TaskTracer: Enhancing Personal Information Management Through Machine Learning

Simone Stumpf
Oregon State University
School of Electrical Engineering and Computer Science
Corvallis, OR
stumpf@eecs.oregonstate.edu

Jon Herlocker
Oregon State University
School of Electrical Engineering and Computer Science
Corvallis, OR
herlock@eecs.oregonstate.edu

ABSTRACT
TaskTracer is a research project at Oregon State University to identify innovative technological solutions to information overload and task management at the personal and workgroup levels. The core TaskTracer hypothesis is that people naturally want to organize their information, regardless of data type or storage location, into units that they think of as projects or activities or tasks. By building software that is aware of those tasks, we can greatly decrease costs, errors, and frustration. We believe that a database of past user actions on the desktop, segmented by user task, can enable substantial productivity-enhancing features when combined with machine learning. In our research, we are working to demonstrate that our system can help people find their task-related information faster, recover faster and better from interruptions, and maintain better awareness of personal and workgroup tasks. We have taken a very practical approach, building fully-functional prototypes and deploying them in the wild. Our current research is focused on several aspects: designing improved machine learning techniques to support automatic labeling of events; identifying machine learning techniques in intelligent user interfaces that balance automatic assistance and usability, building innovative interfaces to leverage our activity data, and measuring the impact of such innovative interfaces on reducing costs, errors, and frustration.

Categories and Subject Descriptors
H.3 [**Information Systems**]: Information Storage and Retrieval, H.5 Information Interfaces and Presentation

General Terms
Human Factors

Keywords
TaskTracer, Personal Information Management, Machine Learning

1. CORE TASKTRACER HYPOTHESIS
The core TaskTracer hypothesis is that knowledge workers organize their work into tasks, i.e. activities that make sense to them. Associated with each task is a set of information resources (documents, electronic messages, contacts, etc.) and tools (computer applications, databases) employed to access and manipulate these resources. For instance, in writing a paper, the author may seek content from PowerPoint presentations, existing bibliographies in other documents, web searches, emails to colleagues, as well as working with web-based processes and programs to complete or submit or publish the finished document.

We assume that at any instant in time, the individual is only working on a single task, but may switch rapidly between tasks. Each time there is a task switch, the user may need to access a completely different set of information resources.

Current applications and operating systems have no concept of this multi-tasking. Even worse, users are forced to organize their information by application (folders in email, folders on the network share, collections of bookmarks, etc.). Since work processes are generally organized by task, users are forced to duplicate task-oriented organizational hierarchies in multiple applications. This disconnection between the task-oriented projects of the knowledge worker and the non-task-aware personal computer leads to considerable unnecessary errors and overhead costs.

In general we focus on reducing costs (physical or cognitive) and errors to increase productivity. We believe that we can improve productivity through intelligent software layered on top of the existing Microsoft Windows environment, such as that

- All resources (email messages, documents, web bookmarks, contacts, pictures, etc) are organized together for workers by their tasks.
- Personal information management is supported better, through automatic recording of what a user did during a task.
- Tools are available to help users better recover context in a task after an interruption, showing what they were working on before the interruption.
- Applications are aware of their current task, and adapt to support it, like knowing exactly where a user might want to save a file.
- We can help the reuse of information for future similar tasks.

2. TASKTRACER DATA COLLECTION AND DATABASE
Through an extensive data-collection framework for the personal computer, TaskTracer collects detailed observations of user interactions in the Microsoft Windows and Office environment: email, word processing, spreadsheets, and Internet browsers (for

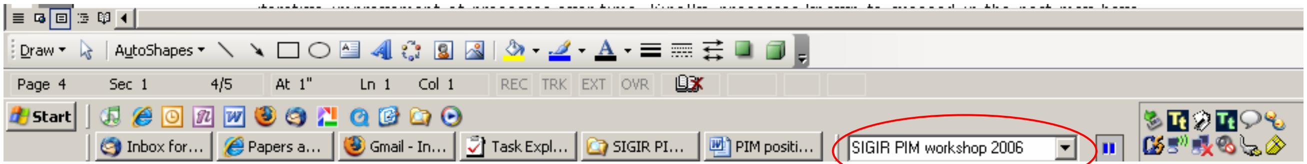

Figure 1. TaskSelector

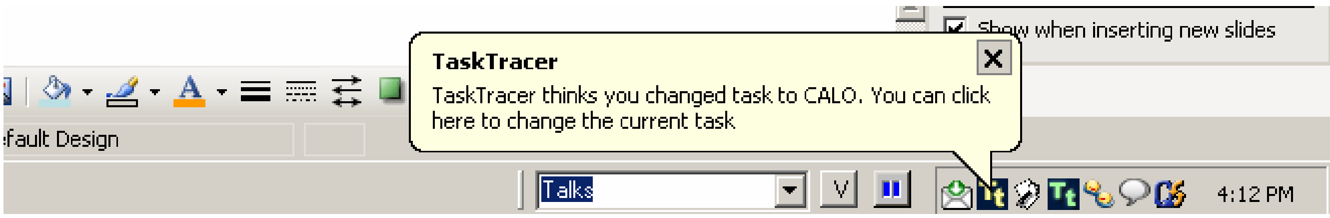

Figure 2. TaskPredictor notification

more details see [1]). We call such a record of a task a task profile. Listeners for new applications, regardless of language environment or even network location, can be written in a matter of hours. This rich database of user activity is the core resource in TaskTracer.

In the initial training phase, at the start of a task, users manually specify what task they are doing and from then on each user interface (UI) event will be automatically tagged with this particular task identifier. This reduces manual interaction by the user for information resource association since they will only need to tell TaskTracer when they switch from one task to another. Tasks can be specified quickly via the TaskSelector (Figure 1).

2.1 TaskPredictor

We are also designing the system so that there are rewards for informing the system that the task has changed. To build a tool that can benefit a wide audience, we believe that it is necessary for the system to provide some sort of automatic inference of the user's current task, to reduce the cost of task switches (Figure 2).

The TaskPredictor leverages the raw event stream to build a series of prediction segments which consist of unbroken sequences of time in which a particular window has focus and the name of the document in that window does not change (For further information on the technical details of the implementation please see [2]).

2.2 TaskExplorer

From the task profile, we can build a list of all resources that have been used while working on a task – this allows us to create a user interface component that displays in a unified fashion all the information resources associated with a selected task (Figure 3). The user is able to open any resource from this UI, regardless of application types, addressing the information fragmentation problem. By default, this list of resources in TaskExplorer is ordered by date accessed, so this user interface can also help workers to regain context after interruptions – the list of most recently accessed resource serves as a cue for recall. This is

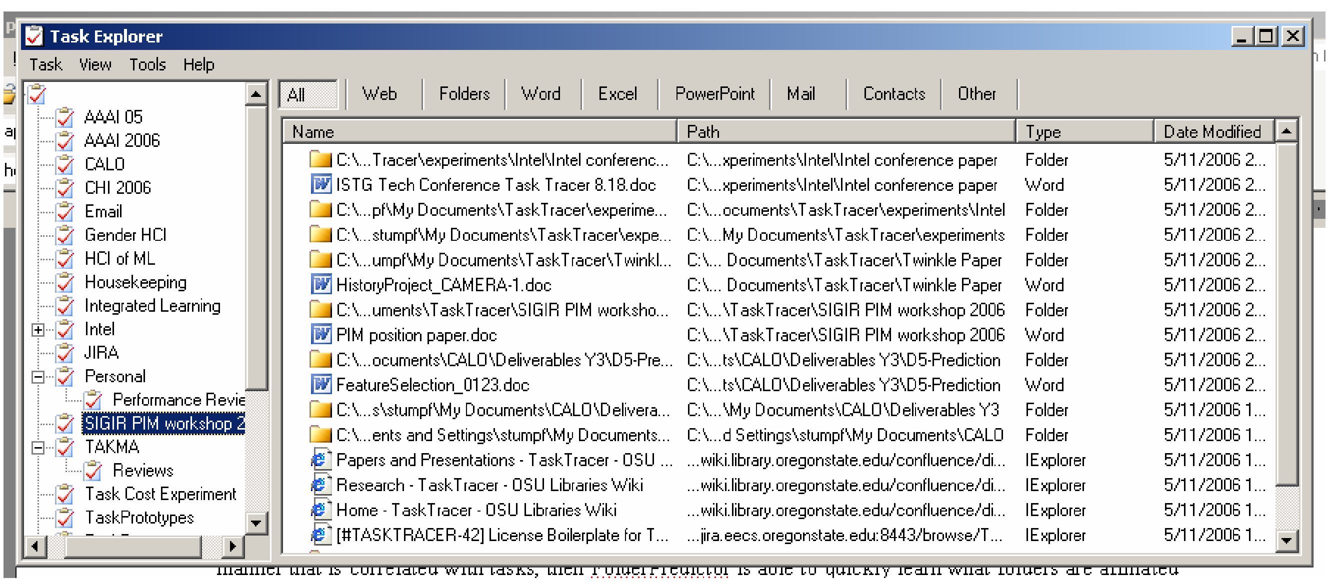

Figure 3. TaskExplorer

particularly valuable if a task is returned to after an interruption of days, weeks, or years.

2.3 FolderPredictor

In addition to providing applications like the TaskExplorer, we can configure individual applications so that they adapt to support the current task. One example of this is the FolderPredictor [3]. The FolderPredictor hooks into all the standard file open dialog boxes in Microsoft, and predicts what folders the worker will want to use in locating or saving files. If workers organize their files into folders in a manner that is correlated with tasks, then FolderPredictor is able to quickly learn what folders are affiliated with a task. Then those folders are placed one click away, resulting in substantially less time searching and navigating to the appropriate folder (Figure 4).

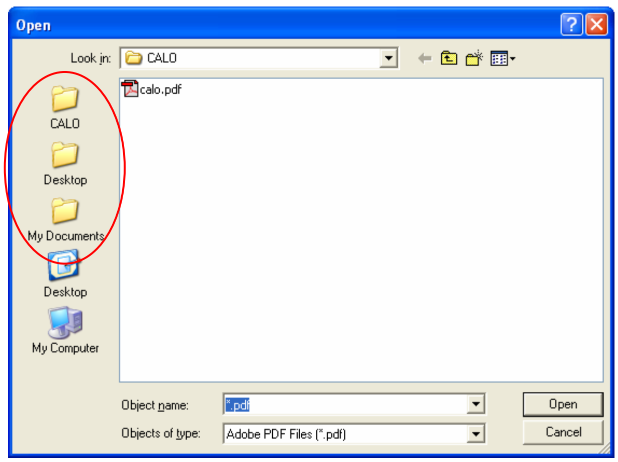

Figure 4. FolderPredictor

3. CURRENT WORK

We are currently in the process of extending TaskTracer capabilities for prediction, information organization and information reuse. The usefulness of our current work is being evaluated in field studies in the summer 2006.

3.1 EmailPredictor

EmailPredictor works with Microsoft Outlook and aims to predict the task for incoming email messages. Messages are automatically tagged with predicted tasks and can be searched, sorted and filtered on those tasks. Outlook rules can be set up to automatically folder these emails if the user desires.

3.2 Task-aware Windows Explorer

While there is now better desktop searching, many people still prefer to browse through folder structures. We have adapted the functionality of Windows Explorer by extending FolderPredictor. We provide a "prediction toolbar" in Windows Explorer that shows the three most likely folders that the user may want to visit associated with the current task.

3.3 Revisiting web pages

Users face information overload when trying to locate specific webpages, especially as a majority of visits to webpages are revisits. While automatically created browsing history lists offer a potential low-cost solution to re-locating webpages, even short browsing sessions generate a glut of webpages that do not relate to the user's information need or have no revisit value. We use a combination learning technique to better support web users who want to return to information on a webpage that they have previously visited by building more useful history lists [4].

3.4 TaskPrototypes

Our goal with TaskTracer is also to explicitly support reuse of processes and resources. One common method of reuse is what we call document templating. For example, researchers must write grant proposals every year yet they reuse information from other documents as templates. We are allowing tasks to be designated as templates for other tasks, where the resource content can be reused.

4. FUTURE WORK

4.1 Group Information Management (GIM)

Another characteristic is that the work of knowledge workers is information intensive as well as interdependent with the work of others. We are in the process of addressing this problem. We aim to support two ways of collaborative work: shared tasks and group tasks. In a shared task, a user makes her task information available to someone else. For example, a researcher who has successfully written grant proposals in the past may make the record of that task available to a new hire (this is an instance of document templating in a one-way collaborative setting). In a group task, a number of knowledge workers who work on a task together can create a joint task, in which each contributes and receives task information for the group.

5. ACKNOWLEDGMENTS

This project is supported in part by Intel, the National Science Foundation under grant IIS-0133994 and by the Defense Advanced Research Projects Agency under grant no. HR0011-04-1-0005 and contract no. NBCHD030010.

6. REFERENCES

[1] Dragunov, A., Dietterich, T. G., Johnsrude, K., McLaughlin, M., Li, L. and Herlocker, J. L. Tasktracer: A Desktop Environment to Support Multi-Tasking Knowledge Workers. In Proceedings of the 10th International Conference on Intelligent User Interfaces, San Diego, CA, 2005.

[2] Shen, J., Li, L., Dietterich, T.G., Herlocker, J. L. A hybrid learning system for recognizing user tasks from desk activities and email messages. In Proceedings of the 11th International Conference on Intelligent User Interfaces, pp. 86-92, Sydney, Australia, 2006.

[3] Bao, X., Herlocker, J.L., Dietterich, T.G.. Fewer Clicks and Less Frustration: Reducing the Cost of Reaching the Right Folder. In Proceedings of the 11th International Conference on Intelligent User Interfaces, pp. 178-185, Sydney, Australia, 2006.

[4] Lettkeman, T., Stumpf, S., Irvine, J., Herlocker, J. Predicting Task-Specific Webpages for Revisiting. Twentieth National Conference on Artificial Intelligence (AAAI-06), Boston, Massachusetts, July 16-20, 2006