-computer interaction in the traditional sense. For instance, the Active Badge system [30] and the Kimura system [17,29] monitor physical locations of users. The Magic Touch system [22] keeps track of physical documents in the physical workspace by reading tags attached to the documents. Finally, the RoamWare system [32] creates an interaction history of physical meetings by detecting mobile devices (such as PDAs, mobile phones, or laptop computers) in the proximity of the user. This history is utilized to facilitate and support communication with colleagues in between physical meetings. Therefore, the general trend towards personal technologies [26] is likely to make the above limitation less severe in the near future.

Finally, future work can explore a number of new possibilities, such as:

supporting not only individual but also group activities;

creating distributed work environments composed of a configuration of technologies, such as desktop computers, laptop computers, PDAs, and other mobile devices;

using various representations of interaction histories to help users reflect on their work, create reports and accounts; and

helping users organize and clean up their virtual environments by using interaction histories for integrating and structuring information on the basis of data about the frequency, recency, and context of its use.

ACKNOWLEDGEMENTS

This work was funded by VINNOVA, the Swedish Agency for Innovation Systems (www.vinnova.se). I would like to thank Allen Cypher, Robert Holmberg, Donskoy Interactive Solutions Company (DISCo), study participants, and CHI reviewers for their help and support.

REFERENCES

1. Barreau, D., Nardi, B. Finding and reminding: File organization from the Desktop. *SIGCHI Bulletin*, July 1995.
2. Belotti, V., Duchenaeut, N., Howard, M., Smith, J. Taskmaster: Recasting email as task management. *Position paper for CSCW 2002 workshop Re-designing Email for the 21st Century* (New Orleans, Louisiana, November 16, 2002), 2002.
3. Bødker, S. *Through the interface: A human activity approach to user interface design*. Hillsdale, NJ: Lawrence Erlbaum, 1991.
4. 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.
5. Cypher, A. (ed.) *Watch what I do: Programming by demonstration.* Cambridge, Mass.: The MIT Press, 1993.
6. Dourish, P., Edwards, W. K., LaMarca, A., Salisbury, M. Presto: An Experimental Architecture for Fluid Interactive Document Spaces. *ACM TOCHI*, 6 (2), 1999.
7. Ducheneaut, N., Belotti, V. Email as habitat: An exploration of embedded personal information management. *Interactions*, 8 (5), 2001.
8. Erickson, T. The Design and Long-Term Use of a Personal Electronic Notebook: A Reflective Analysis. *Proceedings of the CHI'96 Conference on Human*

Factors in Computing Systems (Vancouver, Canada, April 1996). ACM Press, 1996.

9. Fertig, S., Freeman, E., Gelernter, D. Lifestreams: An Alternative to the Desktop Metaphor. *CHI'96 Conference on Human Factors in Computing Systems. Video Program* (Vancouver, Canada, April 1996). ACM Press, 1996.
10. Fertig, S., Freeman, E., Gelernter, D. "Finding and Reminding" Reconsidered. *SIGCHI Bulletin*, v. 28, January 1996.
11. Henderson, A., Card, S. K. Rooms: The use of virtual workspaces to reduce space contention in a window-based graphical user interface. *ACM Transactions on Graphics*, 5 (3), 1986.
12. 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.
13. Kaptelinin, V. Creating Computer-Based Work Environments: An Empirical Study of Macintosh Users. *Proceedings of the ACM SIGCPR/SIGMIS'96 Conference* (Denver, Colorado, April 11-13, 1996)
14. Kaptelinin, V. Computer-mediated activity: Functional organs in social and developmental contexts. In: B. Nardi (ed.) *Context and Consciousness: Activity Theory and Human-Computer Interaction*. MIT Press, 1996.
15. Kaptelinin, V. *Supporting Higher-Level User Actions with Representations of Project Contexts and Interaction Histories.* Working Paper. Department of Informatics, Umeå University, 1997.
16. Kaptelinin, V. UMEA: User-Monitoring Environment for Activities. *UIST '02 Companion. Posters and Demos from the 15th Annual ACM Symposium on User Interface Software and Technology* (Paris, France, October 27-30, 2002), 2002.
17. MacIntyre, B., Mynatt, E. D., Voida, S., Hansen, K. M., Tullio, J., Corso, G. M. Support for multitasking and background awareness using interactive peripheral displays. *CHI Letters, 3 (2), 2001.*
18. Malone, T. W. How do people organize their desks? Implications for the design of office information systems. *ACM Transactions on Office Information Systems* 1, 1983.
19. Nardi, B. (ed.) *Context and Consciousness: Activity Theory and Human-Computer Interaction*. MIT Press, 1996.
20. Nardi, B., Anderson, K. and Erickson, T. *Filing and finding computer files.* Technical Report # 118. Cupertino: Apple Computer, Inc., 1994.
21. Nardi, B., Whittaker, S., Isaacs, E., Creech, M., Johnson, J., Hainsworth, J. Integrating communication and information through ContactMap. *CACM*, 45 (4), 2002.
22. Pederson, T. Magic Touch: A Simple Object Location Tracking System Enabling the Development of Physical-Virtual Artefacts in Office Environments. *Journal of Personal and Ubiquitous Computing*, 5, 2001.
23. Rich, C., Sidner, C. L. Segmented Interaction History in a Collaborative Interface Agent Planning Based Approaches. *Proceedings of the 1997 International Conference on Intelligent User Interfaces*, 1997.
24. Robertson, G., van Dantzich, M., Robbins, D., Czerwinski, M., Hinckly, K., Risden, K., Thiel, D., Gorokhovsky, V. The Task Gallery: A 3D Window Manager. *Proceedings of the CHI 2000 Conference on Human Factors in Computing Systems* (The Hague, The Netherlands, April 1-6, 2000). ACM Press, 2000.
25. Shapiro, D. Technologies for self-organization. In: Dittrich, Y., Floyd, C., Jayaratna, N., Kensing, F., and Klischewski, R. (eds.) *Social Thinking - Software Practice. Dagstuhl Seminar Report* (Dagstuhl, Germany, September 1999), 1999.
26. Sharples, M. The design of personal technologies for lifelong learning. *Computers and Education*, 34, 2000.
27. Shneiderman, B. (1999) *The Future of History*. Available at: http://www.cs.umd.edu/hcil/about/events /history-workshop/slides/Shneiderman/ index.htm/
28. Smith, D., Irby, C., Kimball, R., Verplank, W., Harslem., E. Designing the Star user interface. *Byte*, 7 (4), April 1982.
29. Voida, S., Mynatt, E. D., MacIntyre, B., Corso G. M. Integrating virtual and physical context to support knowledge workers. *IEEE Pervasive Computing*, July-September 2002.
30. Want, R., Hopper, A. The active badge locator system. *ACM Transaction on Office Information Systems*, 10 (1), 1992.
31. Wexelblat, A., Maes, P. Footprints: History-Rich Tools for Information Foraging. *Proceedings of the CHI'99 Conference on Human Factors in Computing Systems* (Pittsburgh, PA, May 1999)
32. Wiberg, M. Bridging physical and virtual group meetings with a PC and multiple hand-held devices. *Adjunct Proceedings of the ACM CHI'2000 Conference on Human Factors in Computing Systems* (The Hague, The Netherlands, April 1-6, 2000).
33. *Window Managers for X: The Basics*. Available at: http://www.plig.org/xwinman/basics.html