TimeZoom: A Flexible Detail and Context Timeline

Raimund Dachselt
Dresden University of Technology
Multimedia Technology Group
D-01062 Dresden, Germany
dachselt@acm.org

Markus Weiland
Dresden University of Technology
Multimedia Technology Group
D-01062 Dresden, Germany
mweiland@3danim.de

CHI 2006, April 22–27, 2006, Montréal, Québec, Canada.
ACM 1-59593-298-4/06/0004

Abstract
In this paper we present TimeZoom, an interactive timeline widget to be combined with a tabular display of data in calendar, e-mail, project planning, or other applications. Different time levels are vertically stacked and can be smoothly zoomed, permitting arbitrary granularity of time units. In addition, single or multiple focus regions with various levels of detail can be defined to allow the display and comparison of time-dependent data, while preserving the overall context.

Keywords
Information visualization, time-dependent data, detail and context technique, interaction technique, time navigation, fisheye view, e-mail client, calendar

ACM Classification Keywords
H.5.2. Information interfaces and presentation (e.g., HCI): User Interfaces – Graphical User Interfaces (GUI)

Introduction
As the primary attribute of e-mails [8], tasks and appointments, time plays a crucial role in desktop applications such as e-mail clients, calendars, and project planning software. The interaction with time-dependent data is also a dominant feature of numerous other applications, e.g. commercial media production tools for

time-variant media including video or sound editing as well as 3D animation tools. Therefore, powerful time navigation techniques are important in many domains where data is inherently coupled with time.

The granularity of time (i.e. time unit) required by a specific application can differ considerably. Take for example a calendar with a resolution ranging from 1 year down to 5 minutes or even PAL video editing, where 1 hour contains 90,000 frames (3600s*25 fps). This calls for both precise control and high-level overviews of time-variant data. Most traditional interactive time visualizations cope with this challenge by offering discrete views at different levels of detail (LOD). The multiple-view approach leads to several shortcomings:

- Views are discrete and fixed, thus they are bound to one specific LOD and have a static size.
- High-level views hide too much data, whereas detailed views suffer from lack of context and orientation.
- Scalability, e.g. for mobile devices is not supported.
- Missing support for zooming lenses, i.e. high-detail views or focus areas within a coarser time view.
- Requirement of additional cognitive efforts for reinterpretation and orientation [6,2] due to missing smooth transition between views.
- Display of absolute time is missing, i.e. scroll-bar based views (e.g. in e-mail applications) only show relative position within the collection.

The goal of the work presented in this paper is to develop a flexible timeline widget improving on these issues by offering selective high-detail techniques and an augmented level of context.

Related Work

In a photo album application for large photo collections Harada et al. [3] employ a vertical timeline as the primary navigational element. The visible timeline range initially spans a full year containing sets of photo albums. Users may arbitrarily refine the presented time span by selecting a section or choosing an album. In doing so, the timeline adjusts its scale to the best-suited granularity for the respective level of refinement. At the lowest level the presentation switches to a grid view for individual photo selection. Thus a consistent timeline view on all levels and animations for smooth transformation between levels are missing.

In the context of e-mail applications Sudarsky et al. [7] presented a horizontal timeline offering preset views at different granularities. E-mail data is visualized in an associated matrix. In contrast, Mandic et al. [4] utilize both screen dimensions for representing time at fixed and discrete levels of detail. Thus, with respect to time, users get a detailed as well as a high-level view of their e-mails. However, this is achieved at the cost of one screen dimension, which we wanted to keep free for additional data visualization.

With DateLens [1], the concept of fisheye views is applied to a calendaring application for mobile devices. It offers a typical view with weekdays aligned on the horizontal and weeks on the vertical axis. Individual days thus fill the resulting matrix which can be adjusted through a modified scrollbar widget. The view can be biased by the user towards individual days which is done in an animated fashion. Our approach is related to this work, though we avoid a fixed granularity of time in order to prevent overloading the display.

Mills et al. [5] propose a video editing timeline offering multiple LOD, permitting users to recursively confine the visualized time segment. As each refinement results in an additional timeline widget, multiple LOD are simultaneously exposed. As a consequence, the required screen estate grows proportionally, thus limiting the use on smaller displays.

Conception and design of TimeZoom

TimeZoom is realized as a horizontal timeline along one screen dimension. It supports displaying multiple levels of granularity as well as regions of focus from a single unified and scrollable view. TimeZoom by itself does not visualize data, as its sole purpose is to provide a flexible spatial reference system for displaying time-related data. However, we conceived the widget with a novel e-mail client in mind. Depending on the application domain, arbitrary tabular data visualizations (such as matrices) can be combined with TimeZoom as long as columns represent time units. This is sketched in figure 1 by the vertical lines above the timeline.

Basic design and timeline levels

Time is measured in hierarchically contained units such as hours, days, and weeks. TimeZoom visualizes this as a set of levels representing a time unit. They are rendered as flat rectangles of different heights in a vertical stack, starting with the coarsest time unit at the bottom up to the finest unit on top. To ensure application domain independence, the particular number of levels and their unit type is not predefined. As an example, figure 1 depicts parts of TimeZoom for a typical calendaring application with five levels containing years, months, weeks, days, and hours. Different background colors for each level facilitate orientation and navigation.

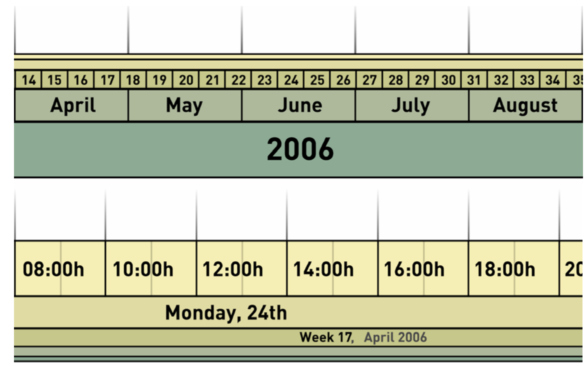

Figure 1. Cut-outs of TimeZoom for a calendar application showing two different zoom levels focusing on years and hours.

Each time level is horizontally subdivided into several cells depending on its associated unit type. As an example, the month level contains twelve cells labeled *January* to *December*. When visualizing such large spans of time, it becomes apparent that equally treated level heights will not be feasible on normal computer screens. TimeZoom therefore introduces the notion of weighted level heights or a *vertical fisheye* effect. To achieve a consistent and reasonable height of the whole widget and display all levels at the same time, the vertical extent of each level is modified while changing the currently focused time unit. Take for example figure 1, where *years* are focused at the top and *hours* at the bottom. Assuming the focused or dominant level has a normalized height of 1, the size of neighboring levels is recursively decreased according to a specific zoom factor. In the first design it is set to 1/1.7, leading to normalized level heights of 1, 0.59, 0.35, 0.2 etc. Since TimeZoom supports non-discrete zooming, height values are interpolated. Increase and decrease of all levels is smoothly animated while interacting with the widget.

Note, that cell subdivision disappears when the height of a level falls below a certain threshold. This can be seen at the top of figure 1, where *day* and *hour* subdivisions were automatically omitted.

This flexible design of TimeZoom has several implications on the labeling of cells. Applying the vertical fish-eye effect comes at the cost of resulting cell sizes being insufficient for full labels. Depending on the time unit, there are various abbreviations conceivable. Take for example a day level cell labeled *Monday, 24th*, which could be reduced to *24th*, *24* and even *M* when size is decreased. Should abbreviations not be possible or insufficient, cells of a level could be merged by a suitable algorithm as can be seen with the even hours at the bottom of figure 1. For the case that cell height is not sufficient for any label at all, TimeZoom establishes the concept of label inheritance. With it, labels of coarser grained levels pass their information down to the next finer grained level. This information is added there to provide as much contextual information as possible. See for example at the bottom of figure 1 the appended *April 2006* for *Week 17*. Finally, horizontal scrolling could easily move centered cell labels off-screen. An algorithm therefore aims at retaining labels on-screen as long as sufficient cell space is visible.

Regions of focus

Time Zoom allows the definition of *regions of focus*. A region is a confined range of time with a zoom level independent of the overall timeline. As such, users are able to simultaneously obtain coarser or finer grained views with multiple areas of interest. Regions can be defined and selected, indicated by colored selection bars (compare figure 3). Selected regions can be modified in terms of width (i.e. time span) and zoomed level of time detail. In addition, they can be horizontally moved along the timeline and can be deleted. Figure 2 displays two such regions, each exposing a different zoom level of detail. It quickly becomes apparent that altering a region's zoom results in a break of continuity with respect to the unmodified surrounding timeline. TimeZoom thus offers a transitional zone which serves as a connector between a region and its surroundings. Figure 3 illustrates the three zones relevant for focus regions. Zone **a** represents the region itself, zone **b** indicates the transitional zone, while zone **c** stands for neighboring timeline segments. Depending on the application domain, different settings for the extent of zone **b** are conceivable. As an example, a width of zero would constitute a hard cut like in figure 2. By furthermore altering the interpolation function applied to the transition zone, different lens types can be simulated. Two examples are depicted in figure 4.

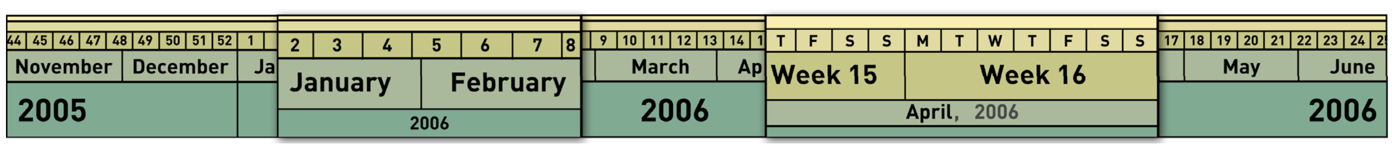

Figure 2. TimeZoom with two additional focus regions displaying different higher levels of time detail.

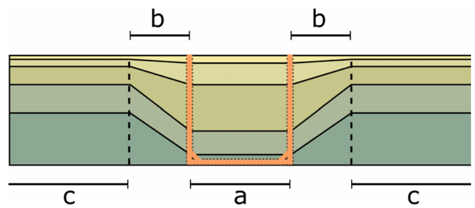

Figure 3. Definition of the different zones for focus regions.

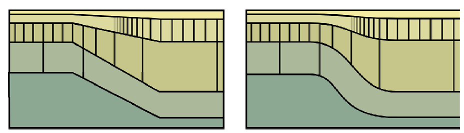

Figure 4. Two different styles for the transitional zone **b**.

Interacting with TimeZoom
Interaction techniques for this timeline depend on a number of variables, such as application domain, input and display device, or timeline focus parameters. For this reason the best choice of settings will vary from case to case and actually needs to be investigated for a concrete scenario. However, some basic interaction ideas and interface elements are presented which result from work with our initial Flash-based prototype. Though it supports all basic interactions and the definition and modification of focus regions, not all interaction concepts are implemented yet. See figure 5 for parts of the user interfaces.

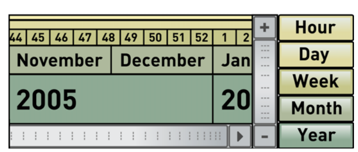

Figure 5. Parts of the prototypical TimeZoom user interface.

With TimeZoom, *horizontal scrolling* can be done by directly dragging the timeline or a focus region to the desired position. As an alternative the thumbwheel underneath can be used like an unlimited scrollbar. Discrete movements in intervals of the currently dominant time level can be achieved by clicking on arrow buttons on both sides. This allows stepping through days, weeks, months etc. Transformations caused by thumbwheel and iterative scrolling are always applied to the entire timeline including any existing focus regions.

Zooming of the whole timeline or the selected region, i.e. changing the dominant time level or vertical fisheye focus, can be done by different means. This can be either accomplished by activating the time buttons on the right (see figure 5) to directly set the desired time level or by continuous movements using the thumbwheel located to the right of the timeline or alternatively rotating a vertical mouse wheel. The additional +/- buttons offer small-scale incremental zooms.

TimeZoom also offers means for *creating, deleting, selecting, and modifying regions of focus*. New regions can be created by double-clicking at the desired position or issuing an 'up'-movement mouse gesture. New regions are automatically selected, which is indicated by a surrounding colored widget (see figure 3). Other existing regions can be selected by double-clicking inside that region. With the help of the selection widget users are able to shift regions and to expand or shrink their size (i.e. time span) using the left and right borders. Thereby the zoom level of the adjacent timeline needs to be adjusted to conserve the absolute amount of time visible on screen. Finally, regions of focus can be deleted by double-clicking the bottom bar of a selected region, by issuing a 'down' mouse gesture, or by shrinking the region's size to a value close to zero.

Discussion and future work

We have presented the design of TimeZoom, a flexible hierarchical timeline allowing multiple regions of interest. Our experiences with the first prototypes as well as informal user feedback are encouraging. Major benefits are the single, unified and consistent view on various time levels as well as the idea of flexible focus regions with different levels of detail being smoothly integrated into the same view. In both cases a vertical and horizontal detail and context view is offered to the user, thus reducing cognitive workload.

TimeZoom was developed to be integrated into user interfaces concerned with time-dependent data. It should be combined with columns containing data items or even a complete matrix above the widget. Columns for each time unit may contain appointments, tasks, e-mails, group members, video frames or audio samples etc. In case of a matrix, the rows can for example be used to represent contacts for e-mails, categories of different tasks, or audio and video tracks. Several applications could benefit from this widget, among them task management, e-mail, or calendar applications as well as media editing or 3D animation packages.

In order to justify this statement and to evaluate the applicability in various contexts we are planning to carry out future improvements in two different steps. First we need to optimize the interaction with TimeZoom according to users' feedback in an evaluation. To give an example we will experiment with different types of transition zones and interaction devices in our prototype. In a second step we would like to combine TimeZoom with a calendar or e-mail application in order to test its usability with real data in a challenging application context. This will reveal further challenges such as the appropriate layout of data in arbitrary time granularities. For future work we are also considering implementation of this timeline on mobile and other devices to evaluate its scalability.

References

[1] Bederson, B. B., Clamage, A., Czerwinski, M. P., and Robertson, G. G. DateLens: A fisheye calendar interface for PDAs. In *Trans. Computer-Human Interaction 11*, 1 (2004), 90-119.

[2] Chang, B. and Ungar, D. Animation: from cartoons to the user interface. In *Proc. UIST 1993*, ACM Press (1993), 45-55.

[3] Harada, S., Naaman, M., Song, Y. J., Wang, Q., and Paepcke, A. Lost in memories: interacting with photo collections on PDAs. In *Proc. JCDL 2004*, ACM Press (2004), 325-333.

[4] Mandic, M. and Kerne, A. Using intimacy, chronology and zooming to visualize rhythms in email experience. In *Ext. Abstracts CHI 2005*, ACM Press (2005), 1617-1620.

[5] Mills, M., Cohen, J., and Wong, Y. Y. A magnifier tool for video data. In *Proc. CHI 1992*. ACM Press (1992), 93-98.

[6] Robertson, G., Card, S. K., and Mackinlay, J. D. The cognitive coprocessor architecture for interactive user interfaces. In *Proc. UIST 1989*, ACM Press (1989), 10-18.

[7] Sudarsky S., and Hjelsvold R. Visualizing Electronic Mail. In *Proc. InfoVis (IV) 2002*. IEEE (2002), 3-9.

[8] Yiu, K., Baecker, R.M., Silver, N., and Long, B. A Time-based Interface for Electronic Mail and Task Management. In *Proc. of HCI International 1997*, 2 Elsevier (1997), 19-22.