A Temporal Model for Reflective Multitasking

Steven Jeuris
The Pervasive Interaction Technology Laboratory
IT University of Copenhagen
sjeu@itu.dk

Steven Houben
The Pervasive Interaction Technology Laboratory
IT University of Copenhagen
shou@itu.dk

CHI'13, April 27 – May 2, 2013, Paris, France.
ACM 978-1-XXXX-XXXX-X/XX/XX.

Abstract
Knowledge work consists of complex planning, archiving and multitasking processes. Despite the widespread proliferation of task management tools, they often fail to embrace the entire workflow of knowledge workers, separating the means to define and execute goals. In this paper we introduce a temporal model for multitasking to guide the design of tools which support knowledge workers fully in their daily work. The model does not only focus on supporting ongoing tasks but takes into account the entire lifecycle of them, including the emergence, evolution, and storage for later recovery and re-use. We explore the application of the model by using it to reflect on the design of two task-centric desktop interfaces targeted at supporting knowledge workers.

Author Keywords
Task-Centric, Knowledge work, Temporal model

Introduction
It is well established that people organize their work in terms of higher-level thematically connected units of work [1], often referred to as *tasks*. *Multitasking* – working in parallel on several distinct tasks and frequently switching between them – is a common practice for knowledge workers. Keeping track of task contexts can be mentally demanding, resulting in *information overload*.

Additionally, interruptions that emerge from collaborative work often give rise to new tasks or task switches [4]. In order to address this, several approaches have been proposed in which the desktop environment can be reorganized, alleviating some of the problems arising from cluttered workspaces and interruptions.

Information which isn't immediately recognized as being important for future tasks is often retained, just to be sure no work is lost. *Archiving* this information is a manual process in which resources are structured as users see fit. These structures often reflect the actual tasks users performed. Knowledge workers use a range of strategies for *task management* including using calendars, to-do notes or other mundane tools. In any case, planned tasks at some point result in actual work sessions. When those tasks start, users need to manually construct the appropriate context. Tools that do support multitasking mostly ignore these equally important processes of archiving and planning work.

We see an opportunity to bridge the gap between multitasking, interruption management, archiving and task management, which are all dimensions of the same problem: *reflective multitasking*. In this paper we introduce a temporal model for multitasking and discuss how to apply this to system-design to support reflective knowledge work.

Temporal Model for Multitasking

Our suggested temporal model of knowledge work (Figure 1) consist of four elementary practices: multitasking, interruption handling, archiving and planning. These practices, done by knowledge workers, are connected through two fundamental processes: reflection and projection.

Multitasking
Multitasking is a common and widely spread practice which involves managing a large number of windows, files and other resources that are associated with the different task contexts [1]. Tools should support this by allowing the user to easily aggregate the related resources into meaningful structures reflecting the ongoing parallel tasks, as well as providing support to easily switch between them, restoring the appropriate task context when doing so [5]. This minimizes the reconfiguration work which the user otherwise has to do manually.

Interruption handling
Modern knowledge work is a highly collaborative process that requires coordination and communication with other people. Maintaining a general awareness of the progress of other people's work can play an important role in deciding what tasks to perform. Interruptions thus influence this synchronization process as they are used to interfere in the work progress of another user to align tasks. Interruptions often lead to new tasks or revisitation of old ones [4]. By providing a mechanism for the system to handle them, along with their context, interruptions can be better associated with new tasks or channeled to the appropriate task [3].

Archiving
Different archiving practices for long-term document storage have been observed; studies have shown that these practices can be reduced to three major approaches of personal information management (PIM): a piling, filing and structuring strategy [2].

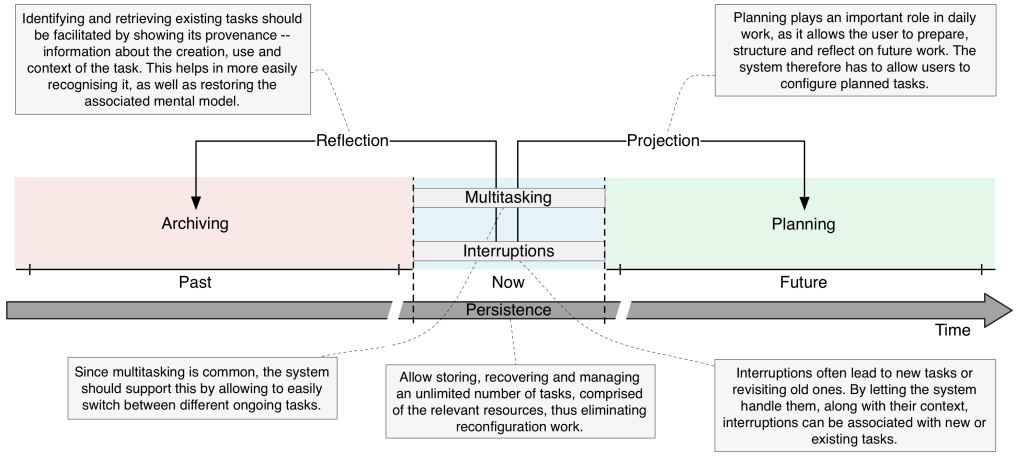

Figure 1: Modern knowledge work consist of archiving, multitasking, interruption handling and planning. Much activity work (multitask and interruption handling) however is heavily embedded in reflecting back on earlier (archived work) or future (planned) work.

Many knowledge workers use the desktop interface to pile files, as it allows them to spatially organize information. The more structured practice, in which files are moved into a folder hierarchy, is the result of limitations of current desktop interfaces. Tools should allow multitasking in a way that the spatially organized desktop piles, as well as the window configurations can be archived as is, without forcing users to hierarchize files or close/open windows (*persistence*). This will afford users to go back and pick up on completed tasks without the need to reconstruct the conceptual model. Identifying and retrieving (*reflection*) an existing task should be facilitated by showing its provenance –information about the creation, use and context of the task.

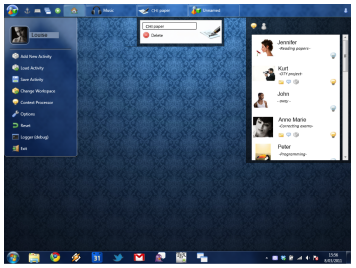

Figure 2: co-Activity Manager

Planning

Planning plays and important role in daily work, as it allows the user to prepare, structure and reflect on future work. The widespread adoption of calendars, to-do list applications and work management tools indicate their value in task management. These tools however, are intrinsically disconnected from the resources that are used to actually perform the work. Although resources such as files and applications play an important role in executing these tasks, they cannot be made part of the conceptual representation when planning them. This leads to a disconnection between the goals and means to execute them. Tools should thus allow users to create plans (*projection*) that can hold contextual information related to the execution. By including resources such as files and windows that are relevant for a future task, they become more connected to ongoing tasks.

Using the Model

To reflect the *task-centric* work of knowledge workers, some systems enhance the desktop interface by providing the user with *dedicated workspaces* in which they can organize their work. co-Activity Manager (cAM) [3] assigns a *virtual desktop* to each task and enhances the traditional interface by introducing collaborative features as shown in Figure 2. Switching between different desktops is possible using a bar which represents the ongoing tasks, in addition to the Windows taskbar which only shows those windows open within the active task. Evaluation of the system indicated that tasks were sometimes created in order to remind the user of future work, making the bar besides an overview of tasks also resemble a to-do list. The context of the open set of tasks can be saved, but it is up to the user to archive them in the traditional file system. These observations correspond to some of the identified practices in our temporal model.

Laevo builds on top of this task-centric approach but envisions to fulfill all dimensions of our temporal model. Similar to cAM, *multitasking* is supported by enabling the user to create a dedicated workspace per task, and allow for easy switching between them. As work progresses, tasks are visualized on a timeline, along with incoming *interruptions*, e.g. originating from collaborators. Figure 3 shows a set of workspaces outlined vertically which can overlap in time, indicating parallel tasks. When closing a workspace it isn't removed from the timeline, but rather *archived* so it can be restored at a later moment in time.

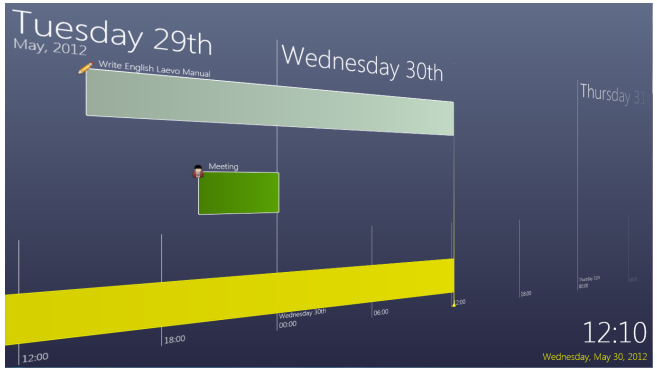

Figure 3: Laevo

The final practice from our temporal model, *planning*, is supported by enabling the user to preassign workspaces for future tasks. They can either be assigned to a self-introduced interruption in time, resembling a to-do item, or a time slot can be allocated for them on the timeline. The temporal representation of tasks empowers users to better understand the evolution of past tasks (*reflection*) and have an overview of upcoming tasks (*projection*).

Conclusion

In this paper we presented a temporal model for multitasking to guide the design of tools that support knowledge workers fully in their tasks. The model not only focuses on supporting ongoing work but takes into account the entire lifecycle of tasks, including their emergence, evolution, and storage for later recovery. In this workshop we want to introduce our identified set of requirements associated with the model, reflect on our experiences when introducing them into a task-centric tool, and discuss the challenges encountered along the way.

References

[1] González, V., and Mark, G. Constant, constant, multi-tasking craziness: managing multiple working spheres. In *Proc. of CHI 2004*, 113–120.

[2] Henderson, S. Personal document management strategies. In *Proceedings of the 10th International Conference NZ Chapter of the ACM's Special Interest Group on Human-Computer Interaction*, CHINZ '09, ACM (New York, NY, USA, 2009), 69–76.

[3] Houben, S., Vermeulen, J., Luyten, K., and Coninx, K. Co-activity manager: integrating activity-based collaboration into the desktop interface. In *Proc. AVI 2012*, 398–401.

[4] Jin, J., and Dabbish, L. Self-interruption on the computer: a typology of discretionary task interleaving. In *Proc. of CHI 2009*, 1799–1808.

[5] Voida, S., Mynatt, E., and Edwards, W. Re-framing the desktop interface around the activities of knowledge work. In *Proceedings of the 21st annual ACM symposium on User interface software and technology*, ACM (2008), 211–220.