Laevo: Calendar-Based Window Management

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

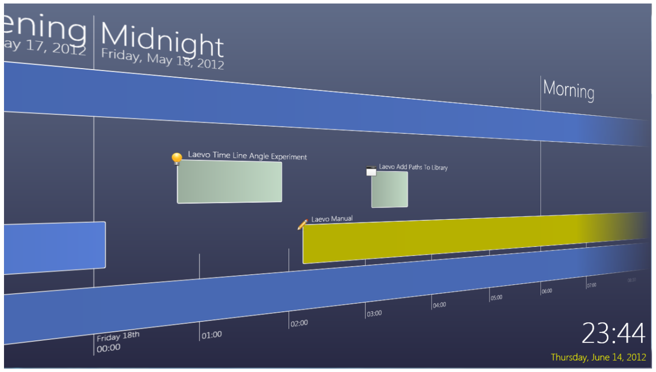

Figure 1: The timeline visualizing workspaces.

CHI'13, April 27 – May 2, 2013, Paris, France.
ACM 978-1-4503-1952-2/13/04.

Abstract

Window managers primarily focus on handling an increasing number of windows by allowing them to be grouped together ad hoc. Most existing approaches however provide little or no support for long-term persistent configurations of windows, thereby imposing unnecessary reconfiguration work on users. When old tasks are revisited, the associated window configuration needs to be manually restored, which could be prevented by allowing users to store any workspace they might have use of at a later time. This requires a truly scalable window management system capable of persisting, restoring and managing window configurations. In this paper, we introduce Laevo, a system that allows the grouping of windows within dedicated workspaces which are managed on a timeline, resembling a calendar.

Author Keywords

Window management, task management, workspaces, multitasking

ACM Classification Keywords

H.5.2 [Information interfaces and presentation (e.g., HCI)]: User Interfaces – Windowing systems.

Introduction

A window manager's task is to control the placement and appearance of windows. Since multitasking has become a way of living for knowledge workers [4] and increased screen space results in users keeping more windows open [10], there is an ever increasing number of windows which need to be handled. Traditional window managers fail in doing so, resulting in cluttered workspaces when opening many documents. E.g. Windows 95's taskbar would quickly overflow when browsing the Internet, since every separate web page had its own window. Therefore modern window managers usually provide some form of functionality to address window clutter, but no approach has been proven to be the most effective.

Studies have shown how users engage in multiple higher level activities oriented towards a certain goal [4]. Grouping windows together allows expressing this, or any other logical relation, between windows. Past research has repeatedly shown positive results when offering this functionality in user studies, e.g. [2, 10]. Following this approach we developed Laevo, a window manager which allows grouping windows together in dedicated workspaces, namely virtual desktops. We argue a user should not only be able to create ad hoc window configurations, but also needs to be able to persist them in order to prevent having to redo the configuration work which went into setting them up. This is worthwhile since retrieving the relevant resources for a certain working context can be a time consuming process [4]. For workspaces revisited after a long period of time, presenting window configurations exactly as they were last set up can help in rebuilding the mental model associated with them. In order to allow for long-term storage of these configurations we propose a truly scalable window management methodology, capable of managing an arbitrarily large number of windows. This is achieved by the temporal representation of window configurations on a timeline in a system resembling a calendar. (See Figure 1)

Window management

In an effort to provide additional functionality to organize windows, several window management techniques have been introduced, of which we will give an overview here.

Application grouping is used in Windows XP up to Windows 8 to group windows from the same application together under one item on the taskbar. This can reduce taskbar clutter, but imposes one extra indirection since the desired window can be hidden behind the application group. A *virtual desktop* is the extension of the physically available screen space through the use of software. Some virtual desktop managers allow you to work on not one, but several different desktop environments, each of which are *dedicated workspaces* only showing the windows open on that particular desktop. The user needs to switch to a desktop in order to work on it. The Rooms system [5] was one of the earliest systems to provide such functionality. Virtual desktops have since become common in many operating systems (OS's), either by default or as optional extensions. Adding functionality by which users can organize their windows *spatially* allows for increased ways in which to express relations between windows, leveraging human spatial memory. One such example is Scalable Fabric [10], in which windows shrink in size when moving them into the periphery, allowing to organize them in labeled groups. Switching between groups is supported with a single mouse click. Organization doesn't need to be restricted to 2D space, e.g. BumpTop [1] allows dragging and tossing around objects in 3D space, taking the desktop metaphor even further.

Task management

We define *tasks* as actions which are performed in order to obtain a certain goal. In a desktop environment, one task is usually associated with several applications and documents. This has lead to several approaches by which tasks can be represented more explicitly in order to benefit the user.

Applications which are used for many tasks can become candidates for managing them, *integrating task management into the application*. Taskmaster [3] does so by embedding task-centric resources directly into the email client. Mylyn [6] is an extension for the Eclipse programming environment which allows switching between programming tasks and automatically builds up their context. Recognizing that knowledge workers organize their work into tasks, some systems are *fully dedicated to managing them*. Activity-Based Computing (ABC) [2] reframes the existing desktop metaphor around the higher-level construct of *activity*, focusing not only on desktop work, but the ability to roam and share activities. ABC posits activity should be the basic computational unit, instead of the file or application. TAGtivity [8] allows *tagging* resources within a PC environment. While tags don't necessarily need to represent tasks, they were shown to be used in such a way, allowing to maintain multiple working contexts.

Integrating task and window management

Window configurations often reflect the tasks users perform. A window manager which allows users to group windows together supports expressing these higher-level tasks, and distinguishing between them. Although not all window configurations represent tasks, we see an opportunity to fully integrate task management with window management, while still allowing for sufficient flexibility to manage any ad hoc window configurations.

A calendar is an often used tool to plan tasks. We repurpose the calendar metaphor to not only plan tasks, but also to provide an overview of ongoing tasks, allow switching between them, and reflecting on past tasks. The calendar effectively becomes a window manager capable of storing a large number of window configurations in time. Where previous research was focused on either window or task management, to our knowledge this is the first attempt to integrate the two.

Within information management research, very similar time-based systems have been introduced. In TimeScape [9] users can spatially arrange information on the desktop. Time travel is supported to go to past or future states of the desktop, allowing for a time-based way of archiving documents. TimeSpace [7] presents the user with a timeline on which personal information spaces with containing documents are shown. However, none of these are concerned with window management.

Laevo

We identified the following conceptual requirements which on top of aiding the user in creating traditional ad-hoc window configurations, also allows storing and efficiently recovering them over longer periods of time.

Persistence Allow storing, recovering and managing an unlimited number of workspaces, comprised of the relevant windows and resources for a certain working context, thus eliminating reconfiguration work.

Reflection Identifying and retrieving an existing workspace should be facilitated by showing its provenance –information about the creation, use and context of the workspace. This helps in more easily recognizing it, as well as restoring the associated mental model.

Multitasking Since it's common for knowledge workers to multitask [4], the system should support this by allowing to easily switch between different workspaces.

Projection Task management needs to be supported since we envision integrating it with window management. This is possible by allowing to preassign workspaces to planned tasks, as well as to-do items.

Interruption handling Interruptions often lead to new tasks or revisiting old ones [3]. By letting the system handle them, along with their context, interruptions can be associated with new or existing workspaces.

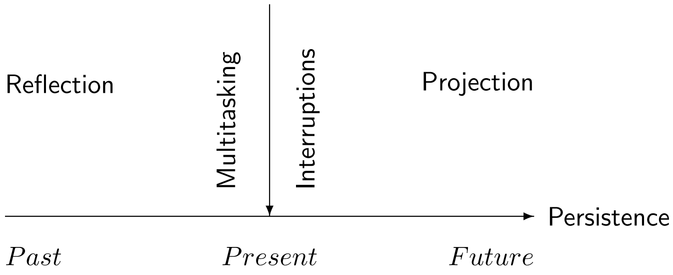

Figure 2: Conceptual requirements overview

By representing persistent workspaces and interruptions in time, Laevo attempts to fulfill these requirements. A conceptual overview of how this is reflected on a timeline is presented in Figure 2. The right side of the time line represents the future, where projection can take place. The left side represents the past, where an overview of past workspaces is shown.

User Interface

Since we want to clearly differentiate the meta-work of task management from performing actual work, we opted for a full-screen visualization of a timeline on which the workspaces are managed. (See Figure 1) To strengthen the notion of time a perspective transformation is used, following the conceptual metaphor of expressing time as a path in space. This saves screen estate without seriously hampering readability. Providing an overview of different work contexts over time can be a powerful cue for recall by invoking episodic memory; it is visible which workspaces were active concurrently, how long they were active, and in which order work was performed in them. The granularity of information to be shown can vary drastically; some lasting weeks, others mere minutes. In order to address this, using two simple manipulations – zooming and panning – the user is able to set up any visible time span which is most appropriate. There are no predefined zoom levels; any intermediate zoom level is possible, offering one continuous experience.

Figure 3 visualizes how the timeline along with one open workspace changes while moving from an hourly perspective which offers a complete overview of the history of the workspace, to a daily perspective which allows seeing what is coming up in the future. Labels change whenever a given perspective becomes the most dominant one, cycling between hours, part of days, days, weeks, months and years. The current perspective can always be determined by the labels in the top left corner. While panning the timeline, the earliest top label sticks to the screen side until it is pushed off by the next one. (See Figure 1)

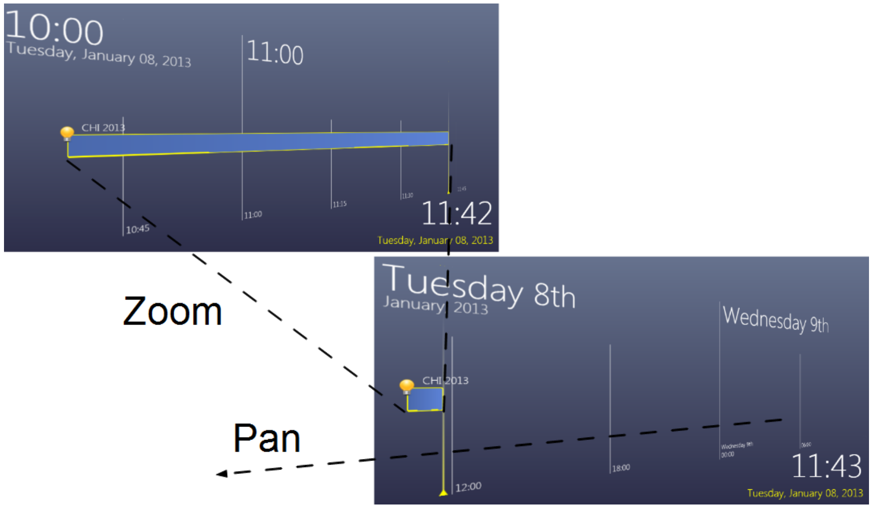

Figure 3: Zoom and pan transformation

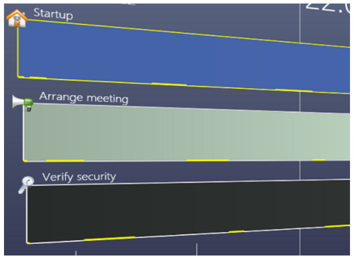

Figure 4: Workspace representations

Workspaces are represented by rectangular areas which stretch along the timeline horizontally, indicating when the working context was active. Only one workspace can be open at any given time, but several can be active simultaneously, allowing for multitasking. Figure 4 shows three parallel used workspaces which can be laid out vertically on the timeline. Yellow lines along the bottom of a workspace indicate when it was open. When opening a workspace by clicking on it, we only want the relevant activity context to be visible in order to reduce information overload. To this end we opted to use virtual desktops; they provide users with a familiar but dedicated working environment which still supports all the existing tools they are used to. Only when opening up the timeline using a shortcut key are users confronted with the new environment. Closing workspaces results in them becoming inactive, no longer being part of the present multitasking session. They remain visible on the timeline so they can be revisited at a later time.

Following the redundancy gain principle, a workspace is represented by several distinct visual cues so it is more likely to be recognized quickly. A name can be provided which can either be edited in-place on the timeline, or in a pop-up menu as shown in Figure 5, where additionally a color and icon for the rectangle can be chosen.

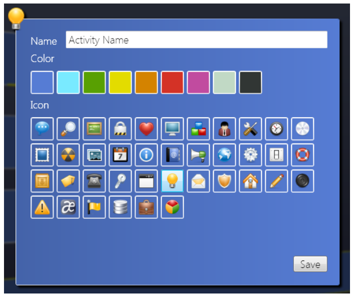

Figure 5: Edit workspace representation

A working context is ever changing; a clear distinction between different contexts can't always be drawn upfront. Therefore the system provides a flexible mechanism by which windows can easily be moved and shared between different workspaces. Similarly to how files are cut, copied and pasted, shortcut keys are provided to perform these operations on windows once they have focus. Cut windows disappear immediately and are reassigned to the workspace in which they are pasted. Several cut and copy operations can be performed in a row, placing them on a stack until a paste operation processes them all.

Evaluation

The system was evaluated by 12 knowledge workers during a full day in-field study. We were interested in general usability of the system, as well as how likely users are to adopt the suggested metaphor.

Using an all positively phrased version of the System Usability Scale (SUS) resulted in a mean score of 83.5, which is reflected by the positive feedback given during interviews. Users reported they liked the tidiness which dedicated workspaces offer, using descriptions like "clear", "organized", "breathing space" and "clean". It was recognized however that the real power lies in long term usage, which wasn't a possibility for many due to the prototype nature of the system. One user continued using the interface during several weeks, and reported having the feeling she was able to get more work done.

Future work

In this paper, we presented a new window management metaphor, incorporating task management. Initial feedback during user studies shows such a system has much promise, but steps need to be taken so it can be used over longer periods of time so its true potential can surface. Our current goal is to prepare the system for long-term field deployment, as well as continuing on the implementation of our conceptual requirements. This includes interruption handling as well as working towards full integration of task management so the system can replace existing calendar systems.

References

[1] Agarawala, A., and Balakrishnan, R. Keepin'it real: pushing the desktop metaphor with physics, piles and the pen. In *Proc. of CHI 2006*, 1283–1292.

[2] Bardram, J., Bunde-Pedersen, J., and Soegaard, M. Support for activity-based computing in a personal computing operating system. In *Proc. of CHI 2006*, 211–220.

[3] Bellotti, V., Ducheneaut, N., Howard, M., and Smith, I. Taking email to task: the design and evaluation of a task management centered email tool. In *Proc. of CHI 2003*, 345–352.

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

[5] Henderson Jr, D., and Card, S. Rooms: the use of multiple virtual workspaces to reduce space contention in a window-based graphical user interface. *ACM Transactions on Graphics (TOG) 5*, 3 (1986), 211–243.

[6] Kersten, M., and Murphy, G. Using task context to improve programmer productivity. In *Proc. of SIGSOFT 2006*, 1–11.

[7] Krishnan, A., and Jones, S. Timespace: activity-based temporal visualisation of personal information spaces. *Personal and Ubiquitous Computing 9*, 1 (2005), 46–65.

[8] Oleksik, G., Wilson, M., Tashman, C., Mendes Rodrigues, E., Kazai, G., Smyth, G., Milic-Frayling, N., and Jones, R. Lightweight tagging expands information and activity management practices. In *Proc. of CHI 2009*, 279–288.

[9] Rekimoto, J. Timescape: a time machine for the desktop environment. In *In extended abstracts CHI 1999*, 180–181.

[10] Robertson, G., Horvitz, E., Czerwinski, M., Baudisch, P., Hutchings, D., Meyers, B., Robbins, D., and Smith, G. Scalable fabric: flexible task management. In *Proc. of AVI 2004*, vol. 25, 85–89.