Tracking task context to support resumption

Damien Clauzel, Claudia Roda, Georgi Stojanov

American University of Paris
Computer Science Department
147 rue de Grenelle
75007 Paris, France

{dClauzel, cRoda, gStojanov}@AUP.fr

Abstract

This paper presents our current research on task context. We describe the approaches investigated to collect data from user activities. These data are used to provide support to attention and motivation. We also briefly describe a proof-of-concept prototype.

Keywords: activity analysis, task context, attention, agents, AtGentive

Acknowledgement

The work described in this paper was partially sponsored by the European Community under the FP6 framework project AtGentive IST-4-027529-STP.

1 What do we want to do?

Current desktop interfaces force an "application oriented" rather than "task oriented" approach to computer based activities. In order to complete a task (say write a report) the user is forced to fragment the task in subtasks (such as collecting data from a word processor to write some text, collecting data from a spreadsheet in order to paste it in the text). This artificial fragmentation of the original task imposes an increased cognitive load on the user. In situations in which the user needs to switch between several tasks and to interrupt one task to return to it later, such cognitive load increases further. This type of multitasking is by no means unusual in today's working and learning environments; see for example Thompson [4].

The work briefly described in this paper aims at increasing our understanding of whether a task oriented (as opposed to application oriented) approach to desktop user interfaces would reduce cognitive load in the performance of complex tasks in multitasking environments. In particular, we want to study

whether restoring task context significantly improves attention allocation at task resumption. Our objective is to assist the user in restoring the context of interrupted tasks.

Task context has been investigated in several ways. Some researches are activity oriented, while some others are focusing on interfaces and HCI (Lewis and Rieman [2], Robertson [3]). But the question of task context is not limited to computer environments; cognitive neuroscience is also investigating the meaning of context, for example in mammal brain (Johnston and Everling [1]).

We give an operational and interface oriented definition of task context. A task context is the ensemble of the interface elements accessed by the user in the performance of a task. These include: windows, menus, icons, texts, history of elements, cursor and caret position, keyboard and mouse uses. Our current definition is intentionally restrained so that we can quickly work on a basic core of data and provide an initial task support to the user. Later, we will expand our current definition to include other aspects of task context such as parameters coming from the external environment (room noise level, psycho-physiological measures, etc).

2 Data collection method

As stated above, for the time being, we concentrate on tracking user activity on the computer (e.g. keyboard, windows). To this end, we need to collect enough information on the environment in which the task takes place. So in our research we use the following approaches for collecting data (see bellow). In the future we expect to integrate these methods with other measurements of user activity, e.g. gaze and posture tracking, video camera, psycho-physiological measurements, etc.

Screen mapping: One approach for collecting informations on user activity is to track the cursor position and graphical events (windows moving, menus unfolding, etc.) so we can map in terms of $(x; y)$ coordinates what the user is doing. The non-trivial task is then to map those coordinates to specific activities, which can lead to complete mistakes in interpretation if a user action is missed; it also induces problems when dealing with dynamic and customisable interfaces.

Smart applications: This solution consists of having customized applications which will send an event when a functionality (menus entry, icon, widget...) is activated. This implies having tracking ready software, which reduces the user choices in software available for working. The benefit of this solution is to have plenty of information on user actions.

Spywares: This method relays on the use of a keylogger on the user's computer, as well as several files and process watching daemons for monitoring the user activities: processes, application running, files opened, windows list, focus tracking, text typed, etc. The idea is to track at the operating system level the

activity of the user. This is a fairly complex solution with security issues, but which allows transparent data collection without having to alter user's software.

3 Data collection experiment

We have developed several tools for tracking the user activity at the higher level, the task. The idea was to see if by providing tools for helping the users to define a work context they would be able to use them efficiently and to improve their work. We developed a global graphical interface (shown in figure 1) composed of several modules which log the user activity while providing him a way to manipulate his computer environment. With our system, we are able to monitor the windows, applications, documents, emails, tasks, and their properties: size, content, coordinates, how often and how long they are used, etc.

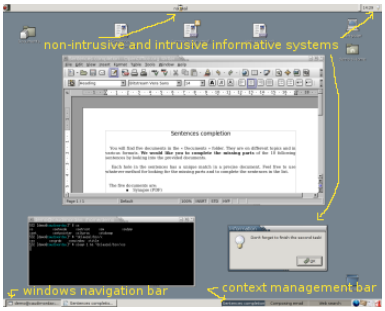

Figure 1: Prototype's interface

Instead of dealing with several windows and icons, the users create several groups of graphical elements called "tasks" which contain all the elements that are necessary to do a specific activity. Then by using the *task management tools*, they can organize their work in a global manner rather than having to handle individual application items.

The tools we have created allow the user to manage the environment by creating, deleting and switching among the existing tasks (collection of windows). Tasks are presented to the user as bars containing icons. The system collects data on the user's high level activity whilst allowing human or artificial observers to interact in a more or less conspicuous manners, e.g. by using animations, sounds and pop-up dialogues. As the first results look promising, we plan to improve our prototype by tracking user's low level actions, starting with functions activated within an application.

References

[1] Johnston, K. and Everling, S. [2006], Neural Activity in Monkey Prefrontal Cortex Is Modulated by Task Context and Behavioral Instruction during Delayed-Match-to-Sample and Conditional Prosaccade–Antisaccade Tasks, Journal of Cognitive Neuroscience **18**, 749–765. 2

[2] Lewis, C. and Rieman, J. [1994], Task-Centered User Interface Design, a Practical Introduction, HCI Bibliography : Human-Computer Interaction Resources. 2

[3] Robertson, G., Horvitz, E., Czerwinski, M., Baudisch, P., Hutchings, D., Meyers, B., Robbins, D. and Smith, G. [2004], Scalable Fabric: Flexible Task Management, in Proceedings of Advanced Visual Interfaces (AVI 2004), ACM Press, pp.85–89. 2

[4] Thompson, C. [2005], Meet the life hackers, the New York Times Magazine . 1