DateLens: A Fisheye Calendar Interface for PDAs

BENJAMIN B. BEDERSON and AARON CLAMAGE
University of Maryland
and
MARY P. CZERWINSKI and GEORGE G. ROBERTSON
Microsoft Research

Calendar applications for small handheld devices are growing in popularity. This led us to develop DateLens, a novel calendar interface for PDAs designed to support complex tasks. It uses a fisheye representation coupled with compact overviews to give the big picture in a small space. The interface also gives users control over the visible time period, as well as supporting integrated search to discover patterns and outliers. Designed with device scalability in mind, DateLens currently runs on desktop computers as well as PDAs. Two user studies were conducted to examine the viability of DateLens as a replacement for traditional calendar visualizations. In the first study, non-PDA users performed complex tasks significantly faster with DateLens than with the Microsoft Pocket PC 2002™ calendar (using a PDA emulator). In addition, they rated DateLens as being easier to use than the default calendar application for a majority of the tasks. In the second study, the participants were expert Pocket PC users and the software was run on their own devices. Again, DateLens performed significantly faster for the complex tasks, and there were satisfaction differences favoring each calendar for different kinds of tasks. From these studies, it is clear that DateLens is superior for more complex tasks such as those associated with longer time periods. For daily event tracking, users familiar with the default Pocket PC calendar strongly preferred its daily view and behaviors.

Categories and Subject Descriptors: H.5.2 [**Information Interfaces and Presentation**]: User Interfaces—*Graphical user interfaces (GUI), Interaction styles, screen design*; I.3.6 [**Computer Graphics**]: Methodology and Techniques—*Interaction techniques*

General Terms: Design, Experimentation, Human Factors

Additional Key Words and Phrases: Fisheye distortion interfaces, information visualization, calendar interfaces, PDAs, animation, graphics

The portion of this work performed at the University of Maryland was funded in part by a generous gift from Microsoft Research.

Authors' addresses: B. B. Bederson and A. Clamage, Human-Computer Interaction Laboratory, Computer Science Department, Institute for Advanced Computer Studies, University of Maryland, College Park, MD 20742; email: {bedrson,aclamage}@cs.umd.edu; M. P. Czerwinski and G. G. Robertson, Mircrosoft Research, one Mircrosoft Way, Redmond, WA 98052; email: {marycz,ggr}@microsoft.com.

1. INTRODUCTION

More and more people carry small Personal Digital Assistants (PDAs) with them to help manage day-to-day information. While these devices can be helpful for retrieving relevant information when it is needed, our informal polling of colleagues tells us that they are less helpful for more complex tasks, such as those involving larger time spans. This is not surprising since these devices have limited screen space, forcing users to jump around through multiple screens, making it harder to relate disparate pieces of information.

We designed DateLens,[1] a new calendar interface for PDAs, to better support these more complex tasks such as picking a good weekend to go camping, counting the number of Mondays in November, finding the start and end dates of a trip. These are instances of tasks that we classify more generally as scheduling, navigating and counting, and searching, respectively.

Our secondary goal was to design a calendar interface that would scale down to smaller devices such as mobile phones, and up to larger devices such as desktop displays. This is important since individuals are likely to access their calendar information from these and other devices. Offering a single interface across devices would give users a consistent experience, and, eventually, the ability to more readily switch between devices using whichever one is more accessible.

The DateLens design addresses these goals by using a fisheye distortion technique coupled with carefully designed visualizations and interactions appropriate for a pen-based device and a small display [Bederson et al. 2003]. Figure 1 shows DateLens running on a Pocket PC device.

The basic approach used by DateLens is to start with an overview of a large time period using a graphical representation of each day's activities. Tapping on any day expands the area representing that day, and reveals the list of appointments in context. Users may change focus days, zoom in further for a full day view, search for appointments, and reconfigure the viewable space.

This interface shows varying time span displays within the same framework using animated transitions between view changes, and thus, may improve users' ability to maintain a sense of where they are. This paper describes the interface along with the results of two user studies comparing DateLens to the standard Microsoft Pocket PC 2002™ calendar interface. Evidence from these studies supports our hypothesis and provides many useful ideas for improving the DateLens user interface design.

1.1 Related Work

Fisheye distortion techniques, initially called bifocal displays, were introduced by Spence and Apperly [1982] 20 years ago. At that time, the basic concept was to distort the information space so focus items were enlarged while peripheral items were shrunk. A few years later, Furnas [1986] generalized this approach by suggesting a "degree-of-interest" function, which calculates the relevance of

[1]DateLens is available for download at www.cs.umd.edu/hcil/datelens.

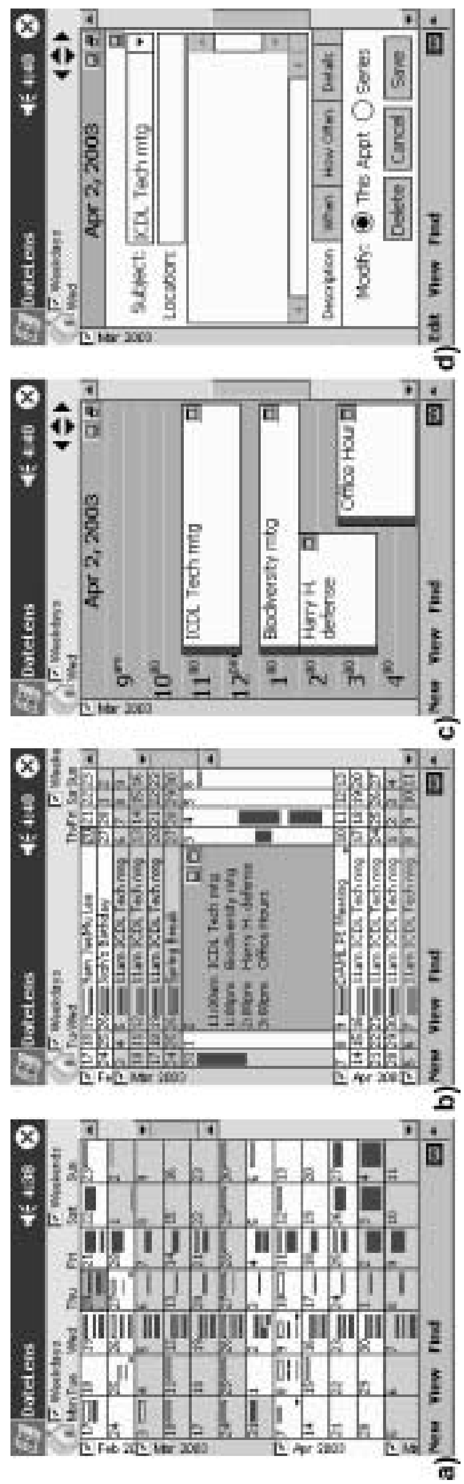

Fig. 1. The DateLens interface with the view configured to show 12 weeks and consecutive levels of detail as the view is: a) shown as an overview; b) zoomed into one day; c) focused on that day; and d) zoomed into an appointment. All transitions between views are animated.

each item in the information space. This result is then used to calculate the size and visibility of that item.

Fisheye distortion techniques have been applied to a number of domains, from graphs [Sarkar and Brown 1992] to trees [Lamping et al. 1995] to menus [Bederson 2000], among others. Their effectiveness has been mixed, but in at least some cases, such as hierarchically clustered networks .[Schaffer et al. 1997], fisheye interfaces have been shown to benefit users. The common theme has been that fisheye views are appropriate when users need to see details of some specific items in the context of a large information space.

The idea of using fisheye distortion to view calendars is not new. It was first suggested over ten years ago by Furnas [1991], where he described a textual Lisp-based calendar program. We followed the basic approach Furnas created at that time. A tabular display shows days in the calendar, and clicking on individual days causes the amount of space allocated to that day to be increased. Furnas' calendar used varying amounts of space to show different days, so that the focus day was largest, and other days were sized in inverse proportion to the distance from the focus day (although days in the past were always tiny because the assumption was that users were more interested in the future.) This program, while impressive for its time, did not support graphical representations of appointments, animation, searching, or full screen views, and did not have an interface to control which and how many weeks to display. It was not designed with small displays in mind. In addition, it was not evaluated with users, and was not pursued past the publication of the above-mentioned technical report.

Fisheye visualizations have also been used successfully to view and interact with generic tabular information. The best known example of this is probably Table Lens, which presents an interface for tabular data [Rao and Card 1994]. This visualization approach was designed for tables with many rows, but a modest number of columns. It represents each cell with a horizontal bar whose length is proportional to the value of the cell for numerical data, and whose position represents categorical data. The height of each row is scaled to fit the available space. Users may then focus on individual or multiple cells (or rows or columns). In addition, users can sort rows to help see relationships within the data. While this approach is somewhat similar to the present work in that it uses a fisheye distortion to view tabular data, it is not directly applicable to calendar information as it is designed for spreadsheet style information that has one item per cell, rather than the multiple and possibly conflicting appointments of calendars. In addition, it does not support the searching or navigation that calendar users require. Nevertheless, the acceptance of this technique (as demonstrated by its successful commercialization [Inxight 2003]) gives hope that users will be able to understand and navigate calendar information in a tabular format using a fisheye view.

Researchers have also developed other techniques to visualize and interact with calendar information. Plaisant and Shneiderman [1992] were among the first to develop small visual representations of calendar information. Mackinlay et al. [1994] developed a 3D "spiral calendar" visualization. This approach, while not suitable for small devices since it displays several visual representations simultaneously, does have a fisheye-like quality in that it displays detailed appointment information with visual links back to larger scale calendars. So, users can see what week an appointment comes from, what month that week is in, what year that month is in, and so forth.

Perhaps surprisingly, fisheye techniques have rarely been used for interfaces for PDAs and other devices with small displays. One use was by Björk et al. [1999] who used "flip zooming" to display web pages and then personal information including calendar data [Björk et al. 2000] on a PDA as demonstrated within their PowerView application. Flip zooming consists of presenting one medium-sized focus page and several tiny pages in the periphery that can be used for navigation. At a high level, the basic approach of flip zooming is similar to DateLens. However, DateLens differs from flip zooming in that flip zooming is designed to support hierarchical textual data while DateLens supports tabular data with a natural visual abstraction. Furthermore, flip zooming has system-defined viewpoints while DateLens allows users to define views. Finally, DateLens adds two important new features: integrated search and animated transitions.

2. DATELENS

DateLens is the fisheye calendar interface we designed for use primarily on a PDA (Figure 1). It was designed and built at the University of Maryland and Microsoft Research then joined the project to run the experiments. DateLens runs on both Pocket PC devices as well as on full-sized Windows machines.

As described in the related work section, DateLens draws its design from a range of earlier work. While the individual features of DateLens represent variations of existing approaches, our primary contribution is the integration of a host of techniques to create a novel application that is both usable and useful in an important domain. In addition, we present two studies that show how this design works in comparison to a more traditional calendar design for small devices. We hope that DateLens illustrates how existing techniques can be applied in new ways to new domains, and in doing so, advance the state of the art.

DateLens was built to target currently available devices running the Microsoft Pocket PC operating system. These devices are small enough to fit comfortably in a hand, have high quality 240×320 pixel screens, and fast enough processors to support modest animation.

Since DateLens was designed for a pen-based PDA, we have been careful to design the interaction so that it requires minimal text entry and simple interaction. The entire interface can be accessed with single taps, although dragging offers some modest extra features—including access to tool-tips and fast scrollbar usage.

The rest of this section describes the DateLens interface in detail, including a description of its navigation capabilities, the visualizations that represent calendar information at different sizes, and how search capabilities are integrated into the interface.

2.1 Navigation

A fundamental characteristic of DateLens is its ability to support users in easily customizing their view of the calendar. Most commercial calendar applications provide mechanisms to directly switch between day, week, month, and year views, and to change which range of dates are visible with each view. However, the different views are disconnected and inflexible. One goal of DateLens was to offer the same functionality in terms of a range of views, but to do so in an integrated and flexible fashion. Using animation and fisheye distortion, users can see the relationship between the range of dates they are viewing and the previous view. As such, users should not have to expend as much mental effort to manage context and figure out "where they are."

The basic organization of the display is tabular (Figure 1). Each row represents one week, with columns representing the days of the week. The number of visible rows can be changed from one, representing a single week, to 52, representing an entire year. When a day is tapped, the view is gradually changed to expand the day that was tapped on (Figure 1a, 1b). This and all view transitions are animated over a user-defined period that defaults to about 250 milliseconds. The animations are rendered by linearly interpolating the position of each grid line for the in-between frames. The result is that each view smoothly transitions from one view to the next.

The view can be changed through direct manipulation by interacting with the calendar itself, by manipulating widgets in the periphery of the display, or by using special hardware button shortcuts. One of the challenges was to make it extremely easy to configure the view. The final design only uses interaction mechanisms that most users are familiar with, including tapping on an item that they want more information about, and manipulating familiar buttons and widgets.

Direct Manipulation. DateLens was designed to take advantage of user familiarity with clicking on hyperlinks to find more detailed information about the item they clicked on. It allows users to tap anywhere on a day to focus on that day, minimizing other days (Figure 1a).

Within a focused day (Figure 1b), users can tap on the background, or tap on the maximize button to zoom in to a full day view. Or, users can tap on the minimize button to go back to original view with no days focused.

Within the full day view (Figure 1c), users can tap on an appointment's background or the appointment's maximize button to view the appointment details. Tapping on the day's minimize button returns to the original view, and tapping on the overlapping-windows button returns to the focus day view.

Within the full appointment view (Figure 1d), scrolling shows the full contents of the appointment. Tapping on the minimize button returns to the full day view.

2.1.1 *Peripheral Widgets.* The "range slider" widget on the right side of the display controls how many weeks are visible at a time. It acts like a traditional scrollbar, but the thumb has two additional buttons that are used to manually set the start and end dates of the current view. The view dynamically changes as

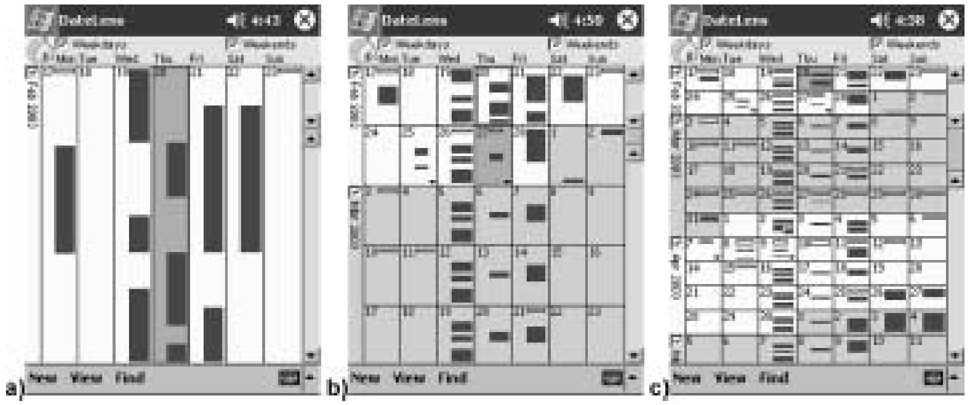

Fig. 2. A series of views as the bottom of the scrollbar thumb is dragged downwards to show: a) 1 week; b) 1 month; c) and 3 months.

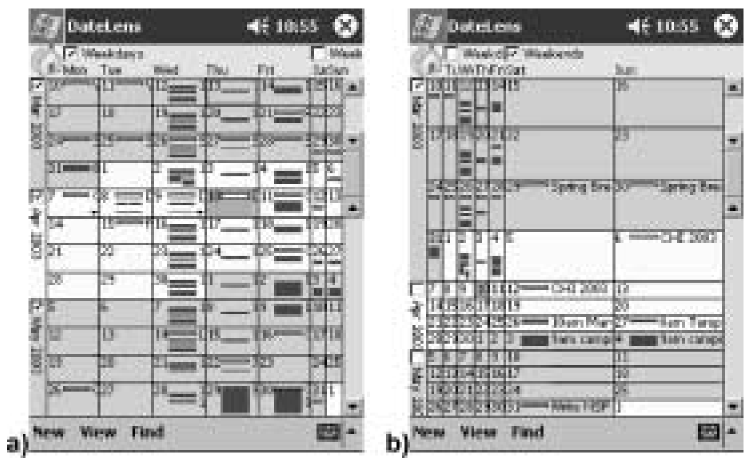

Fig. 3. The views resulting from a) unchecking the weekend checkbox; and b) leaving just weekends in March checked.

the scrollbar is manipulated, but for animation efficiency, appointments within days are only shown when the scrollbar is released. Figure 2 shows a range of views as controlled by the range slider.

Another way to configure the space is to manipulate checkboxes on the top and left sides of the display. These checkboxes specify whether space gets allocated fully to the correlated set of items, or if those items are minimized. The left side of the display has one checkbox for each month. The top side of the display has one checkbox for weekdays and one checkbox for weekends. Figure 3 shows the result of two different configurations of checkboxes.

There is also a “home” button in the top-left corner of the display that resets all navigation settings to a user-configurable state.

2.1.2 *Hardware Buttons.* On desktop computers, graphical user interfaces typically offer keyboard shortcuts so that expert users can quickly access commonly used functions. On PDAs, there is no keyboard, but there are special hardware buttons that applications can use for a similar purpose.

When DateLens runs a Pocket PC device, the “calendar” button can be used to cycle between the preset views of one day, one week, one month, and three months. The “joystick” (a small 4-way rocker switch) offers motion in four directions, and we use that to move the “active” day, which is indicated to the user by a dark blue highlight. Pressing the center of the joystick focuses on that day (or maximizes it if it was already focused). The joystick can also be used when a day is focused or maximized. On a desktop PC, the keyboard can be used to navigate through the calendar by using the arrow keys to change dates, the enter key to zoom in, the escape key to zoom out, and the space bar to switch between preset views.

2.2 Visual Representations

A crucial aspect of the design of DateLens is the visual representation of the calendar for different configurations. We decided to use a “semantic zooming” approach that we developed from our prior work with Zoomable User Interfaces [Bederson et al. 2000]. Semantic zooming means that objects are visually represented differently depending on how much space is available to display them. Using this technique, there are no explicit view modes. Rather, the fisheye distortion algorithms first allocate space, then each cell renders itself using a view that is appropriate to the available space. The graphical views are scaled to fit the available space, while the textual views use a constant-sized font, and the text is wrapped to fit in the available space.

The four available views are:

- *Tiny View*. This shows a graphical representation of the day’s appointments. It includes depictions of all-day appointments with a white rectangle at the top of the day rectangle. It uses color to represent different appointment types, and depicts appointment conflicts using multiple columns. The pen can be dragged across appointments to show tool tips with textual information about the appointment under the pen. In large-scale views, where each row is thinner than a threshold, the black lines separating rows are removed to make the display less “heavy” (Figure 1a).
- *Agenda View*. This shows a textual list of appointments in order by time. There are actually two representations in this view. If there is a smaller amount of space available, a smaller font is used, and the appointment times are not listed. If there is more space available, a larger font is used, and the appointment times are listed (Figure 1b).
- *Full Day View.* This shows a traditional full day view with a schedule of the entire day, and appointments positioned at the appropriate times. It shows all-day appointments and conflicting appointments, and uses color in the same way as the tiny view (Figure 1c).
- *Appointment Detail.* Traditional widgets are used to show the details of a particular appointment (Figure 1d).

The decision of which view to show is, by default, made by DateLens depending on how much space is available in a given cell. There is a threshold cell size for each view that determines what to display.

While the initial design of DateLens was motivated by small displays, one of the attractions of its design is that it scales nicely to larger displays as well. Since the decision of which representation to use is made based on how much space is available for a particular cell, the same layout algorithm and rendering code works on larger displays as well. Figure 4 shows DateLens running on a desktop PC at 1400 × 1050 resolution.

2.3 Search

The last primary component of DateLens is search. Search is important because it lets users identify patterns and outliers within a large time span. When users search in DateLens, the days that contain an appointment matching the search criteria are highlighted. The highlights are maintained while users continue to operate DateLens normally so the space can be explored to understand the results of the search.

In addition to highlighting the visible days within the current view, "attribute mapped scrollbars" [Hill and Hollan, 1994] show which days are highlighted in both the past and future (Figure 5). The scrollbar shows indicators representing which days are highlighted within and outside of the current view. This mapping of search results to the scrollbar is fixed to this single attribute, and is not under user control.

While it is natural to support searching for arbitrary user-entered text strings, that is somewhat problematic because it is notoriously difficult and slow to enter text on PDAs. So, while DateLens supports free text search, it also provides two search mechanisms that do not require text entry: pre-built searches and searches based on existing appointments.

2.3.1 *Free Text Search*. To search manually, users enter text in the text box at the top of the display. A somewhat tricky issue is how to deal with search strings that consist of multiple words. Should the search consist of the conjunction or disjunction of the words, or the actual search string? No one of those approaches worked for all of the experimental tasks in our studies. Instead, DateLens operates like many current Web search engines, using a simulated "vector" based search [Baeza-Yates and Ribeiro-Neto, 1999, pp. 27–30].

Vector searches work by using a number of characteristics of the search to rank the order in which the results are shown. This results in an ordering that usually matches user expectations. Exact string matches are typically listed first, conjunctions (where all the words match) are listed next, and disjunctions (where not all the words match) are listed last.

DateLens is a little different since it does not present an ordered list of search results, but instead highlights whichever days match. Rather than ordering search results, DateLens only presents highly ranked search results. It works by first performing an exact string match, and if there are any results, they alone

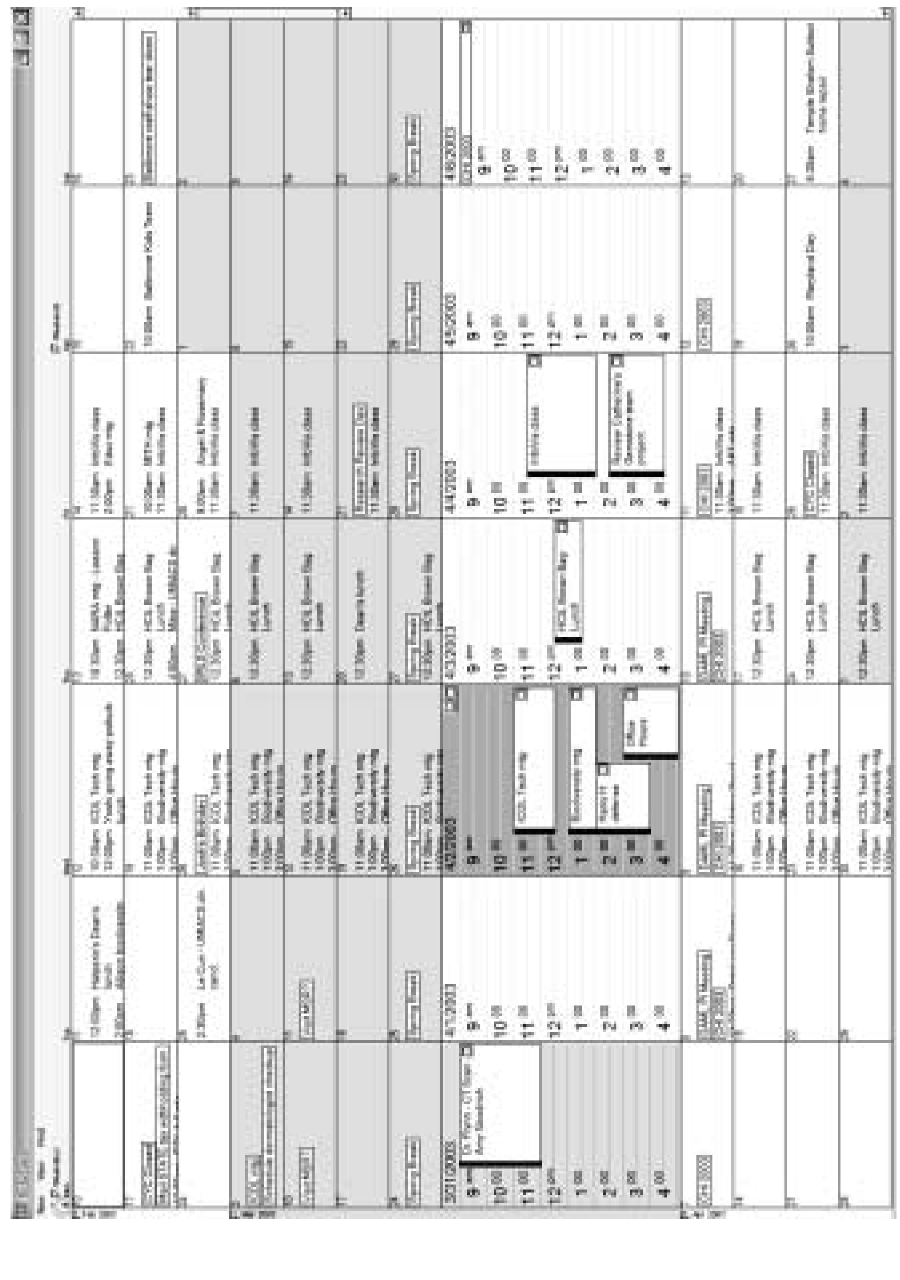

Fig. 4. DateLens scaled up to 1400 × 1050 pixels on a desktop computer. There is room to display an entire week with the full day representation, even while showing three months of the calendar.

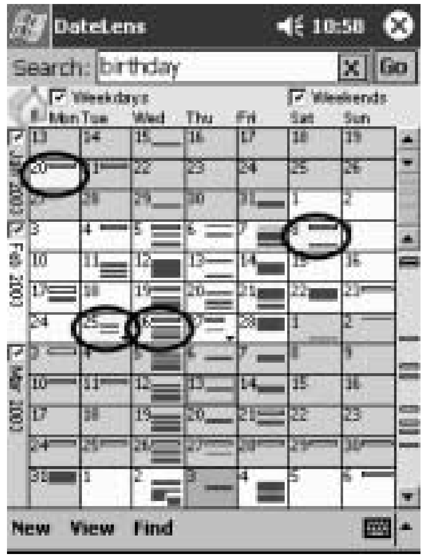

Fig. 5. DateLens showing the results of searching for birthdays (colored highlights are circled for black-and-white printing clarity). A few individual birthdays are highlighted. In addition, the scrollbar shows several days in the future that are highlighted, which also have birthdays.

are shown. If there are no results, then it searches for days with appointments matching all the words in the search string, and highlights those days. If there are still no matches, it then searches for days with appointments that match any of the words in the search string and highlights those. This combination of search strategies mimics the main effect of vector searches, and seems to work well in practice.

2.3.2 *Predefined Searches.* Since it seems likely that many searches by a particular user will be for the same thing, we added support for predefined searches. The goal is to make it even easier to search for commonly sought events, such as travel, meetings, doctor appointments, or holidays.

A simple approach is to search on appointment metadata which is supported by Pocket PC as well as other calendar systems. The problem with this approach is that most users do not annotate each appointment with categories.

Rather than force users to do something they do not want to do, DateLens takes advantage of what information is already available—the appointment text itself. While there are no guarantees that a user will enter a similar event the same way every time, we have found through informal polling of our colleagues, that people often do represent similar events with similar textual descriptions—although they vary significantly from one user to another.

We built support for predefined searches where each search would actually look for a match within any of a set of search strings. For example, searching for "Doctor Appointments" actually searches for "doctor", "dr.", or "dr appt". While these predefined searches are currently hard-coded, our intention is for users to be able to modify them, or define their own.

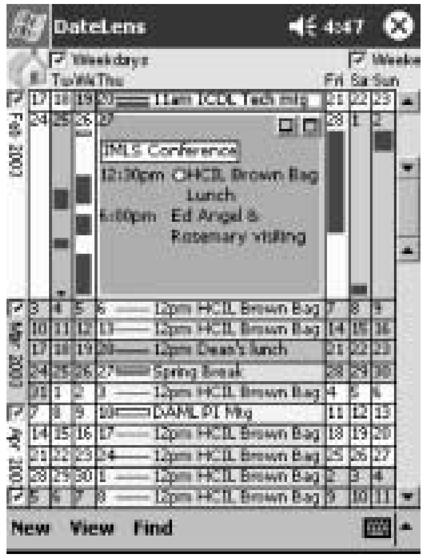

Fig. 6. The results of clicking on the appointment “HCIL Brown Bag Lunch”. Most Thursdays are highlighted in yellow which shows a recurring appointment. In addition, several days are highlighted in orange which match either “HCIL”, “Brown”, “Bag”, or “Lunch”.

This approach has been tested on the authors' calendar data, and it works quite well except for a few idiosyncrasies that we discovered. For instance, one of the authors uses textual graphics such as “– >” to indicate travel. Some of these are searchable as a text string, but some are not because they span multiple lines.

Nevertheless, this approach still appears practical. Since having good quality predefined searches is so useful, some users are likely to adapt the way they write appointments to be more consistent given an easy-to-use search tool. While the general idea of requiring users to adapt to system requirements is undesirable, this is better than the current solution, which requires manual annotation of each appointment with categories.

2.3.3 *Existing Appointment Search.* Since it is quite common for people to create recurring appointments, where the same appointment happens at regular time intervals, it seems natural to have a simple way to support finding and visualizing those recurrences. DateLens does this by offering a way to find all appointments matching an existing appointment. This works by tapping on any appointment. All other days with exactly matching appointments are highlighted in yellow, just as if the text of the appointment subject had been typed into the search box.

We noticed, however, that sometimes users had similar appointments that were not exact matches. Based on the implementation of free text searching, we also search for days with appointments that partially match the specified appointment, and we highlight those in orange (Figure 6). This is a simple solution that can be readily ignored by users if they are only interested in the exact recurrence, but offers more information if desired.

2.4 Implementation

The implementation of DateLens consists of about 20,000 lines of C# code, and runs on whatever platforms the Microsoft Common Language Runtime (CLR) is available on. Currently, the CLR is available on the Pocket PC platform and all desktop versions of Windows except Windows 95. DateLens uses the standard Pocket Outlook database on the Pocket PC platform and the Outlook database on desktop platforms, so users can switch between commercial calendars and DateLens while using the same data. All features described in this article are fully implemented.

The most complex part of the implementation is the layout algorithm used to allocate space for each calendar day. The layout algorithm takes as input the number of days in a week, number of weeks displayed, the checkbox states, the focus day, and the size of the window. The subtle part of the layout algorithm relates to the large set of configurations of the space for which it must work. Specifically, there must be a balance between the minimum size of unfocused cells and the maximum size of focused cells. That is, we have found it makes most sense for each day to stay within a range of sizes whenever possible. So, DateLens defines preferred minimum and maximum sizes for unfocused and focused cells, and allocates space within those ranges whenever possible.

The other subtle part of the DateLens implementation is performance. To make DateLens respond to user interaction rapidly, and to animate transitions smoothly, the overall structure had to be carefully designed. The primary things taken into account that contribute to its performance are:

- Custom rendering loop. Rather than use a toolkit, which might have been easier in some respects, DateLens uses a custom data structure, rendering loop, and "picking" implementation. This was particularly appropriate since the basic data structure is a table, and is easily handled as a two dimensional array.
- Space vs. time tradeoff. Things are precomputed and stored, rather than being computed on the fly. The most obvious place this occurs is in the layout of the days.
- Render only what is needed during transitions such as the basic grid structure. However, some visual aspects, such as highlighted days have to be shown during scrolling since users sometimes look for that while scrolling. In addition, we found that by rendering the appointment details of just the cell that the user tapped on, users were largely unaware of the lower rendering quality during animations. This is because the user's eye is almost always looking at the point of interaction, and is not aware of modest changes in the periphery.

All transitions in DateLens are animated with simple linear interpolation that occurs over 250 milliseconds by default. We picked such a short animation time because the visual changes are quite small (usually not changing by more than a few centimeters).

In order to get DateLens to run on both Pocket PCs and desktops, we largely relied on the multi-platform capabilities of Microsoft's .NET platform. That worked for all of the core functionality of DateLens including animation and rendering. However, in order to take advantage of any device-specific components such as hardware buttons on the Pocket PC and keyboard input on the desktop, we had to write some platform-specific code. In addition, we wrote platform-specific code to access the Outlook database, which uses a slightly different API on the different platforms. In terms of performance, we designed for the slower system (Pocket PC), which then worked well on the desktop. The only significant difference in terms of performance is that on the Pocket PC, we did not render appointment details during animation or scrolling because of the limited processing power; while on the desktop version, we render all details all the time.

3. BENCHMARK STUDY

We performed two studies comparing DateLens to the current shipping user interface of Microsoft's Pocket PC 2002™ calendar (Figure 7). The first was an early benchmark study that we ran before DateLens was running on actual Pocket PC devices, so we performed the study on a desktop computer using a Pocket PC emulator and a mouse. For this first study, we looked at novice users who had not used Pocket PCs before. Once DateLens was running well on Pocket PC devices, we ran a second study with Pocket PC experts from Microsoft using the actual devices with pen input.

The goals of the first study were to examine the initial design ideas behind DateLens, in order to see if the user interface design could be improved, and to compare its overall usability against an existing product.

3.1 Participants

We gathered eleven knowledge workers (five females) who were all experienced Microsoft Windows and Office users, as confirmed through an in-house validated recruiting screener questionnaire. Participants were screened to be between 25–50 years of age (average age of 39.2). In addition, the participants fit some broad characteristics of being target end users of personal digital assistants (PDA), but were purposefully chosen to not own or use one at the current time. We thought this aspect of the user group would be especially interesting since for some reason these users had avoided buying a PDA, and perhaps the presentation of PDA information on a small screen was a primary issue for them.

3.2 Materials

The user's display was an LCD set to 1024×768 resolution with 16-bit color. Each calendar occupied a 240×320 pixel window centered on the display (standard Pocket PC resolution). All participants were run singly in a usability lab, on a Dell Pentium 450 MHz computer running Windows XP. An MS

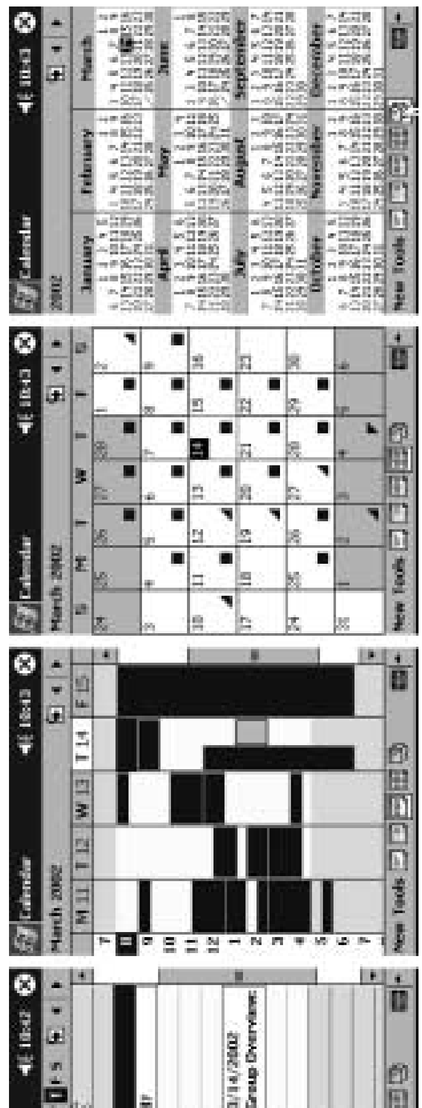

Fig. 7. Screen shots of the Microsoft Pocket PC Calendar program that was used in the study showing day, week, month, and year views.

Natural keyboard and an MS IntelliMouse were used as input devices, though the “wheel” was not functional with the calendars.

3.3 Procedure

Brief (approximately 5 minutes) tutorials were provided to participants prior to each set of tasks on each calendar. The tutorials consisted of a one page sheet of instructions on operating the two interfaces, and the participants then tried each of the described mechanisms. The tutorial focused on the features and functions of each calendar that were necessary for completing the experimental tasks. However, 2 additional minutes were provided for the user to explore the calendar as he or she saw fit prior to starting. The participants performed an isomorphic set of 11 tasks using each calendar (example tasks are listed below). The order of calendar use and task set for the calendar were both counterbalanced in order to minimize the effects of training, or the possibility of one task set being slightly more difficult than the other. However, tasks within a set were not randomized.

We worked with the Pocket PC usability engineers to devise a task list that reflected real tasks carried out frequently by Pocket PC users with their calendars. In addition, based on our own use of the Pocket PC, we added some tasks that were a bit more complicated, requiring more navigation than simply entering an appointment. Therefore, participants completed three different categories of calendar tasks: Searching (e.g., finding the start and/or end dates of appointments), Navigation and Counting (e.g., navigating to particular appointments or monthly views and counting things like the number of conflicts, or the number of Mondays in a month) and Scheduling (e.g., scheduling a meeting, a vacation or a fun night out). The navigation and counting, and the scheduling tasks were the most complex, in general, in that they required users to keep information in short-term memory while navigating or scrolling through timeline views. The navigation and counting category of tasks may not be performed as often by real users as the searching and scheduling tasks, but we added them in an effort to more deeply probe the value of the DateLens visualization, as well as to generalize our findings for the information visualization community, where navigation can be a key issue.

All tasks were given a deadline of two minutes to complete in order to keep the session under two hours (and because a two minute deadline seemed reasonable for being able to discover information from one’s PDA calendar.) Task times and success rates, verbal protocols, and user satisfaction and preference questionnaire data were collected throughout the session. Sessions lasted approximately one and one half hours. The complete tutorials and task lists for this study are available in Appendix A in the ACM Digital Library.

3.4 Results

3.4.1 *Task Times.* Task times for one participant were unavailable, as his session expired before he was able to get to the 4th task using DateLens. A tape jam prevented us from obtaining the task times for one other participant for the Pocket PC, and both participants’ data had to be ignored for the task

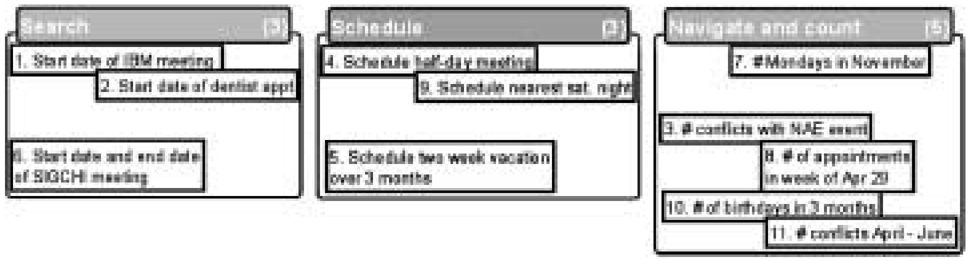

Fig. 8. Summary of tasks grouped by type and complexity. In each group, the top tasks are the "simple" ones, and the bottom ones are the "complex" ones.

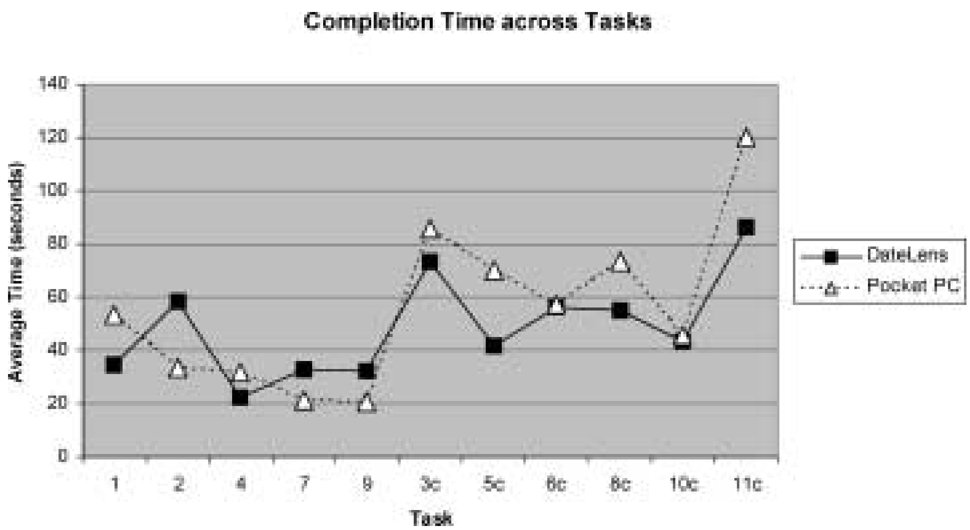

Fig. 9. The time spent by study participants to complete each task using the DataLens and standard Pocket PC calendar interfaces. Tasks are listed with simple tasks on the left, and complex (marked with a 'c' in the task number) on the right.

time analysis. A 2 (calendar type) $\times$ 11 (Task) repeated measures Analysis of Variance (RM-ANOVA) was carried out on the completion times for the tasks. Tasks were performed faster using DateLens (49 seconds versus 55.8 seconds for the Pocket PC, on average), but this result did not reach significance, $F(1, 8) = 3.5$, $p = .08$. There was also a significant main effect of task, $F(10, 80) = 12.9$, $p < .01$, and a significant calendar $\times$ task interaction, $F(10, 80) = 2.05$, $p = .04$. Of particular interest was the fact that, as the tasks became more complex (tasks 3, 5, 6, 8, 10 and 11), the DateLens task time advantage grew reliably. We divided the tasks into "simple" and "complex" by calling the five fastest tasks the "simple" ones, and the six slowest tasks the "complex" ones, and found that DateLens performed 12% faster for the complex tasks. We measured the task speed for this definition as the fastest of the two average completion times (one for each interface). Figure 8 shows a summary of the tasks, grouped by type and complexity. The time that participants spent on each task can be seen in Figure 9.

We decided to carry out another analysis based on our significant effect of task, comparing the tasks we observed to be simpler to those that were relatively more complex (again, task 3, 5, 6, 8, 10 and 11) to explore this result further. A 2 (calendar type) × 2 (simple v. complex task average) RM-ANOVA was performed on the task times. Again, no significant effect of calendar was obtained, but there was a significant effect of task complexity, $F(1, 9) = 79.55$, $p < .001$, and a significant interaction between calendar type and task complexity, $F(1, 9) = 8.4$, $p = .017$. The post-hoc analyses revealed that the Pocket PC calendar provided a significant advantage for simpler tasks, while the DateLens application provided a significant advantage for the more complex tasks (all p-values less than .05 using Bonferroni adjustments for multiple tests). Our observations of users in the study led us to believe this latter finding was primarily due to the fact that DateLens allowed flexible views across time in a user-defined manner during complex tasks. By allowing the users to view as little or as much information as needed for a particular task and by using animation to help users more clearly see view transitions, short-term memory load was reduced in the DateLens application. In the Pocket PC application, rigid views (day, week or month) meant that multi-view questions required retaining search results in short-term memory while switching through the required views for the complex tasks. In addition, the integrated search mechanism and its resultant views in DateLens made finding particular sets of events via keyword matching quite effective. Unfortunately, it is not possible to tease out exactly how often users used the search mechanism as opposed to adjusting the views in this study. The searching tasks were the likely candidates for using search, but sometimes users tried search and then quickly jumped to adjusting the view or vice versa, so in many cases a combination of both was involved. Since the Pocket PC showed a day view by default, simpler tasks were well supported via a familiar, Outlook-style view, and this may have led to significant performance gains for those tasks. These results can be seen in Figure 10.

3.4.2 *Success Analysis.* A 2 (calendar type) × 11 (Task) repeated measures Analysis of Variance (RM-ANOVA) was carried out on the success rates for the tasks. Overall, tasks were successfully completed significantly more often using DateLens (on average, an 88.2% success rate, versus 76.3% for the Pocket PC), $F(1, 9) = 37.1$, $p < .001$. In addition, there was once again a significant main effect of task, $F(10, 90) = 12.9$, $p < .001$. The interaction, however, was not significant. These data are shown in Figure 11, where it becomes clear that certain tasks (3, 5, 6, 8, 10 and 11) were successfully completed more often with DateLens, on average, but the result was not significant. For the most difficult task (#11), no participant using the Pocket PC completed the task successfully.

3.4.3 *Satisfaction and Preference.* Users completed “ease of use” ratings on a scale of 1 to 5 (1 = very difficult, 5 = very easy) after every task. DateLens was rated higher across a majority of the tasks, especially the most difficult task (task 11—how many conflicts are there for the next 3 months?—had the lowest success rate for both calendars). A 2 (calendar type) × 11 (Task) RM-ANOVA was carried out on the satisfaction data. DateLens was rated higher than the

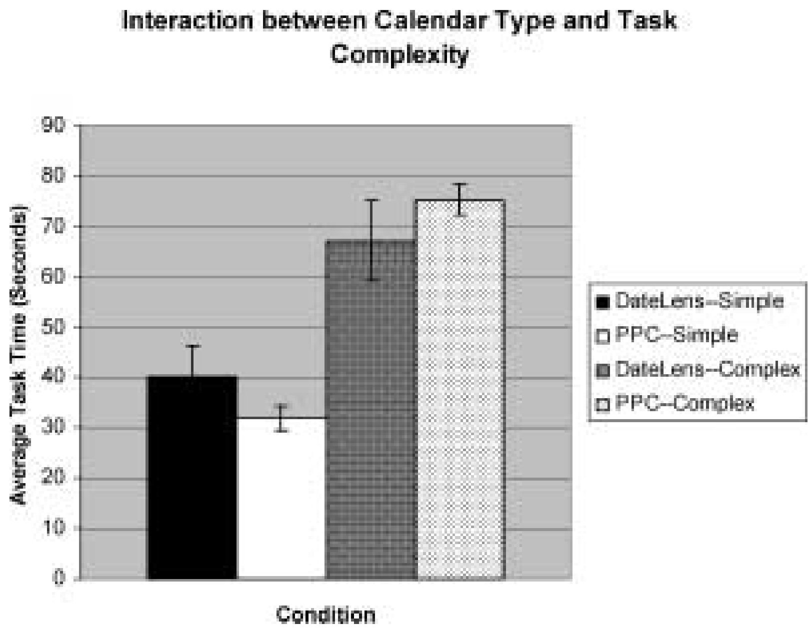

Fig. 10. Effects of calendar type by task complexity on average task times.

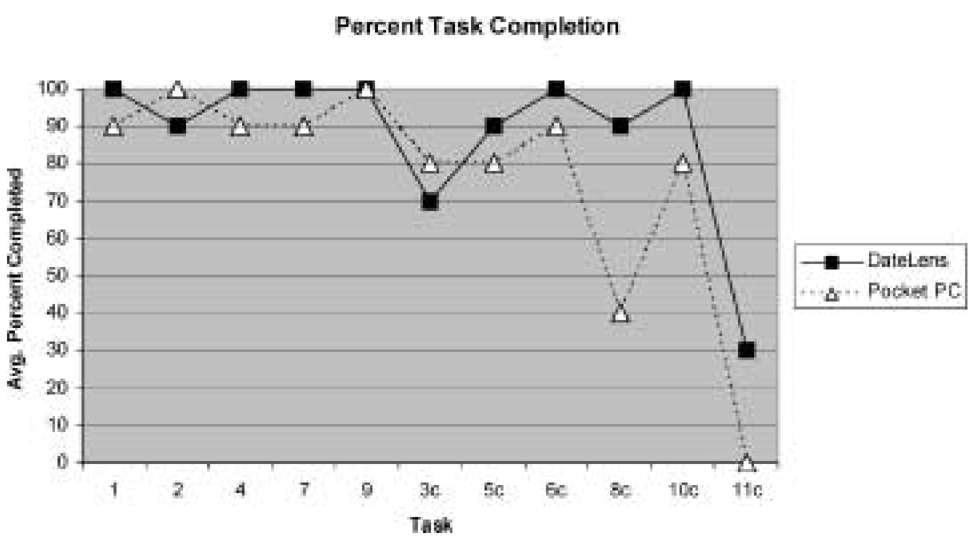

Fig. 11. The percent of tasks completed by study participants for each task. Tasks are listed with simple tasks on the left, and complex (marked with a 'c' in the task number) on the right.

Pocket PC in terms of task satisfaction, on average, $F(1, 9) = 4.37$, $p = .06$, a borderline significant result. The average task ratings are shown in Figure 12. As can be seen in the Figure, only tasks 2, 3 and 7 were rated highest for Pocket PC. This only correlates partially with success rate, since tasks 2 and 3 were

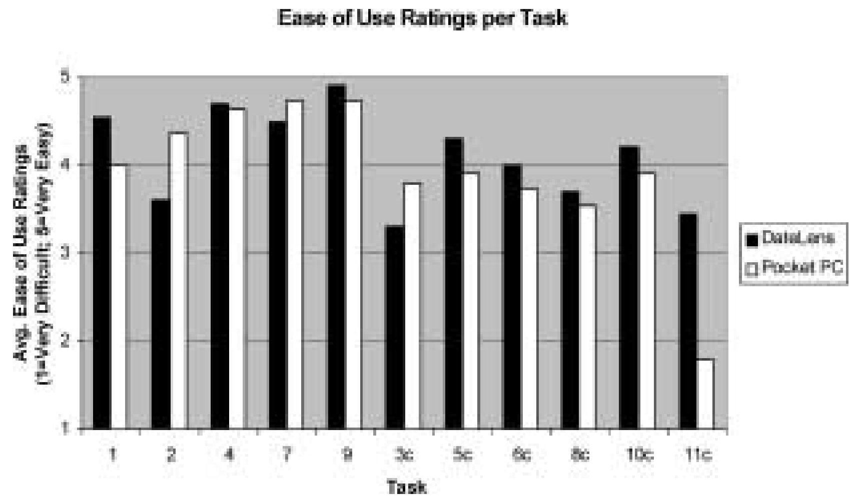

Fig. 12. The ease of use rating for each task by study participants. Tasks are listed with simple tasks on the left, and complex (marked with a 'c' in the task number) on the right.

carried out more successfully with that calendar but task 7 was carried out more successfully with the DateLens application.

Participants completed an overall preference questionnaire at the end of the session. No significant differences emerged in this preference data, though the Pocket PC was slightly more preferred overall (6 out of 11 participants chose the Pocket PC Calendar; one participant abstained and stated that she wanted features of both calendars in the ideal calendar; 4 participants chose DateLens). Most participants said that they would prefer a combination of features from each of the two calendars during the post-session debriefing. The most often cited reason for choosing the Pocket PC calendar was the participants' familiarity with the Outlook XP calendar, which is similar in many ways. The overall preference data is shown in Figure 13.

3.4.4 *Usability Issues.* Many usability issues were observed with this initial version of DateLens, as well as the Pocket PC calendar, and good design feedback was received from the participants about how best to move toward redesign. We summarize those issues pertaining to DateLens here.

Many users disliked the view of the calendar when more than 6 months were shown at once, claiming that the individual days were simply too small at that point to be useful. In addition, users wanted to see all 24 hours of a day's full view, but the prototype was limited in functionality to simply show a 9-5 view for this iteration of user testing. More importantly, a visualization of search results tried to show as many "hits" in the calendar as possible without making the view so crowded as to be useless. If the result a user was looking for was scrolled out of view (into the future), there was no visual indicator as such (the attribute mapped scrollbar that shows search results was added after the study was run.) Users voiced strong concerns about the readability of text, and being able to set their own default views according to their individual eyesight needs.

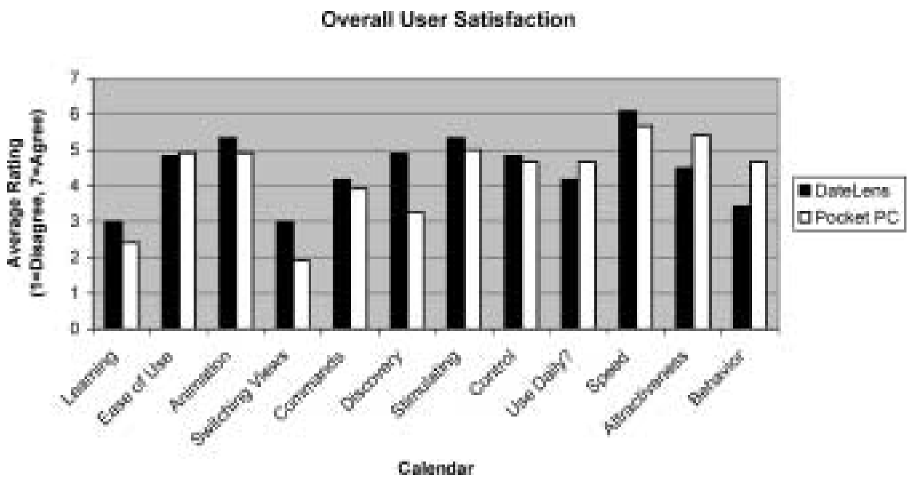

Fig. 13. Overall user satisfaction.

Users also wanted more control over how their weeks were viewed (e.g., should the week start with Sunday or Monday?). Finally, users wanted better visual indicators of conflicts for both calendars, for example, red highlights and/or a "conflicts" filter. These usability issues were of a generic nature and were seen to affect the use of DateLens pretty much during every task, though the search issues obviously strongly affected the tasks that required direct searching. As a result, many users attempted to use search to target known events in a task, but often abandoned this effort half-way through or adjusted the view in addition to attempting to search. Improvements in text legibility and adding an attribute-mapped scrollbar might enable better utilization of the search mechanism in future versions of DateLens. Users in this study did not choose to browse when completing tasks.

In summary, DateLens performed quite well despite its novelty and this being its first iteration of user testing, especially for tasks that turned out to be complex and that required custom views or large amounts of navigation. The responsiveness to direct user manipulation, the ability to easily create custom views on the fly, its clear presentation of conflicts, and integrated search utility were all design innovations that participants thought would be valuable to any calendar used daily for planning and reviewing one's schedule. The Pocket PC calendar was seen by participants to be consistent with other Microsoft calendar products, and this was viewed positively. Several participants wanted to see a combination of the two calendars taking advantage of the good features of both in a final product.

4. FOLLOW-UP USER STUDY

Our second user study also compared DateLens to the Pocket PC 2002 Calendar with the following differences in the software and study design:

- Both conditions were run on the Pocket PC device with a stylus for input instead of using a desktop with mouse.

- The DateLens software was improved to respond to several of the biggest usability concerns—including the ability to see appointments outside 9am–5pm, and the ability to see search hits outside the current view (through marks on the scrollbar).
- We performed a combined longitudinal and lab study. We wanted to examine the usefulness of the visualization over a longer time period than the typical lab situation would allow, in the user's own work environment with their personal data. We believed that we would learn much more about the usefulness of both calendars if users used them in situ and as part of their daily routine. However, we also wanted the accuracy and control afforded by the laboratory experience. We decided to combine both methods for an acceptable evaluation. After deploying DateLens to our users, we had them use it for three entire days before we performed a comparative evaluation of it and the Pocket PC calendar. Users were still more familiar with the Pocket PC calendar in the end, but at least they got a chance to experience the DateLens software for a number of days before the experimental session occurred.
- We chose participants who were experienced Pocket PC users. This had the benefit of giving us feedback from real target end users who were familiar with PDA calendaring. Since DateLens is a research project, and personal calendar data is very important to individuals, we decided to include only willing participants from Microsoft due to privacy concerns that might be an issue with external participants. This introduced a potential bias, but it is likely to be in favor of the Microsoft product. Because of this, we feel that any advantage seen in DateLens is likely to be more pronounced among a general user population.

4.1 Participants

Eight Microsoft employees (1 female) who use Pocket PC heavily every day were recruited for participation. The participants ranged in age from 28 to 36 years. All were expert PC users and had been using the Pocket PC for over 2 years. Their job titles varied from usability engineer, designer, and researcher to group manager. Participants were given vouchers for food or coffee from the cafeteria as a gratuity.

4.2 Materials

Participants were sent email study instructions that included download and installation information for DateLens, a brief tutorial on how to use the key features in the DateLens software and a survey that participants were to fill out after using the software for 3 days. Participants used their own Pocket PC devices, which varied widely in terms of the amount of memory included, form factor and even version of Windows CE (some of the participants were running a beta version of the next release not available to the public). However, in order to standardize the study to some degree, we required all users to drop back to using the 2002 version of their Pocket PC software. Users filled out another questionnaire after performing an experimental task session with both the

Table I. Average Satisfaction Ratings After Using the DateLens Software for 3 Days (1 = Disagree, 7 = Agree)

This calendar was difficult to learn how to use:	4.57
This calendar was easy to use for the tasks you performed today:	3.86
The animations allowed me to stay better oriented as I switched to different calendar views in the user interface:	3.71
It was difficult to switch to the proper view in the calendar:	4.00
I sometimes wondered if I was using the right command:	4.00
I was able to discover more about a schedule using this calendar compared to existing tools:	3.57
Working with this calendar was mentally stimulating:	4.33
I felt in control of this calendar when I was using it:	3.00
I would like to use this calendar every day:	3.00
The speed of this calendar is about right:	3.71
This calendar has a very attractive presentation:	2.29
This calendar sometimes behaves in a way I don't understand:	4.86

DateLens and Pocket PC software. The design of this second questionnaire is based on the IBM Post-Study System Usability Questionnaire [Lewis 1995].

4.3 Procedure

Participants were asked to use DateLens to view their own personal calendar data for 3 days. After they had done so, they filled out a survey about their initial reactions. Upon receiving this survey, we asked each participant to perform two sets of isomorphic tasks, once using DateLens and next using the Pocket PC calendar. The order of presentation of each of the two calendars was counterbalanced, as was the task set. The task set was replicated from those tasks used in the first study, however some abstraction and/or changes were required due to the fact that we were using the participants' real data. The entire task set for the second study is available in Appendix B. If a participant did not have any data in their calendar corresponding to a task, the task was either skipped (this occurred only twice) or a very close alternative was used in the target's place (e.g., find your next dentist appointment was allowed if there was no doctor's appointment on the participant's calendar). All tasks were timed using an electronic timing program on the experimenter's laptop. In addition, the experimenter recorded comments, errors and observations provided by the participants for both calendar visualizations while performing the tasks. Finally, after completing all tasks on each calendar, the participant filled out a standardized satisfaction questionnaire.

4.4 Results

After using DateLens for three days, users rated its initial ease of use. Mean ratings for DateLens are shown in Table I, based on a Likert scale rating system of 1 = Disagree, 7 = Agree). Users felt that the DateLens calendar needed a more attractive design, and users indicated that they did not feel in control when they used the calendar, and that it behaved unpredictably at times. For these reasons, their likelihood of using the software daily was rated lower, on average. However, these ratings really point to remaining usability problems

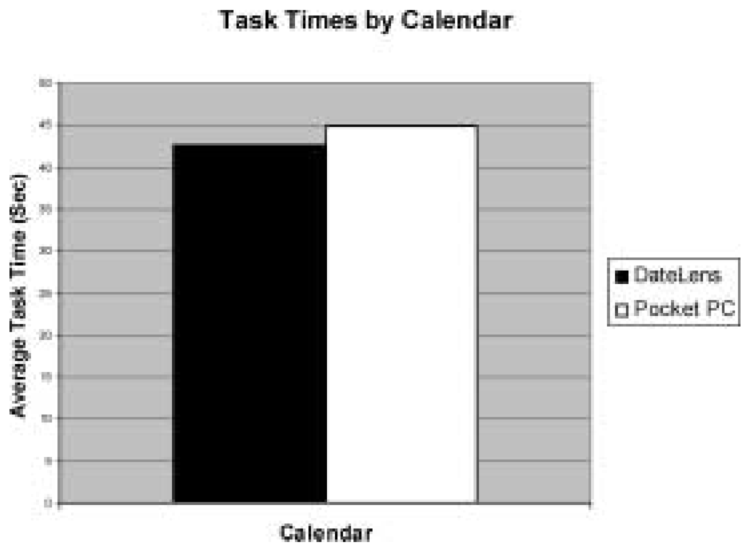

Fig. 14. Overall average task times for each calender.

(more on this is detailed below) and are not necessarily indicative of flaws in the concept of the fisheye calendar. They do point to the challenge of making novel visualizations work effectively in everyday tasks especially with users who are already quite familiar with existing tools. These ratings also provide a good barometer for what aspects of the software should be focused on for further improvements to the design.

The results from the experimental session provided task time results allowing us to compare the two calendars quantitatively. A 2 (calendar) × 11 (task) RM-ANOVA was performed on the task times. No main effect was observed for calendar, $F(1, 5) = 0.36$, $p = .58$, but a main effect of task was observed, $F(10, 50) = 4.0$, $p < .001$. No interaction was obtained. This data is shown in Figure 14. However, based on our complexity analysis from the first study, we suspected it was the complexity of the task that might be driving results in this second study as well. Therefore, the same analysis was carried out again. A 2 (calendar type) × 2 (complexity of task) RM-ANOVA was carried out on the average task time data. Results were identical to the first study, in that no significant main effect emerged for calendar type, but a significant main effect for complexity was revealed, $F(1, 7) = 29.83$, $p < .001$, and a significant interaction between calendar type and complexity was observed, $F(1, 7) = 8.9$, $p = .02$. Post-hoc tests showed that once again the Pocket PC was reliably better for the simple task set, while the DateLens calendar provided significant benefits for the more complex task set, with a 17% average time improvement. These results can be seen in Figure 15.

The post-experimental task satisfaction questionnaire results were also analyzed. DateLens received a slightly worse average satisfaction rating, as can be seen in Figure 16. This difference did not reach statistical significance in a 2 (calendar) × 13 (question) ANOVA, $F(1, 7) = 1.34$, $p = .28$. The only significant

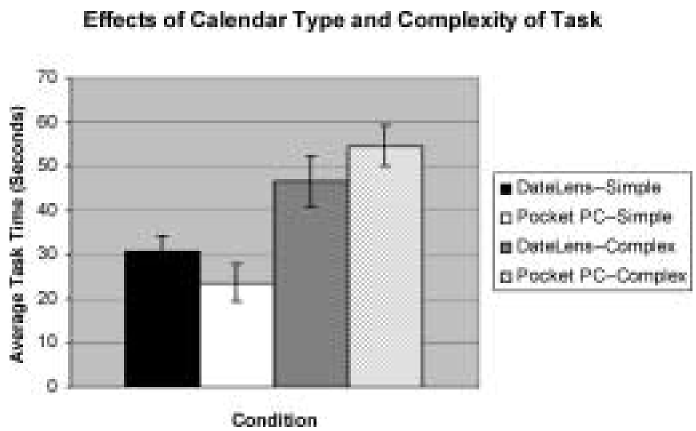

Fig. 15. Effects of calendar type and task complexity on overall average task times in study 2.

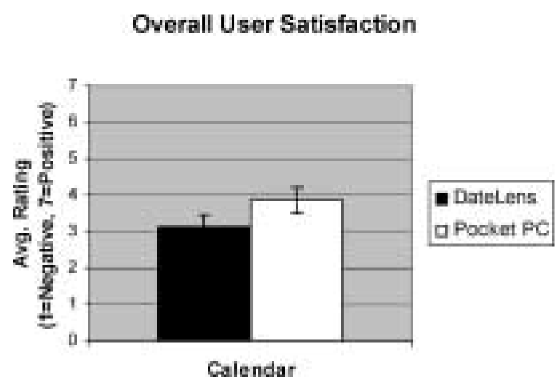

Fig. 16. Overall average satisfaction for each calendar based on the experimental tasks.

effect observed in the analysis was that of the interaction between calendar and questionnaire item, $F(10, 70) = 3.07$, $p = .002$. Post-hoc t-tests showed only one significant difference among the items at the $p = .05$ level. Question 4 asked participants whether they could effectively complete the experimental tasks using each system, with DateLens rated significantly higher than Pocket PC, $t(7) = -2.9$, $p = .02$, two-tailed. There were two borderline significant differences in favor of the Pocket PC: Question 7 asked about user comfort while using the systems, and not surprisingly the Pocket PC was rated borderline significantly more comfortable, $t(7) = 2.2$, $p = .06$, two-tailed. Question 8 asked users how easy it was to learn to use each system, with Pocket PC rated borderline significantly easier to learn, $t(7) = 2.2$, $p = .06$. All of the satisfaction ratings can be seen in Table II. Note that the actual questionnaire recorded satisfaction with low values being positive, but we are reporting the data here flipped to see the results more clearly.

Several usability issues were observed for both calendars in this study, although for DateLens the usability issues that were observed were now quite

Table II. Average Post-Task Satisfaction Ratings for Each Calendar (Note That Lower Ratings Are Better). DL = DateLens, PPC = Pocket PC

Questionnaire Item	DL	PPC
Overall, I am satisfied with this system.	3.9	3.3
Overall, I am satisfied with how easy it is to use this system.	4.4	2.6
It was simple to use this system.	4.6	2.9
I could effectively complete the tasks and scenarios using this system.	2.8	4.5
I was able to complete the tasks and scenarios quickly using this system.	3.3	3.6
I was able to efficiently complete the tasks and scenarios using this system.	3.9	4.0
I felt comfortable using this system.	4.0	2.0
It was easy to learn to use this system.	4.8	2.6
The “look” of this system was pleasant.	4.0	3.3
I liked using this system.	3.6	2.8
The organization of information in this system was clear.	3.5	3.0

different from the first iteration because of changes we made to the design in the interim. Therefore, we will focus on the usability issues we observed for the DateLens visualization, with an eye toward ideas for further design improvements. DateLens was in a period of rapid development, and we ran the second study at a point before it was finished. So, some problems were observed that have since been fixed. The most serious issues we observed concerned tap targets in the user interface simply being too small. Experienced Pocket PC users often use their fingers to tap on targets in the user interface, either because they have misplaced their stylus, they don't have time to grab the stylus, or because the unit is docked and finger tapping is preferred over stylus tapping in the vertical orientation. Even with the stylus, users often invoked incorrect behaviors and actions accidentally when attempting to scroll or make UI selections in DateLens. Future versions of the UI could enlarge the size of frequently used targets, especially the scrolling controls and checkboxes. In addition, the scrollbar had two sets of arrows, which were not distinguishable except for by their spatial positioning. Users wanted to see a new design for the specialized scrollbar, in which the different arrows had discriminating features so that the user could tell which feature mapped to which function. We are experimenting with alternative designs to support this, such as replacing the arrows on the inner buttons with hash marks. Finally, some issues remained surrounding the visualization of search results. While users really enjoyed the ability to search with keywords and some pre-defined defaults (something that the Pocket PC does not do in an integrated fashion in the calendar), search results sometimes were only viewable off screen, and users did not necessarily see the “hits” feedback in the elevator of the scrollbar, indicating off-screen results. This would lead to users not knowing if a search had completed or if it had ever been invoked at all. Adding an hour-glass style piece of feedback or scrolling the user to the most temporally proximal search result would alleviate user confusion in these instances. Clearly these usability issues affected most tasks, though again the search usability issues were more prone to detract from the usefulness of that functionality in DateLens. And, once again, because of issues with search, participants often chose to use a combination approach of searching plus view adjustments, or moved to using more view adjustments to navigate rather than relying solely on search. In general, this set of users did not browse to complete their tasks.

5. CONCLUSION

From these two studies, we see that DateLens has strong potential. It offers better performance for the more complex tasks that we looked at, many users preferred it over the traditional static calendar design, and it offers the potential of having a single design that scales across device sizes. Furthermore, these positive outcomes came in comparison to a refined shipping product. DateLens is still rough around the edges, being a research prototype. Based on the observed usability problems, we are confident that DateLens could perform better than it did in the studies. In fact, there is no conceptual reason that DateLens could not support the "simple" tasks as well as the Microsoft calendar, yielding a solution that is better in all cases. In fact, after we ran the studies, we modified DateLens to better support the style of use we observed. For example, we added support for making the single day view the default, we added toolbar buttons at the bottom to jump directly to day, week, month, or 3 month views, and we added a navigation bar at the top to directly jump to different days of the week.

Given our experience with DateLens, what can we say are DateLens' strengths? They appear to be what we hypothesized—the support of complex tasks such as those associated with longer lengths of time. One of the primary reasons for the difficulty of these tasks is that they require comparison between multiple dates, and an understanding of a large expanse of time. In order to do these things, traditional calendar designs require users to switch between the different views (day, week, month, and year)—keeping the relationship between dates *in their heads*. That is, we believe that traditional calendar designs require greater use of short term memory to solve more complex tasks.

By showing a visual representation of dates in context with animated transitions, users appear to be able to compare multiple dates across larger expanses of time with reduced memory load. This translates into shorter task times and subjective preference for some users. While we saw a smaller benefit in the second study, it is possible that this is due to the extreme familiarity with the traditional static interface among the study participants. In addition, by performing a longitudinal study, we relied on the participants to use their own devices—which had varying resources, and some had older machines on which DateLens performed more slowly. Finally, the 2nd study participants lived primarily by the "day" view, and claimed to not use their Pocket PC much for scheduling tasks. This may be partially because Microsoft employees work in a very technology rich environment, and when scheduling was required, a full size computer was frequently available. Thus, we may have tapped into a user population that might not be served well by the added flexibility in DateLens. The first study participants, on the other hand, reported needing this kind of support for performing their daily calendar tasks.

Since managing one's calendar is so important, many users will be cautious about adopting non-traditional interfaces. Thus, one of the biggest remaining challenges is to refine DateLens so that it is appreciated by a broad spectrum of users. This is a significant lesson of DateLens: for end-user applications, it is not enough to use visualization to improve performance on some tasks as performance must not be hurt on other tasks. Given the complexity and novelty of visualizations, this is a daunting goal, but one that we all must strive for.

Based on our experience and the results of the user studies we performed, we offer the following general design guidelines for builders of calendar interfaces:

- Support complex tasks by reducing the user's short-term memory load. DateLens does this by using a fisheye view that enables users to see larger and more custom sets of dates, as well as detailed date information in context of the surrounding calendar.
- Help users understand view transitions by using animation. While we have not formally studied the effectiveness of animation in DateLens, our experience points to its utility, and our previous work has shown the benefit of animation in similar situations [Bederson and Boltman 1999].
- Offer an easy way to switch between views and the ability to "live" in different views. Many of our users preferred to keep DateLens in the single day view between uses, but others liked to keep it in the month view and navigate from there.
- Offer users the ability to configure the interface to control the visualization. While we attempted to design DateLens to present information as clearly as possible, we observed users with different appointment characteristics than we had designed for.
- Offer the option of showing search results integrated into the calendar view. We received many positive comments about this ability in DateLens.
- For PDAs, create interaction targets that are as large as possible so that users can use the calendar with their fingers instead of a stylus. This remains a difficult but important challenge when the goal of the visualization is to show as much information as possible. A corollary to this guideline is to allow the user to adjust the font size to be appropriate in novel visualizations as well.

What is perhaps the most interesting question is also the most difficult. That is, can we now offer any advice about where fisheye distortion displays are likely to be useful more generally—outside of the domain of calendars? We can answer this only with our intuition, and do as follows. We found that key characteristics of our solution that made DateLens work were:

- A simple navigation mechanism. All nodes are visible, and a tap always expands a node.
- A simple representational structure. The familiar tabular representation of calendars made a grid a natural representation for us to follow. This rectilinear structure seems to be easy for people to understand. There have been reports that fisheye distortion of other structures can be disorienting to users. We didn't hear a single comment of this nature from either user study.
- Clear small representations of data. A basic characteristic of fisheye views is that sometimes, data must be presented in a very small space. Thus, it is crucial that all data has some meaningful representation that can fit in a

tiny space. We found calendar data to work well in this regard, and we found several natural and meaningful small representations. The smallest of which was just the date, which fit in a 12×12 pixel box.

While it is certainly possible that fisheye applications that don't have these characteristics will also be useful, we believe that applications that do focus on these traits have a higher likelihood of success.

6. FUTURE WORK

There are several areas of future work for DateLens. While technically, we are pleased with DateLens' performance and robustness, there is clearly room for design improvement. We found a number of usability issues during the user studies that must be addressed. Naturally, there is also a long list of features that users have asked for that must be looked into as well, such as support for faster data entry. Understanding how these changes affect users and keeping DateLens easy enough for novice users to feel comfortable with will be an ongoing and crucial challenge.

More user studies must be run after the next round of usability improvements have been made to develop a deeper understanding of the potential effectiveness of fisheye visualizations. There clearly are a number of usability issues in the current implementation of DateLens which got in the way of its effectiveness in the current studies. In addition, we would like to design a specific study to tease apart the individual influences of integrated search and the flexible, fisheye visualization to complex tasks. We would also like to know more about the frequency that users typically perform the more complex tasks that DateLens supports well.

Finally, further design issues are likely to come up when the DateLens interface is applied to smaller devices (such as cell phones) and larger ones (such as tiled displays). While the basic paradigm scales nicely, there are likely to be specific details that need to be changed for different sized displays.

ACKNOWLEDGMENTS

We greatly appreciate the comments of our colleagues through the many revisions of DateLens. We particularly appreciate John SanGiovanni who has been a tireless supporter of this project at Microsoft Research and has helped us through many iterations of the software. We also appreciate the efforts of Neema Moraveji, a University of Maryland student who helped us to initially integrate DateLens with Outlook.

Finally, we thank Susan Wilhite for her help in running the first user study, along with Ben Shneiderman, Catherine Plaisant and Hilary Hutchinson for their comments on drafts of this paper.

REFERENCES

BAEZA-YATES, R. AND RIBEIRO-NETO, B. 1999. *Modern Information Retrieval*. New York: ACM Press.

BEDERSON, B. B. 2000. Fisheye Menus. *UIST 2000, ACM Symposium on User Interface Software and Technology, CHI Letters 2*, 2, 217–225.

BEDERSON, B. B. AND BOLTMAN, A. 1999. Does Animation Help Users Build Mental Maps of Spatial Information? *In Proceedings of Information Visualization Symposium (InfoVis 99)* New York: IEEE, 28–35.

BEDERSON, B. B., CLAMAGE, A., CZERWINSKI, M. P., AND ROBERTSON, G. G. 2003. A Fisheye Calendar Interface for PDAs: Providing Overviews for Small Displays. In *Proceedings of Extended Abstracts of Human Factors in Computing Systems (CHI 2003)*. ACM Press, 618–619.

BEDERSON, B. B., MEYER, J., AND GOOD, L. 2000. Jazz: An Extensible Zoomable User Interface Graphics Toolkit in Java. *UIST 2000, ACM Symposium on User Interface Software and Technology, CHI Letters 2*, 2, 171–180.

BJÖRK, S., HOLMQUIST, L. E., LJUNGSTRAND, P., AND REDSTRÖM, J. 2000. PowerView: Structured Access to Integrated Information on Small Screens. In *Proceedings of Extended Abstracts of Human Factors in Computing Systems (CHI 2000)*. ACM Press, 265–266.

BJÖRK, S., HOLMQUIST, L. E., REDSTRÖM, J., BRETAN, I., DANIELSSON, R., KARLGREN, J., AND FRANZÉN, K. 1999. WEST: A Web Browser for Small Terminals. *UIST 99, ACM Symposium on User Interface Software and Technology, CHI Letters 1*, 1, 187–196.

FURNAS, G. W. 1986. Generalized Fisheye Views. In *Proceedings of Human Factors in Computing Systems (CHI 86)*. ACM Press, 16–23.

FURNAS, G. W. 1991. *The Fisheye Calendar System*. (Report No. TM-ARH-020558). Bellcore, Morristown, NJ.

HILL, W. AND HOLLAN, J. 1994. History-Enriched Digital Objects: Prototypes and Policy Issues. *The Information Society 10*, 2, 139–145.

INXIGHT. 2003. http://www.inxight.com.

LAMPING, J., RAO, R., AND PIROLLI, P. 1995. A Focus+Context Technique Based on Hyperbolic Geometry for Visualizing Large Hierarchies. In *Proceedings of Human Factors in Computing Systems (CHI 95)*. ACM Press, 401–408.

LEWIS, J. R. 1995. IBM Computer Usability Satisfaction Questionnaires: Psychometric Evaluation and Instruction for Use. *Int. J. Hum.-Comput. Inter 7*, 1, 57–78 .

MACKINLAY, J. D., ROBERTSON, G. G., AND DELINE, R. 1994. Developing Calendar Visualizers for the Information Visualizer. In *Proceedings of User Interface and Software Technology (UIST 94)*. ACM Press, 109–118.

PLAISANT, C., AND SHNEIDERMAN, B. 1992. Scheduling Home Control Devices: Design Issues and Usability Evaluation of Four Touchscreen Interfaces. *Int. J. Man-Machine Studies 36*, 375–393.

RAO, R. AND CARD, S. K. 1994. The Table Lens: Merging Graphical and Symbolic Representations in an Interactive Focus+Context Visualization for Tabular Information. In *Proceedings of Human Factors in Computing Systems (CHI 94)*. ACM Press, 318–322.

SARKAR, M., AND BROWN, M. H. 1992. Graphical Fisheye Views of Graphs. In *Proceedings of Human Factors in Computing Systems (CHI 92)*. ACM Press, 83–91.

SCHAFFER, D., ZUO, Z., BARTRAM, L., DILL, J., DUBS, S., GREENBERG, S., AND ROSEMAN, M. 1997. Comparing Fisheye and Full-Zoom Techniques for Navigation of Hierarchically Clustered Networks. In *Proceedings of Graphics Interface (GI 97)*. Canadian Information Processing Society, 87–96.

SPENCE, R. AND APPERLEY, M. 1982. Data Base Navigation: an Office Environment for the Professional. *Behaviour & Information Technology 1*, 1, 43–54.

Received March 2003; revised August 2003; accepted October 2003