

Palm Redesign

REDESIGN AND RETROSPECTIVE

Final Project

05-410 Introduction to HCI Methods | Fall 2004

Bonnie John, Instructor
Chris Neuwirth, Instructor
Aaron Bauer, TA
Gahgene Gweon, TA
Andrew Ko, TA

GROUP 3

David Hurley
David Hwang
Julian Missig
David Murray
Purin Phanichphant

TABLE OF CONTENTS

page 2	1. Executive Summary
page 3	2. Redesign <ul style="list-style-type: none">2.1 Date Book2.2 Consistency and Comprehensibility2.3 Conclusion
page 16	3. Retrospective <ul style="list-style-type: none">3.1 Contextual Inquiry and Contextual Design (CI/CD)3.2 Heuristic Evaluation (HE)3.3 Cognitive Walkthrough (CW)3.4 Think Aloud (TA)3.5 Interaction Relabeling3.6 Conclusion

APPENDIX A: Design Changes

APPENDIX B: Evidence

1. EXECUTIVE SUMMARY

Over the past several months, our interface redesign team performed various usability analyses in order to identify possible problems with the PalmOS 3.5 Color interface. This report summarizes our findings, offers possible redesign solutions, and reflects on all the methods used to arrive at those findings.

Out of the five male undergraduates that constitute our group, two have majors in Computer Science, one a major in Industrial Design, one a major in Cognitive Science, and finally one has majors in both Computer Science and Voice Performance. All five members are working toward a second (or third) major in Human-Computer Interaction. Four of our five members have used Palms before, three members currently own Palms, and one had no prior experience with a Palm.

We employed Contextual Inquiry (CI) and Contextual Design (CD) to identify how a Palm might fit into one user's workflow and workplace, followed by Heuristic Evaluation (HE), Cognitive Walkthrough (CW), and Think Aloud (TA) usability evaluations which were performed on the existing PalmOS interface to assess current problems. Finally, we participated in an Interaction Relabeling (IR) exercise in order to expand

our perspective and explore different forms of device interaction.

Three of the evaluations we performed focused on the Palm's Date Book, and the Contextual Inquiry focused on memory aids but primarily revolved around scheduling. Hence, we selected as our cardinal target for a redesign the Date Book itself. We improved the event creation process in such a way that it alleviates a number of problems discovered in the Think Aloud analysis. We also evolved the default event display to be visual and clearly descriptive of one's schedule.

In addition, during the Heuristic Evaluation and Cognitive Walkthrough evaluations, it became apparent that our group is particularly cognizant of consistency issues. While there is not one particular application to deal with, we felt these issues consisted of a large enough portion of our final evidence that they deserved to be mentioned. We therefore devoted the rest of our attention to improving consistency and comprehensibility in several areas—namely issues with scrolling, labeling, and naming. We propose a hardware solution for scrolling and a renaming of Preferences and ShortCuts.

2. REDESIGN

2.1 Date Book

- 2.1.1 Date Book's Name
- 2.1.2 Day View
- 2.1.3 Creating a New Event
- 2.1.4 Setting Alarms
- 2.1.5 Synchronizing
- 2.1.6 Disabling Buttons & Menus
- 2.1.7 Remifications

2.2 Consistency & Comprehensibility

- 2.1 Scrolling
- 2.2 Labeling & Naming

2.3 Conclusion

2.1.1 Date Book **Date Book's Name**

Our Heuristic Evaluation indicated that users might have difficulty with the name “Date Book” (B.2.1 G3-HE-53). The ambiguity of the word can suggest both “Date” as in calendar time and “Date” as in a romantic meeting. This application, however, can be used for scheduling any kind of event.

We feel, therefore, that the Date Book should be renamed “Calendar” (A.2). This better describes the purpose of the application, and more appropriately matches the modern-day terminology for such programs. Many new cell phones include similar basic functionality, and the popular Sony Ericsson models include a “Calendar” under “Organizer.” The participant in the Contextual Inquiry study also referred to her paper scheduling reminders as a “calendar”

as well (B.3 L.15, L.19). Therefore this renaming is quite appropriate in the context of current devices.

One potential tradeoff to this is that users used to the “Date Book” name may become confused by the change, particularly if they have other applications installed which would cause it to appear in a different place in the application list. Other users may still not fully understand that a “Calendar” is used for scheduling events as well as simply displaying the current calendar day. We continue to call the application “Date Book” throughout this document to prevent confusion, but in screen shots of the applications list it has been changed to “Calendar.”

2.1.2 Date Book **Day View**

There is significant evidence that the default view of daily events (the day view) is rather confusing to users. The user in our Think Aloud analysis was attempting to find “7:00” in the actual day view, rather than simply attempting to create a new event with that time (B.2.3 G3-TA-01). He also misinterpreted the fact that he was setting up a meeting with a definite time span and wrote in the end time, rather than specifying that the meeting should end at that time using Date Book (B.2.3 G3-TA-13).

It follows in analyzing the Think Aloud study that the core of the above problems may be the result of a mismatch of conceptual models between the system and the user. Specifically, the Date Book

portrays start and end times for an event and shows the actual time period during which an event is taking place. Although we cannot read the user’s mind, it seems likely that the user’s confusion stems from the fact that the information is presented primarily through text rather than using visual cues.

Our proposed solution is to make the default day view more visually oriented and consistent with reality. Rather than simply being a screen of text input lines (Figure 1), it is a screen of blocks indicating when meetings are taking place (Figure 2). In our solution, the day view also defaults to containing all hours of the day, but with some hidden by a scroll bar. This is a closer match to what the user

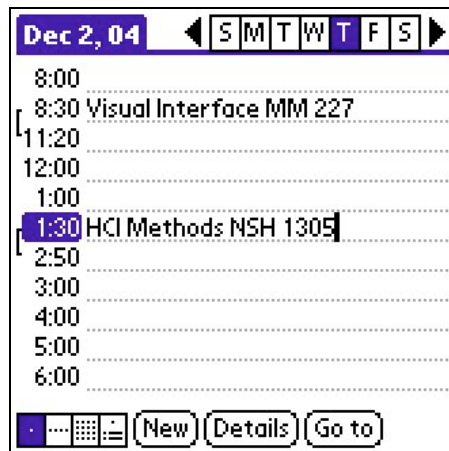


Figure 1

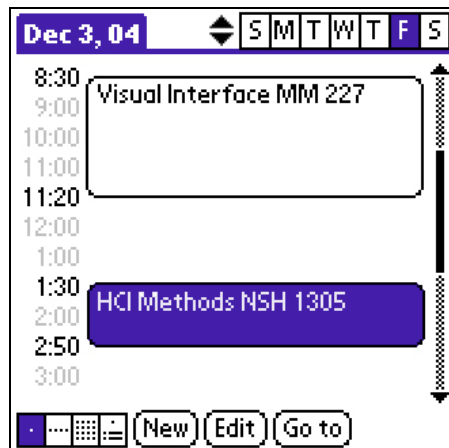


Figure 2

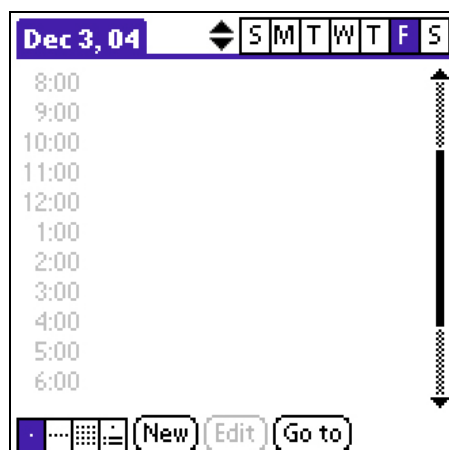


Figure 3

might be expecting, and would allow users to find “7:00” even if they do not realize they should tap “New” (B.2.3 G3-TA-01).

A day with no events would simply have times in a column down the left side (Figure 3), which is similar to the current implementation. The event times in our solution default to one-hour intervals as in the current implementation. Our proposed solution, however, does not have text input at every hour. While we recognize this as a convenient shortcut to inputting a new event, we feel that its efficiency is not worth the extra confusion it causes—the Think Aloud user has difficulty with disappearing events, not knowing how to create a new event, and not understanding time span (B.2.3 G3-TA-01, G3-TA-04, G3-TA-13). Therefore, in our implementation the user must tap “New” to create a new event in our solution.

This loss of really efficient event entry is the largest tradeoff. Users will now have to tap on “New,” then at absolute minimum two more buttons before they have created an event. While the current implementation is effective in this regard, the list of text entry areas was simply too confusing for our Think Aloud participant (B.2.3 G3-TA-01, G3-TA-04, G3-TA-13).

2.1.3 Date Book **Creating a New Event**

Set Time

Start Time: 12P↑ 00

End Time: 1 05

2 10

3 15

4 20

5 25

6 30

7 35

8 40

9 45

10 50

11 55

All Day

No Time

OK Cancel

Figure 4

New Event

Start Time: 12P↑ 00

End Time: 1 05

2 10

3 15

4 20

5 25

6 30

7 35

8 40

9 45

10 50

11 55

All Day

No Time

OK Cancel

Figure 5

We determined that the process of creating a new event could also be improved. Our Cognitive Walkthrough predicted that people might have difficulty using the time scroll arrows (B.2.2 G3-CW-02). In the Think Aloud usability study, we witnessed a particular user who had difficulties with certain aspects of creating an event. The user had great difficulty attempting to set an alarm for an event (B.2.3 G3-TA-02), and shows that he did not understand that he was creating an event with a specific duration (B.2.3 G3-TA-05). Instead, he entered text indicating when the event should end. In addition, the user experienced difficulty when he created an event without giving the previous one a name, causing it to disappear (B.2.3 G3-TA-04).

The first step of our solution involves changing the “Set Time” dialog one gets when one creates a new event, which can be done either by tapping on a time or tapping “New” in the Date Book (Figure 4). Our new dialog is, again, more visually oriented and linked to the real world (Figure 5). The dialog is called “New Event” clarify that one is setting the time for a new event. The time selection scrolling has been changed from irregular scroll arrows to a standard scroll bar so the user can clearly see that the times can be scrolled (B.2.2 G3-CW-02). Instead of displaying boxes which the user clicks on and edits, we have a tab-like system which unlike the current implementation invites interaction and provides a clear visual pagination. Once the user has set a start time, they tap the end time and set that. The start time and end time have larger boxes around them, indicating that they are very important things to be looking at in this dialog. This is one of our several attempts to make it clear to the user that

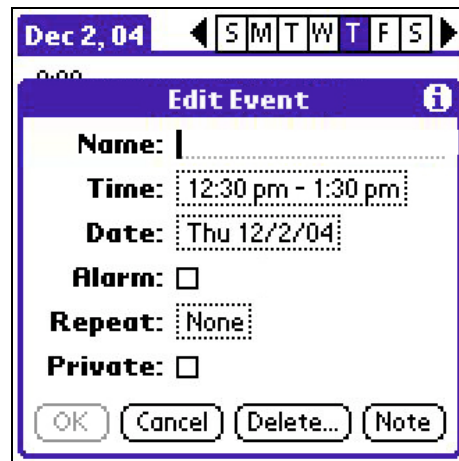


Figure 6

there is in fact an end time to be set, further resolving the difficulty in the user's understanding that events have durations.

Next in our solution, once the user has tapped "OK," the "Edit Event" (formerly "Event Details") dialog is presented with a new "Name" field (Figure 6). Using this method, the user is presented with all of the options when creating a new event at one time, rather than having to discover the options himself. The user should accordingly be able to discover the alarm feature during the process of creating an event because it is presented to the user, which Heuristic Evaluation and Cognitive Walkthrough would predict has a greater chance of being noticed. The change in method of creating a new event (in combination with the aforementioned day view changes) can also eliminate the possibility of the user creating a disappearing event by not allowing the user to tap "OK" in the "Edit Event" dialog until they have given the event a name.

The major tradeoff for our "New Event" (formerly "Set Time") dialog design is that the user would have a difficult time if he or she did not understand our tab-like system. However, we have reason to believe this is unlikely because tabs are a commonly used metaphor taken from physical tabbed folders and tabbed pages. Unfortunately, our visual layout of this "New Event" dialog does take up more screen space than the previous design, and they are not *exactly* tabs as seen in most operating systems. The other tradeoff is forcing the "Edit Event" dialog to appear every time the user wishes to create an event. The user will now have to dismiss another dialog before they are finished. We feel this is an acceptable tradeoff because people have to name events either way—this just simplifies the path where the user wishes to do more than that.

The Think Aloud usability study also demonstrated that some users may have quite a bit of difficulty figuring out how to set an alarm for an event (B.2.3 G3-TA-02). In addition, our Cognitive Walkthrough showed that the feedback for the alarm time was not sufficient for users to understand that they had set an alarm for the correct time (B.2.2 G3-CW-03, G3-CW-04).

We believe that our new event creation process already solves some of these problems because the user will be able to see that an alarm is available in the “Edit Event” dialog, which appears as a part of this process (2.1.3, A.4). Given that the purpose of the “Edit Event” (formerly “Event Details”) dialog is *editing* the currently selected event, the “Details” button should also be renamed “Edit” to further allay the problems finding this dialog as experienced by the Think Aloud user (B.2.3 G3-TA-02). Within the “Edit Event” dialog, we have added the word “prior” to the alarm time specification, as in

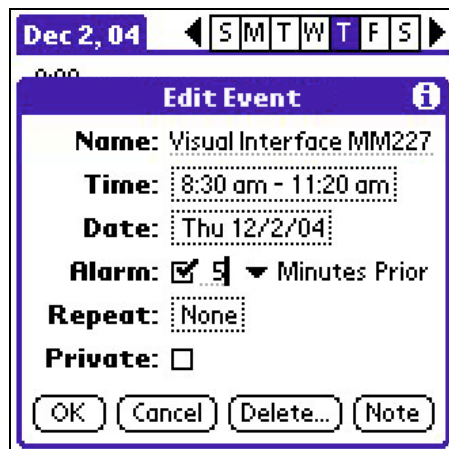


Figure 7

“Minutes prior” (Figure 7) in order to clarify that the alarm will happen *before* the event takes place, which helps our hypothetical Cognitive Walkthrough user complete the task (B.2.2 G3-CW-03).

These problems and the fact that the user in the Think Aloud spent a good portion of time navigating for an alarm-related application (B.2.3 G3-TA-02) prompted the idea of creating an entire application whose purpose it is to display upcoming alarms in order to alleviate this issue and provide another way of accessing alarm information for a user who has difficulty finding it in the Date Book (B.2.2 G3-CW-04). This would be a dynamic display of all alarms, populated from the Date Book. It would also be possible to create new alarms in it without a Date Book Event attached to them. Given that this specific design change is mostly unmotivated, further information can be found in Appendix A.5 Setting Alarms.

We cannot find many possibilities for confusion in renaming “Details” to “Edit,” except that existing Palm users may be slightly confused by the change. It is also possible that people will not realize they can *view* information as well as edit it by tapping “Edit.” For this reason an alternative solution would be to consider Mac OS X’s terminology: “Get Info.” We prefer the “Edit” terminology because it is most similar to what most computer users are used to—it fits in to the “New/Edit/Delete” schema of terminology.

During the Contextual Inquiry our group did an affinity diagram of Usability Aspect Reports and Contextual Inquiry breakdowns to help decide on focus points for further inquiry and a possible redesign. Based upon information obtained from the Contextual Inquiry interview, we identified synchronization and communication with other devices as a possible area of improvement (B.1). We decided that breakdowns in the Workflow Diagram (B.3.2, B.3 L.19, L.51, L.82, L.199), as well as possible insertion points in the workflow for the Palm (B.3.3), provided evidence for the addition of synchronization components to the Date Book.

In the Contextual Inquiry, the user's old Palm Phone lost all of her information several times (B.3 L.82). She complained specifically that the old Palm Phone did not synchronize with her Mac (B.3 L.19), and because this old Palm Phone did not synchronize with the user's Mac, it often contained duplicate information (B.3 L.51). The problem could be remedied by providing better synchronization support, particularly to a desktop computer such as this user's Mac.

The Palm does have synchronization capabilities in the form of HotSync. Unfortunately, it was not immediately obvious to our Think Aloud participant what exactly HotSync was for (B.2.3 G3-TA-14). On the Palm, HotSync takes the form of an application which a user can run occasionally when the Palm is plugged in and attached to a computer in order to synchronize Date Book, Address Book, and other information. The old Palm Phone probably did not synchronize with the Contextual Inquiry user's Mac (B.3 L.19) because it used a connection type (serial) that her Mac no longer supports.

Bluetooth is an interesting technology in this context. It is relatively cheap (see A.6 for price per chip), fairly easy to get, and popular in many modern phones, laptops, and PDAs (even in PalmOne's higher-end models, see A.6). We propose a solution, therefore, which supplements Palm's HotSync capabilities and takes advantage of Bluetooth when it is available. When users wanted to synchronize just their calendar or just their address book, they normally must go through quite a few steps in the HotSync software on the PC (see A.6). Inspired by this potential problem, we believe that synchronization of individual application's data should be readily available from the menu bar (Figure 8). This would allow users to quickly get data to and from various devices as they need it, rather than requiring that they wait for a full HotSync or synchronize this data via computer. This makes the Palm the center of the user's digital devices rather than requiring that their desktop fulfill that role, minimizing the redundant setup necessary for users that synchronize with multiple computers, for example in the office and at home.

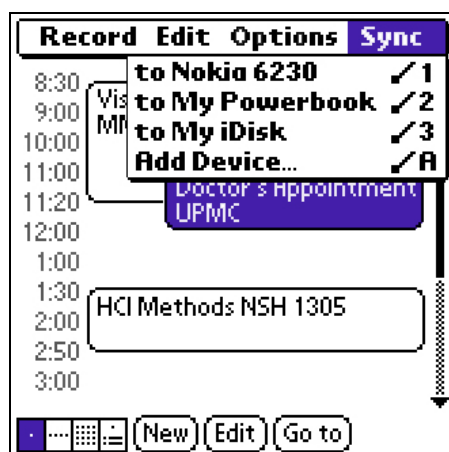


Figure 8

This is not an entirely motivated change, and it is not as worthy of attention as our other recommendations. But given the possibilities alluded to from our Contextual Inquiry, both in the breakdowns (B.3 L.19, L.51, L.82, L.199) and opportunities,

we feel that this would be an excellent opportunity for the creators of the next Palm interface to give users something they may not have realized they needed, but will not be able to live without.

2.1.6 Date Book Disabling Buttons & Menus

In our Heuristic Evaluation of the Palm interface, all of the members of our group discovered issues with buttons which in certain instances do nothing but present an error dialog. A user is provided with no visual cues that these buttons do nothing, and therefore may become confused when presented with an error dialog (Figure 9). This violates both the heuristics of “Error Prevention” and “Visibility of System Status” and wastes the user’s time. From our evaluation, we believe this is expected to happen frequently with the “Details” button and the text edit menu in the Date Book application (G3-HE-07, G3-HE-16).

Our proposed solution is that the “Edit” button and appropriate menu items be changed to grey when they will not do something useful (Figure 10). This is an

immediate indication to the user that nothing can be done by tapping the button. Windows, Mac OS X, and many other desktop environments disable buttons and menus in this fashion.

Taking this solution a step further, however, we recommend that the gray buttons actually *still* present the user with an error message when tapped (perhaps better worded by using “event” instead of “record”) to provide further user assistance. This would seem like a non-solution, but this actually helps the user figure out *why* something was disabled (see A.8), and is, hence, well-justified. The problem of figuring out *why* a button or menu item is disabled may be the reason the *Palm OS User Interface Guidelines* explicitly recommend against dimming or graying out buttons in favor of an error dialog. After reading the information given in this error dialog once, the user will know that tapping “Edit” when it is gray will only give them an error message. This solution prevents further error messages by giving immediate visual feedback. The tradeoff, and the major reason this probably was not implemented earlier, is that this will not work on two-color Palm devices. Two-color Palm devices do not seem to be sold anymore, however, so there is no reason this could not be implemented for all new devices.

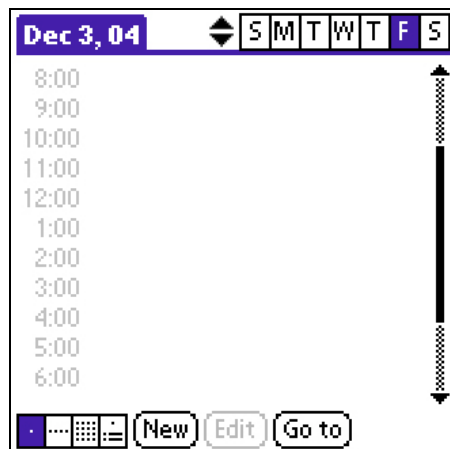


Figure 10

While these design changes were presented only within the context of the Date Book, some of them have ramifications beyond only this application and therefore should be addressed. Please refer to Appendix A.7 Ramifications of Synchronizing and Appendix A.9 Ramifications of Disabling Buttons & Menus for further information. If these ramifications are not taken into consideration, then the changes to the Date Book will make it inconsistent with other applications, which would be problem in itself.

Scrolling is a very common action on the Palm, particularly as web browsing and reading longer bodies of text become more common. Unfortunately, many of the implemented methods of scrolling are suboptimal and can even hinder the user's ability to perform the task. Users sometimes get scroll buttons and sometimes get scroll bars. At times scrolling is simply too slow, and at other times scrolling is too fast. Even the physical up/down buttons are not a complete solution (G3-HE-30).

In our Heuristic Evaluation and Think Aloud studies, we found evidence that scrolling was not as efficient or usable as it potentially could be. In some cases, scrolling is achieved through the use of scroll buttons when a scroll bar would be more appropriate and, coincidentally, more in line with what the *Palm OS User Interface Guidelines* would recommend for text (B.2.1 G3-HE-06). In other cases, the scroll buttons are too sensitive to be used accurately (B.2.1 G3-HE-06, B.2.3 G3-TA-12), and sometimes scrolling by tapping is the only option for the user when manual input would be more efficient (B.2.1 G3-HE-56).



Figure 11

We discussed quite a few possible solutions to the scrolling problem, including creating several new scrolling widgets and offering an entirely new set of guidelines for their use. In the end we decided that the current scroll bars are not *that* bad visually, given their size and space constraints, and that an easier solution would be a better method for the user to interact with the scrollbars.

Therefore, we propose that a physical jog wheel be placed on the side of Palm, similar to that seen in Figure 11. A lot of Pocket PC, Tablet PC, and even laptop devices are beginning to feature such hardware and it is not particularly difficult or expensive to build. Having a direct interaction in scrolling is a very powerful thing. Direct and effortless scrolling is the basis of the incredibly popular iPod's interface, and most mice feature scroll wheels for help with scrolling on desktop computers.

One potential problem with this is that some users may not associate the jog wheel with scrolling or may not notice the hardware immediately. Clear labeling or affordances on the physical device would help alleviate the problem. Some users may also be confused as to which object will be scrolled when multiple scrollbars or scroll arrows are available. This situation is harder to rectify and may require the creation of some kind of "focus" marking in the Palm interface, further complicating it and bringing it away from its more simplistic roots. It will also be difficult to decide which side to place the jog wheel on—the left side would seem appropriate for people writing with their right hand, but it may be hard for them to operate the wheel with their left thumb. In addition, any decision made favoring right-handed

users could potentially dissuade left-handed people from purchasing the Palm because it would likely be an awkward position for them. On the other hand, adding a jog wheel

may eliminate the need for the physical up/down buttons on the front of the Palm, freeing space for other potential buttons.

2.2.2 Consistency & Comprehensibility Labeling & Naming

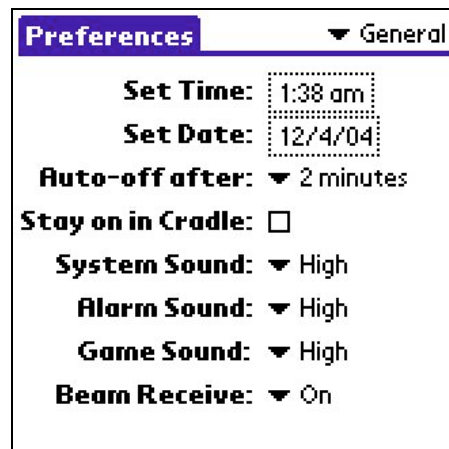


Figure 12

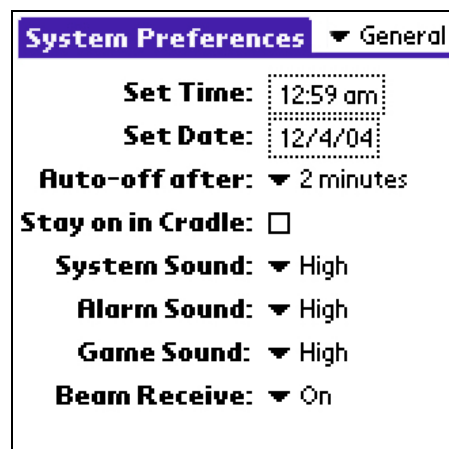


Figure 13

Our group found a lot of minor issues with labeling and naming items and objects throughout the Palm, which although minor when viewed individually, as a whole can have a very negative impact.

As mentioned in 2.1.1, we feel the Date Book should be renamed to “Calendar.” The other application we believe needs to be renamed is “Preferences.” The Think Aloud user did not understand what “Preferences” (Figure 12) was for and was confused enough as to what exactly it was he was doing in that application that he eventually rationalized it to be an alarm application (B.2.3 G3-TA-06, G3-TA-08). We believe that the name “System Preferences” (Figure 13) will much more clearly convey the purpose of the application to the user, as it indicates the context to which these preferences relate. It is the name Mac OS X uses for a similar purpose and it indicates that it is not preferences for any one particular application. The tradeoff is that the name is longer and may not fit everywhere. It seems to fit at least in a shortened form (“Sys Prefs”) everywhere necessary, however (Figure 14).

The To Do application features priority settings for each item (Figure 15). Our Heuristic Evaluation found that users would have difficulty understanding which priority number mapped to high priority and which mapped to low priority (B.2.1



Figure 14



Figure 15



Figure 16

G3-HE-56), so we propose adding labels as seen in Figures 16 and 17. There does not seem to be a tradeoff other than additional used screen space, but in the case of the pulldown menu it is only temporary anyway.

Within the Preferences there exists a page called “ShortCuts” (Figure 18). Our Heuristic Evaluation again indicated that users would have a tough time understanding what “ShortCuts” could be referring to (B.2.1 G3-HE-13). Renaming them to “ShortHands” (Figure 19) could partially solve this problem. The term “shorthand” is colloquially used to refer to abbreviated written text—which is exactly the functionality that “ShortCuts” provide.

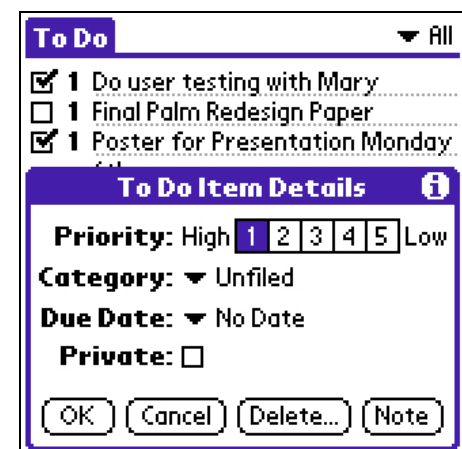


Figure 17

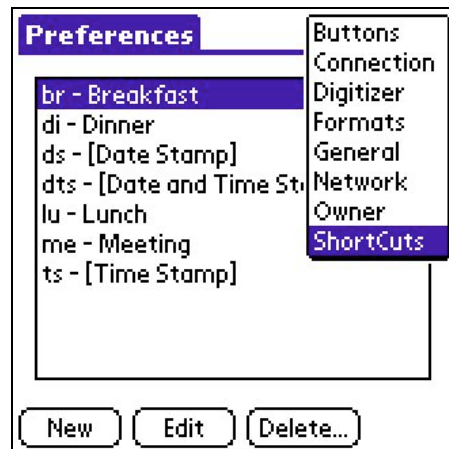


Figure 18

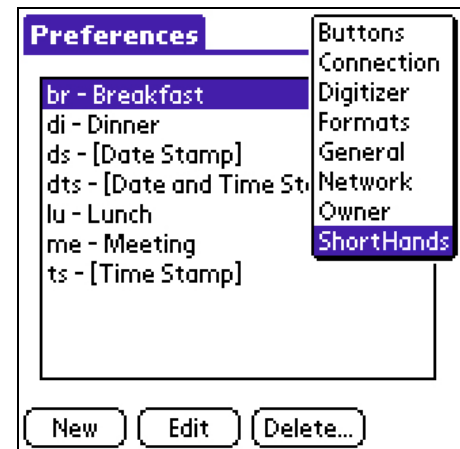


Figure 19

2.3 Conclusion

We have detailed in this document and its appendices a couple of large changes and numerous small changes to the Palm interface which we believe could improve the Palm experience significantly. The Date Book's default display was made visual, the event creation process was improved, and synchronization was suggested as a forward

step. A hardware solution was proposed to alleviate scrolling issues. Preferences and ShortCuts were renamed to clarify their purpose. Most of these changes are not horribly complicated or difficult to implement, and when implemented could completely change a user's perspective on the device for the better.

3. RETROSPECTIVE

3.1 Contextual Inquiry and Contextual Design (CI/CD)

3.2 Heuristic Evaluation (HE)

3.3 Cognitive Walkthrough (CW)

3.4 Think Aloud (TA)

3.5 Interaction Relabeling

3.6 Conclusion

The process of collecting and analyzing the data used to create our redesign was both a collaborative and iterative process. Using the techniques covered in class, we were able to draw from a variety of perspectives and approaches to achieve our final goal—to redesign the Palm such that both interactivity and productivity are improved for users. The techniques that we employed for this redesign were Contextual Inquiry, Contextual Design, Heuristic Evaluation, Cognitive Walkthrough, Think Aloud, and Interaction Relabeling. With foundations and support from empirical studies and research findings, each served our goal in a particular way, and, in some cases, served it quite effectively. The following is a reflection and evaluation on our experience with these techniques.

3.1 Contextual Inquiry and Contextual Design (CI/CD)

There is one fact that applies to all products, interfaces or otherwise: if it is not needed or wanted by any audience, it will not be used. Contextual Inquiry and Design (CI/CD), can be effective in addressing this issue—when used well, it can dramatically enhance a product’s usability *and* desirability. As a method that studies, analyzes, and reflects on the context in which a product is to be introduced, it is effective in discovering which components and processes would be most appropriate for a particular product given the context in which it is to be used. In addition, the empirical element of CI/CD—interviewing, documentation, and modeling—gives the evaluator a reality-based understanding of how a product would most effectively be integrated into the user’s workflow and work environment. In our case, it enlightened us to the fact that synchronization was a significant concern in scheduling for our user. This information could not have been recognized by any other technique that we have learned thus far, as its only evidence of existence is from the user herself in the context of her work.

Because of the organization and detailed specification of the models that are constructed in a contextual design, an evaluator can easily determine the ways in which a product would affect a user’s work by making specific modifications to these models. However, obtaining the level of details necessary to cover a representative amount of ground for this to work can require a significant amount of time and commitment. In fact, it can often be superfluous if the data received by different user interviews overlap.

This, however, is not always the case, and there are solutions to this problem. For example, an evaluator might attempt to save time by interviewing few users, which would work well when a user group’s task and work environment not highly variable. However, if a given user group works in an environment where these things are highly varied, the data collection of the CI/CD would need to reflect this to represent it effectively. For this reason, it seems that CI/CD is most appropriate in contexts with little variability in work tasks and the work environment; however in a diverse and multi-faceted environment where models cannot well-represent the breadth of the field, it may not be the most cost-effective solution.

There is something distinguishing that we feel must be mentioned about CI/CD, and that is that it is arguably the only technique that is *entirely* based in the real world. Unlike Think Aloud evaluation, which begins with the assumption that a user will be interested in completing a given task, CI/CD bases itself on no assumptions other than the ones given by the interviewer’s personal biases and field of study in which the CI/CD is used. This fact alone makes CI/CD irreplaceable. Also, we believe that the Interview portion of CI/CD is highly valuable when done by an effective interviewer, as it communicates a reality-based understanding of the user’s world, which can be enlightening, even to field experts. It seems, hence, that CI/CD is quite possibly the best way to demonstrate to an evaluator that, indeed, “The user is not like me.”

When evaluating an existing application, developers are often concerned with specific problems and benefits of an interface. Heuristic Evaluation (HE) is a cost-effective, time-saving solution which utilizes different evaluators to discover and clearly document interface aspects in the context of a set of pre-defined, research-based heuristics. Given the most appropriate number of evaluators, 4-5 people according to Nielsen (HE Slide 23), this is an effective technique to find a healthy breadth of technical problems with an interface, as "Different people find different usability problems", and "Successful evaluators may find both easy and hard problems". (HE slide 21) This fact is also supported by our experience, namely that 60 of our 68 UARs did not overlap between evaluators, and all 68 were relevant and applicable issues with the interface.

One could argue that different groups find different issues in HE, or even that diverse groups may be more effective evaluators for HEs than homogenous ones. However, we found no evidence that confirmed this. In fact, for our group, this was not entirely true. As a completely male group, the fact that we accumulated 60 unique UARs that covered 17 unique categories for the Palm interface (as determined by our affinity diagram, see B.1) could be an indicator that at least diversity of gender may not be a significant determinant in the breadth of usability problems found in Heuristic Evaluations (research, of course, would be necessary to confirm this).

Regardless, there is a downside to finding a large number of problems in HE-a lot of writing (and reading) must be done. UARs

can often be so long and detailed that the big picture of an interface's overall problems can be overlooked by dwelling in fine details. Moreover, as they are not based on user accounts, the UARs themselves may not necessarily be a true reflection of what happens in reality, begging the questions, "Does this problem really exist for users?" and "Does it matter?" The evaluators must provide their own answers with severity ratings, which are inherently biased by their experience and therefore can be very difficult to justify. In our case, the evidence we provided in our HE was contradicted by the performance of the Think aloud user on two occasions (G3-TA-07, G3-TA-09). Although this does not necessarily discount our findings, it does question their validity. It must also be point out that, to believe that the conclusions of a UAR are viable, it is necessary to believe that the underlying heuristic(s) upon which it is based are viable. Although there is scientific evidence for why they are what they are, a developer that does not believe in one of these Heuristics may not trust the UARs drawn from them. Nevertheless, we believe that Heuristic Evaluation, used in conjunction with Think Aloud, is a cost-effective way to evaluate an interface in the real world that allows for both observation and analysis, breadth and depth. To serve as an effective representation of an interface's strengths and weaknesses, we believe this is the most appropriate context in which it should be used.

Unlike Heuristic Evaluation which allows for the developer of an interface to get the breadth of an interface's strengths and weaknesses, Cognitive Walkthrough (CW) is a technique that serves a depth-focus well, by looking at specific representative tasks and answering questions about how the user would perform on portions of a given task for the interface. It requires the evaluators to describe the assumptions of a user's experience for a given task, and allows a means to express why an interface is or is not effective, based on these assumptions. When you have specific critical tasks that are important for an interface, and an effective way to determine the steps of these tasks, CW allows for a detailed understanding of a user's performance.

The negative side of this depth-focus is fairly obvious: a lack of focus on breadth. An evaluator will have difficulty seeing the big picture of an entire interface's performance with this technique. But another effect that is not quite as obvious is that this, like HE, is contrived from the evaluators' mind. Although we can find evidence for the user's assumptions in our CI/CDs, the answers to the questions given are based solely in the minds of the evaluators. As we place such a strong emphasis on, "The user is not like me" in this class, it is interesting we are studying this technique which, more or less, requires us to make up a hypothetical situation in which

we must decide exactly what the user would do, and justify it. Even with well-grounded assumptions, the problem of the "Designer's Blind Spot" does not go away with this technique of answering questions about what the user will or will not do in a given situation (the fact that our Think Aloud had occurrences that conflicted with our HE supports this as well).

Looking beyond this issue, the preparatory phase of CW can be incredibly valuable, as it requires a critical exploration and analysis of the user's background knowledge and the tasks that are important to an interface. The analysis, regardless of its reliability, requires the team to evaluate problems with the interface from the evaluators' perspective. Considering this, it is in a way useful in the same way that Heuristic Evaluation is useful, but rather than being useful in a breadth-based context, it is useful in the depth-based context of the task at hand. Also, for goal-oriented walk-up-and-use systems for which CW was originally designed (CW Slide 14), this sort of analysis certainly can be very beneficial, as it requires the evaluators to step in a new user's "shoes", and those "shoes" (i.e. the user's experience/assumptions) are based in CI data, at the very least. Overall, it seems this is one technique in which the process is perhaps more valuable than the end result, or, at least, more reliable.

Partially artificial and partially based in reality, there are a number of reasons why the Think Aloud (TA) method “may be the single most valuable usability engineering method” (Nielsen, 1993, Usability Engineering: p.195; TA Slide 4). Watching a user as he executes a task for a given interface and verbalizes his thoughts can be incredibly informative and surprising, assuming that his verbalizations correctly correlate to his mental processes (which may not always be the case, as with Mediated Processes as we discussed in class). Assuming the user is in fact *thinking aloud*, though, observing a user’s real world experience can reveal issues that you never knew were problems and show you how close or far your evaluation of the interface was.

This is not to say, however, that TAs are the end-all-be-all usability engineering method, as there is a reality that not all users are alike, and as such, not all TAs are representative of typical users. Furthermore, it is necessary to accumulate multiple TAs to know whether what one person did is more or less likely to occur for another person. Also, although the Think Alouds are reality based, the tasks that are given are *hypothetical*, and force the user to imagine that he or she is in a certain situation with a certain task. One could argue that, “The user is not like me” applies in this case too, meaning that the user’s performance in achieving a task based on a hypothetical situation may not be entirely what goes in on reality. One solution to this is to combine CI and TA into a Context-Based Think Aloud that would allow users to complete tasks that they are actually interested in achieving in the context in which they wish to achieve them. This would be the reality of a user

using the product in the most true form (of course, this would have tradeoffs, like not being able to control the task at hand).

Nevertheless, for normal Think Alouds, the task chosen and how it is phrased is something that must be carefully considered, as it is possible that confounding effects of choosing a given task may influence the evaluator’s perspective of the entire interface, when the task itself may be the source of problems more than the interface. As was stated in class about keyword searching, approximately twenty percent of certain populations think of the same concept with different words. If this is true, it is very possible that using the word “alarm” instead of the word “reminder” in the description of a task may have a dramatic effect on a user’s performance of completing the task (we recognize the context than keyword lookup, but the idea of people associating certain ideas with different words still applies). This goes to show that, similar to CI/CD, the evaluators’ results are limited to what and how the inquiring is done.

Regardless of these problems, as stated before, TA used in conjunction with HE is a cost-effective way to evaluate an interface in the real world that allows for both breadth and depth. As Think Alouds can sometimes lose sight of the big picture of the interface, since it is a task-specific activity, the broad reach of HE seems to be an appropriate counterbalance. Integrating the elements of TA and CI into one reality-based think aloud, though, may prove to be another technique that would be invaluable in seeing how an interface is used in the real world.

There is no question that the Interaction Relabeling activity in class required us to “think outside the box”—this, in fact, seemed to be the main purpose it served for us. In many ways, it is like the Brainstorming phase of a general designer’s process—the goal is to generate ideas with a bit of refining toward the end, tossing stuff around, sometimes literally, in a playful manner. We found that thinking about relating things that people already know how to do physically to the activities people do in software is a useful activity—certainly it bases the ideas of design on the reality of what users, or more generally, people, currently *do*. The creative presentations of other groups were also quite useful to us, as they helped us generate more ideas. For example, the idea of dragging the start and end time for an event was inspired by group with the Baby Toy suggesting a similar physical act for a different activity.

It was certainly refreshing to not have to think critically about the details of how certain ideas would work, but rather how to relate the physicality of an object to the ideas that are floating around. Although some may argue that this method should occur before data collection has happened because it is such an abstract brainstorm-like method, it seemed in our case especially appropriate to do after an extensive amount of data collection, as many of us found that we were bursting with data and needed a creative outlet to use that data constructively. The downside to this abstract

approach, clearly, is that it is not concrete, and sometimes your success depends on the object that you have and the ways in which you conceptualize that object—certainly an object with little physical functionality may require the mind to stretch quite a bit, as compared to an object with numerous physical features (this stretch of the mind is not necessarily a bad thing, though).

Incorporating the idea of working alone before working together, it seems that this activity could be more effective if every individual had their own object and generated their own thoughts for it, then in a group, came together to share their findings, then went on to this activity. The reasoning for this is based on the same reasons in that HEs are done individually before collectively: group think can sometimes overcome the sometimes incredibly unique and useful thoughts of the individual. In addition, as we were inspired by another group, this may be a great source of new inspiration. Inspiration, in fact, seems to be what this method is all about, and as no other method seems to have such an emphasize, we found this one to be valuable, if for no other reason than to explore. We can’t necessarily correlate our final redesign with all of our ideas from this activity, but on a strictly human level, we all agree that it was, indeed, valuable (and a great way to break away while staying in the same mode of thought!).

It is clear that all of the techniques taught have *some* sort of place in user interface evaluation and design, even if that place is not in every single aspect of the development cycle of a product. The issues that have been addressed, of reality vs. non-reality, breadth vs. depth, analysis vs. observation, and top-down vs. bottom-up approaches, at the minimum, all play significant roles in specifying when certain methods are more appropriate over others. As such, we believe that it could be very useful for this class if, either together or in groups, we were to define a list of criteria from these broader concepts, and categorize each of the techniques we have learned accordingly, in order for us to use as a reference in making good decisions on-the-job.

From the methods we used, the breadth-specificity of HE and the contextual reality of CI/CD were two aspects that we found to be invaluable, along with the user/task specification exploration of CW, the user-understanding of TA, and the out-of-the-box approach to Interaction Relabeling. Other methods and pieces of knowledge, like GOMS and the Model Human Processor (MPH) also shaped our redesign ideas, if for no other reason than that they served to influence our perspective and approach to interfaces, interface tasks, and user experiences.

In addition to the idea of using these methods at stages of a product's development lifecycle where they are well-complemented, we find that there are also strategic decisions that evaluators could make about which methods to use in combination. For example, CI + IR could form a good combination for radical redesign, whereas HE + TA (and maybe +

CW) could be a good combination for iterative development. Moreover, by placing emphasis on different techniques, one is, in fact, placing priorities on different elements of an interface's development, as each one of the techniques we have discussed places priorities on breadth/depth, context/analysis, top-down/bottom-up, and certain products may in fact be lacking in one approach over another.

An important lesson we learned is that most questions in HCI don't have easy answers, but there are techniques to understand why certain answers are more or less appropriate than others. Because of the practicality of these techniques and their relevance today as compared to ten years ago, though, we question how what we have learned will be viewed in ten years, or twenty even. With that in mind, it might have been useful to devote a class toward current HCI research and trends, and what techniques and methods are new and coming up in the foreseeable future.

That said, there is one last thing that we wish to address that is of great importance to us, and that is that the opportunity to meet early in the semester and get group-interaction handouts had a profound positive impact. Before that time, we were struggling as a group, on a number of levels, and two of our members were at great odds with each other because of a past history. After following the steps described in the handouts we received in talking and communicating with each other, the problem disappeared, seemingly overnight, and we became a much improved group because of it. It has gone to show that, for us, there is no better learning than learning the hard way.

APPENDIX A: Design Changes

- A.1 Overview Table of Design Changes
- A.2 Date Book's Name
- A.3 Day View
- A.4 Creating a New Event
- A.5 Setting Alarms
- A.6 Synchronizing
- A.7 Ramifications of Synchronizing
- A.8 Disabling Buttons & Menus
- A.9 Ramifications of Disabling Buttons & Menus
- A.10 Scrolling
- A.11 Labeling & Naming

A.1 Overview Table of Design Changes

Name	Motivation	Redesign Idea	Tradeoffs
Date Book's Name	CI L.15 CI L.19 G3-HE-19 G3-HE-53	Change "Date Book" to "Calendar"	Users used to "Date Book" may be confused, users may not realize "Calendar" is used for scheduling
Day View	G3-HE-59 G3-CW-03 G3-CW-04 G3-TA-01 G3-TA-04 G3-TA-13	Events are viewed as blocks, all hours of day are displayed, events are created by tapping "New" button	Cannot input text next to event hours, loss of efficiency
Creating a New Event	G3-CW-02 G3-TA-02 G3-TA-04 G3-TA-05 G3-TA-13	Events must be created by tapping "New" button, change "Set Time" dialog, "Edit Event" dialog follows "Set Time" dialog	Users unfamiliar with tabs may have trouble, loss of efficiency
Setting Alarms	G3-CW-03 G3-CW-04 G3-TA-02	Alarm is visible in "Edit Event" dialog, change "Details" button to "Edit" button, add "prior" to alarm time specification	"Edit" may confuse users familiar with "Details"
Synchronizing	CI L.19 CI L.51 CI L.82 CI L.199 CI Workflow G3-TA-14	Create "Sync" menu in Date Book	Takes up screen space, users familiar with old setup will not be used to a new menu
Ramifications of Synchronizing	CI L.19 CI L.51 CI L.82 CI L.199 CI Workflow G3-TA-14	Create "Sync" menu in applications that use HotSync	Takes up screen space, users familiar with old setup will not be used to a new menu

A.1 Overview Table of Design Changes (con't.)

Name	Motivation	Redesign Idea	Tradeoffs
Disabling Buttons & Menus	G3-HE-07 G3-HE-16	Disable inactive buttons and menus in Date Book	Only possible on Palms that display more than 2 colors
Ramifications of Disabling Buttons and Menus	G3-HE-07 G3-HE-08 G3-HE-16 G3-HE-17 G3-HE-40 G3-HE-51	Disable inactive buttons and menus in all applications	Only possible on Palms that display more than 2 colors
Scrolling	G3-HE-06 G3-HE-30 G3-HE-56 G3-TA-12	Add physical jog wheel to Palm	Users may not associate wheel with scrolling, can be confusing when there are multiple items to scroll
Labeling & Naming	G3-HE-13 G3-HE-23 G3-HE-56 G3-HE-63 G3-HE-66 G3-TA-03 G3-TA-06 G3-TA-08 G3-TA-10	Change “Preferences” to “System Preferences” Add labels to priority settings Change “Shortcuts” to “ShortHands”	Longer names take more screen space



Figure A.2-1: Date Book's current name



Figure A.2-2: Date book as "Calendar"

Date Book's Name

MOTIVATION

B.2.1 G3-HE-19 Date book name – The name "Date Book" does not match functionality of program

B.3 CI L.15, L.19 – Participant refers to her "paper calendars"

Palm7amTaskPart1.mov 01:10-01:16 – "I need to find the calendar so I can put down the time for the meeting"

PROBLEM

Users may not understand exactly what the Date Book is for, particularly since it is not only for scheduling dates but any kind of event. A "Date Book" could also suggest that it is only for storing specific days, such as a birthday or holiday. However, the application provides much more functionality than that as it can also store more details as a calendar would, such as allowing the user to specify multiple events, as well as specific times for each of these events.

REDESIGN SOLUTION

We feel that the Date Book should be renamed "Calendar." This is a name that reflects the real world much better than "Date Book". Also, in the Contextual Inquiry the participant referred to her scheduling materials as "paper calendars," and in the Think Aloud the user stated that he needed to find the "calendar." Since "calendar" is a more

commonly used term and since users appear to refer to its functionality as a calendar, we feel that it is a much better name.

TRADEOFFS

Existing Palm users may be confused by the change in name initially and may expect that it is a program different from the “Date Book”. Other users may still not fully understand that a “Calendar” is used for scheduling events as well as simply displaying the current calendar day.

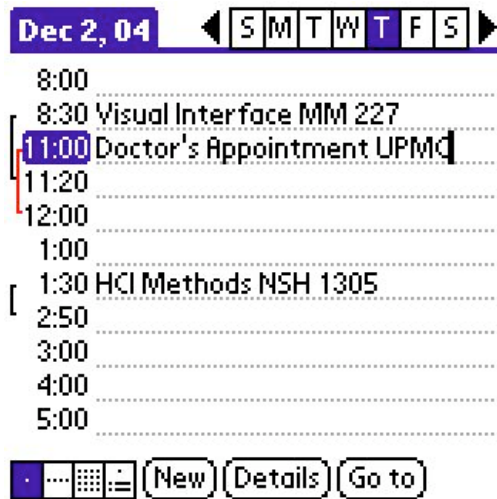


Figure A.3-1: Date Book's current Day View

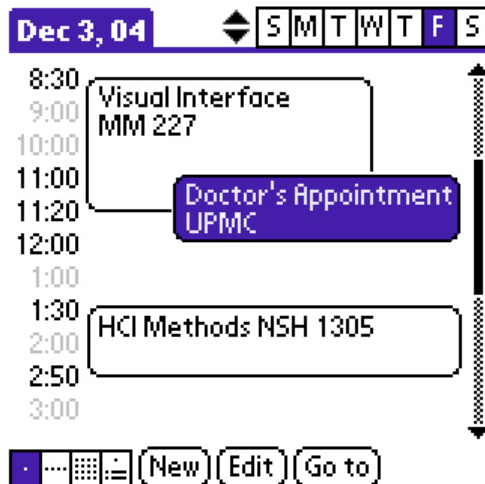


Figure A.3-2: Proposed Day View

Date Book's Day View

MOTIVATION

G3-HE-59 Arrow confusion – In “Date Book”, arrows at top surround s/m/t/w/f/s list, when their functionality is to change the month
 G3-TA-01 User has difficulty finding 7:00 in Event list

G3-TA-04 “6:50” time without label disappears when new time given

G3-TA-13 Date Book's meeting time span misinterpreted and difficult to modify

PROBLEM

The Date Book is intended to portray both start and end times for events. When a user creates an event, he or she is creating an object with these start time, end time, and alarm properties. Within the current interface, the user does not seem to understand this and simply makes a note at a certain time.

The Think Aloud user also has difficulty finding 7:00. He does not realize that he should be tapping “New” to create an event for a time that does not appear on the list.

The left and right week-changing arrows at the top are also difficult to understand. The evaluator thought initially that they might change the current day, when in fact they change the week.

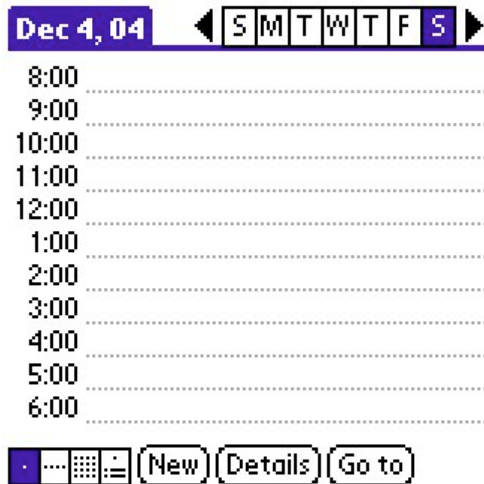


Figure A.3-3: Current Day View with no events

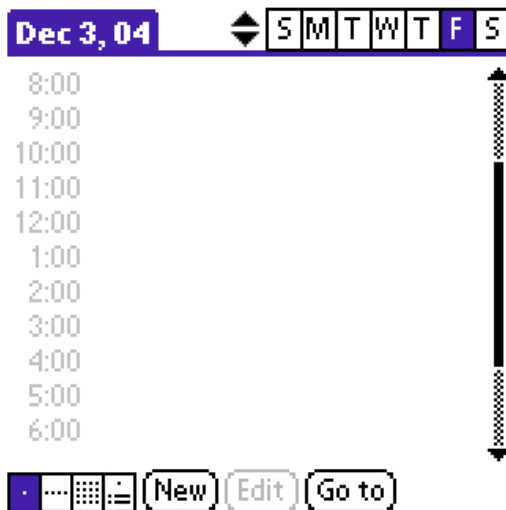


Figure A.3-4: Proposed Day View with no events

REDESIGN SOLUTION

Our proposed solution involves making the day view visual. Rather than being a screen of input lines (Figure A.3-3), it is a screen of blocks indicating when events take place (Figure A.3-2).

The day view defaults to containing all hours of the day, but hidden by a scroll bar. Note that it is a regular scroll bar, not scroll arrows.

The time text is gray unless it is the start or end time for an event, in which case it is black. This highlights the important times while leaving all times visible.

“Edit” has been renamed “Details” as described in A.5.

Events can be selected by tapping and appear in blue when selected. “Edit” becomes enabled when an event is selected, but disabled when no event is selected, as described in A.8.

The week-changing arrows have been changed to be vertical instead of horizontal. This more closely resembles a change in weeks on a calendar.

TRADEOFFS

Loss of really efficient text entry is the largest tradeoff. Users will now have to tap on “New,” and then at absolute minimum two more buttons before they have an event. The text input entries as they are in Figure A.3-3 are simply too confusing to new users.

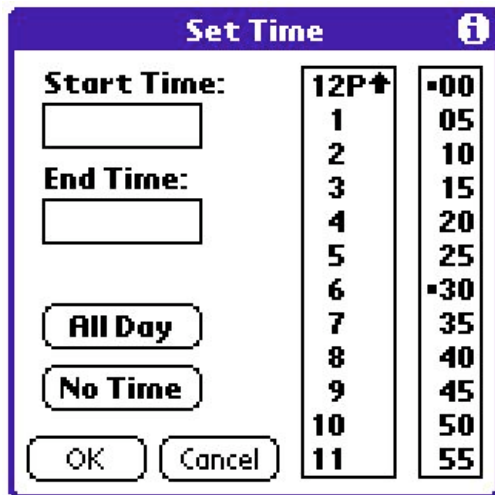


Figure A.4-1: Date Book's current new event dialog



Figure A.4-2: Proposed new event dialog

Creating a New Event

MOTIVATION

G3-CW-02 User will not associate a single tap on “Set Time” up arrow with paging up

TA-02 User cannot figure out how to set an alarm for an event

TA-04 “6:50” time without label disappears when not given name

TA-05 User does not see that end time can be set in Set Time dialog

TA-13 Date Book's meeting time span misinterpreted and difficult to modify

PROBLEM

Our Cognitive Walkthrough predicts that users will have difficulty understanding the scroll arrows.

The Think Aloud user had difficulty setting an alarm for an event. The Think Aloud user created an event and it disappeared because he did not name it.

The Think Aloud user did not realize that events have start times and end times, and failed to correctly set an end time.

REDESIGN SOLUTION

The first step of this solution is the proposed day view (A.3), which helps prevent the disappearing-event error.

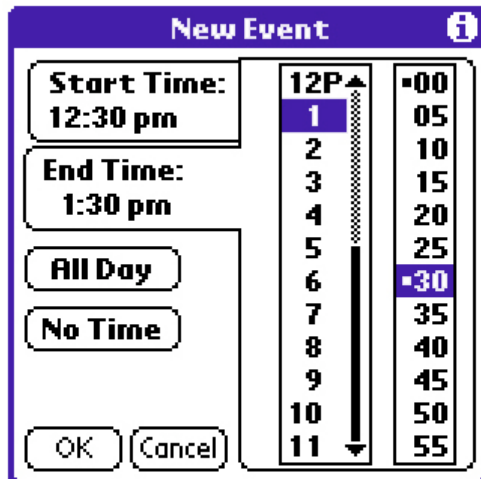


Figure A.4-3: Start time and end time selected

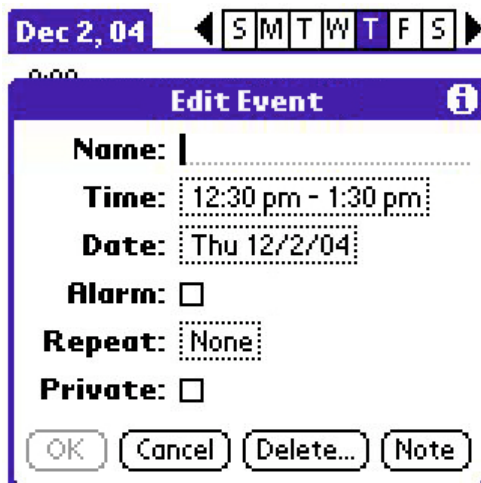


Figure A.4-4: Edit Event dialog

The second step is a revised new event dialog. It is now called “New Event” rather than “Set Time” so the user understands what the dialog is setting. The time scroll arrows have been changed to a more standard scroll bar so the user understands that it is to be scrolled. The dialog uses a tab-like system which invites interaction and provides a clear visual pagination. Using these visual cues, the user should more easily see that there is an end time to be set.

The final step is a revision in the event creation process. After the user selects a time (Figure A.4-3), they tap “OK” and are presented with the “Edit Event” (formerly “Event Details”) dialog (Figure A.4-4). This is where the user can enter the event’s name and see things such as alarms (A.5). The user must enter a name for the “OK” button to become enabled (A.11).

TRADEOFFS

The largest tradeoff is that the user must now go through our multi-step process of dialogs in order to create an event. This is acceptable to us because people have to have to name events either way—and now the paths in which the user wishes to set something other than the name are slightly more optimized.

The other problem would be if the user does not understand our tab-like system. The tab-like system does take up a bit more screen space and may not be entirely clear to the new user.

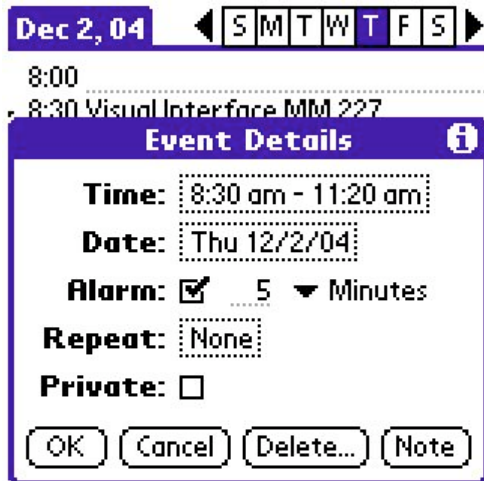


Figure A.5-1: Original Event Details dialog

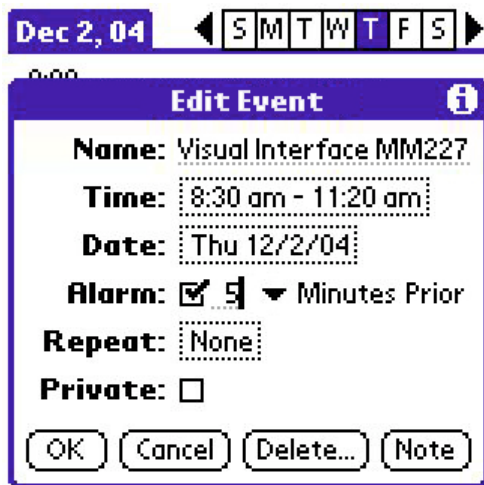


Figure A.5-2: New Edit Event dialog

Setting Alarms

MOTIVATION

G3-TA-02 User cannot figure out how to set an alarm for an event

G3-CW-03 Vague specification of Alarm time in Event Details

G3-CW-04 Insufficient feedback for Alarm time in Event Details

PROBLEM

Users may not understand exactly what the Date Book is for, particularly since it is not only for scheduling dates but any kind of event. A "Date Book" could also suggest that it is only for storing specific days, such as a birthday or holiday. However, the application provides much more functionality than that as it can also store more details as a calendar would, such as allowing the user to specify multiple events, as well as specific times for each of these events.

REDESIGN SOLUTION

We provided a few solutions to this problem. When creating an event, the user is automatically given all of the options that the formerly named "Event Details" dialog would initially provide. This means that if a user is to create a meeting then he will know that the reminder functionality is available because it is asked on-screen as he creates the event (see figure A.5-2).



Figure A.5-3: Modified application screen

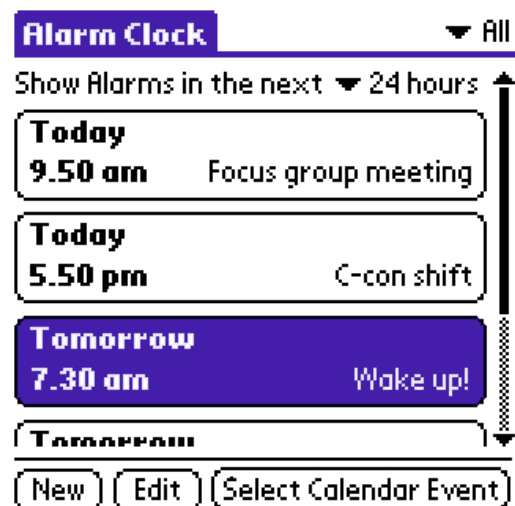


Figure A.5-4: Managing alarms

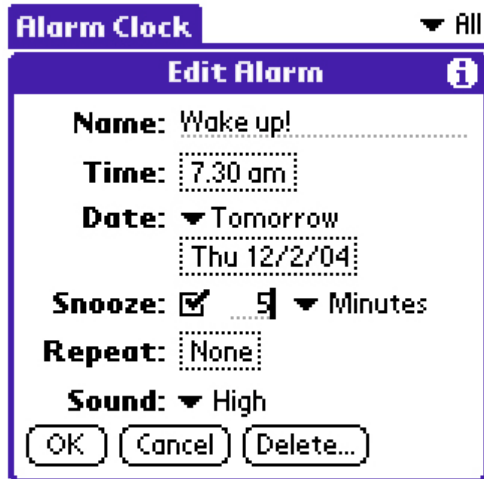
To address the problem of the inherent vagueness in the current implementation of setting an alarm, we decided to clear any ambiguity by specifying exactly what the alarm time meant. Since it is really simply to state how many minutes prior to the meeting you would like the alarm set, we added the label “Prior” to the end (see Figure A.5-2).

Also, in the case that the user does not think to create an event when setting alarm, we believe that there should be a separate alarm application included within the Palm which we decided to call “Alarm Clock” (see Figure A.5-3). Its abilities would include managing all of the alarms created in the Palm (see Figure A.5-4), as well as allowing the user to create an event-independent alarm (see Figure A.5-5). This would allow the user to quickly view and edit all the alarms and reminders that he has set without having to search throughout the Calendar. With the application, the user can simply create an alarm without the need for an event, for example if he wishes to have an alarm to wake up (see Figure A.5-5). Because we decided to create a separate application for an alarm, we were also able to add new features to the alarm, including a Snooze feature, the ability to specify the alarm sound, and the ability to create repeating alarms (see Figure A.5-5). Since this is a primarily unmotivated change, with the exception of some influence from the Think Aloud, some Contextual Inquiries/Designs would need to be done to ensure that this application has its place in the workflow.

TRADEOFFS

In the Date Book, the user has to navigate through one more dialog box than he previously had to in order to create an event. This tradeoff could be potentially significant for specific groups of users, particularly those who always have meetings from 8-6 in one-hour increments at the start of the hour and never use alarms, but we believe that it is worth it to require the users to set these fields initially as it solves this and many

With the implementation of a new Alarm application, one more application in the main screen is added, which causes the user to look at one more application than before. This is a particular problem for existing users who had previously become accustomed to specific



The screenshot shows a handheld device screen with an 'Alarm Clock' application. A purple header bar at the top contains the text 'Alarm Clock' and a small downward arrow next to the word 'All'. Below this is a dialog box titled 'Edit Alarm' with a purple title bar and a small information icon on the right. The dialog box contains the following fields and controls:

- Name:** A text field containing 'Wake up!'.
- Time:** A time selection field showing '7:30 am'.
- Date:** A date selection field showing 'Tomorrow' and 'Thu 12/2/04'.
- Snooze:** A checkbox that is checked, followed by a numeric field showing '5' and a dropdown menu set to 'Minutes'.
- Repeat:** A dropdown menu set to 'None'.
- Sound:** A dropdown menu set to 'High'.
- At the bottom are three buttons: 'OK', 'Cancel', and 'Delete...'.

Figure A.5-5: Creating a new alarm

locations in the screen for specific applications. The addition of this “Alarm Clock” would cause everything to shift by one more and therefore require some adjusting on the part of the user. More memory would be necessary to add the application and its functionality, which could potentially be very costly for Palms with limited memory.

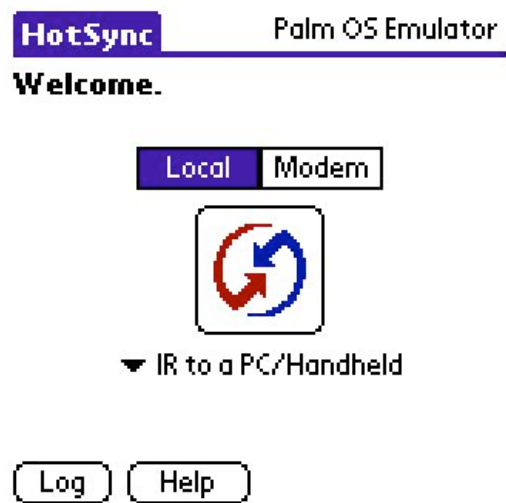


Figure A.6-1: HotSync

Synchronizing

MOTIVATION

- B3.1 CI L.19 – Participant’s old Palm Phone did not back up to Mac
- B3.1 CI L.51 – Participant’s old Palm Phone did not communicate with Mac but stored duplicate info
- B3.1 CI L.82 – Participant’s old Palm Phone lost all info a couple of times
- B3.1 CI L.199 – Participant has error synchronizing with Mac when phone not set up
- B3.2 CI Workflow – Participant uses excel to record contacts rather than an application designed specifically for this task

PROBLEM

There were multiple problems surrounding HotSync (Figure A.6- 1). By making HotSync a separate application, some users may not even be aware that HotSync exists, or which applications are actually synchronized. Making HotSync a separate application is inefficient and can waste the user’s time. Currently, in order to synchronize only Date Book data, the user must first exit Date Book and enter HotSync. The user must then go to their PC or Mac and set the synchronization preferences so that only the Date Book data is transferred. This can be a problem for many users who do not know how to change the settings.

If the user wishes to set the synchronization preferences from the Palm itself, the user must access HotSync’s menu and select “Conduit

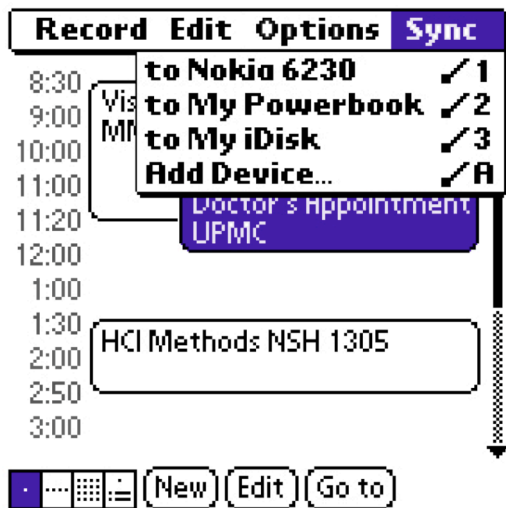


Figure A.6-2: Sync menu

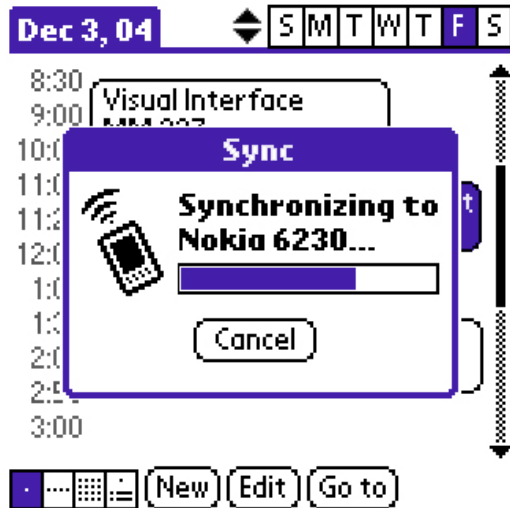


Figure A.6-3: Synchronizing to a device

Setup". From the next screen, the user must then uncheck the box next to the application they do not wish to synchronize. If the user wishes to only sync data from the Date Book, this can be a long, time consuming process. Many users may also have trouble with the term "Conduit" and not understand that the "Conduit Setup" screen is where syncing with an application can be disabled. Setting the synchronization preferences only works when connecting with the modem, which is another problem in itself. Some of the methods of HotSyncing, such as the modem, are outdated.

The Contextual Inquiry participant tried to use Bluetooth when setting up her phone, but would not be able to synchronize to a current Palm that way. The current incarnation of the Palm cannot connect to devices other than a PC or Mac, which limits the Palm's ability to communicate and synchronize. Many users, such as the CI participant, have multiple devices that would benefit from the ability to synchronize with a Palm.

REDESIGN SOLUTION

To make syncing Date Book data more efficient, we created a separate dropdown menu entitled "Sync" (Figure A.6-2), complete with Graffiti® shortcuts, in Date Book. This allows the user to sync directly from the Date Book application and would eliminate the need to go through the process listed above for syncing data. We also wish to expand the number and kinds of devices to which the Palm can synchronize, so we would also incorporate the ability to sync with devices besides a PC or Mac. Once the "Sync" menu is selected, it will display "To [Name of device]". If the user wishes to synchronize to another device not listed, there is an option at the bottom of the list to add another device. HotSync would also be supplemented by newer technology, such as Bluetooth, which would further expand the Palm's ability to synchronize to devices and potentially eliminate some of the older methods of syncing.

BUSINESS JUSTIFICATIONS

Newer high-end Palm devices already contain Bluetooth chipsets (<http://www.palmone.com/us/wireless/bluetooth/>) and Bluetooth chipsets are

only getting cheaper—Philips sells high-volume Bluetooth chips for less than \$3 each (<http://www.commsdesign.com/showArticle.jhtml?articleID=21700471>). A lot of modern phones (such as Sony Ericsson's popular models) have Bluetooth synchronization capabilities, most Apple Macintosh computers ship with the option of built-in Bluetooth, some PC laptops can be found with built-in Bluetooth, and Bluetooth USB modules are relatively cheap and easy to use.

It is also worth noting that Apple has indicated that individual application-based synchronization may be the future of synchronization on their platform as well, as is evidenced by providing the "Sync Services" framework as a compliment to iSync (<http://www.apple.com/macosx/tiger/sync.html>).

TRADEOFFS

Creating a new menu item for syncing will take up screen space. With more menu items, searching for a desired function will take more time since there are more items to search as well as increasing the number of distracters. This would also be different for users who are familiar with the current Palm and may take time for them to become adjusted.

Adding the ability to sync with other devices besides a PC or Mac would require the addition of a device manager of some kind that could handle options when syncing and allow for unwanted devices to be removed. At the very least, the HotSync application would have to be changed so that it could manage syncing with the new devices, but it is more likely that a new device manager application would become necessary. Syncing with devices, such as another Palm, would also require more software to be written for both the Palm and the other device.

Currently, PCs and Macs handle syncing issues, but adding the ability to sync to other devices would require the Palm to handle these issues as well. Integrating technology such as Bluetooth would also require additions to the hardware, namely Bluetooth chipsets. This would shorten the battery life and raise the cost of a Palm, but only marginally so.

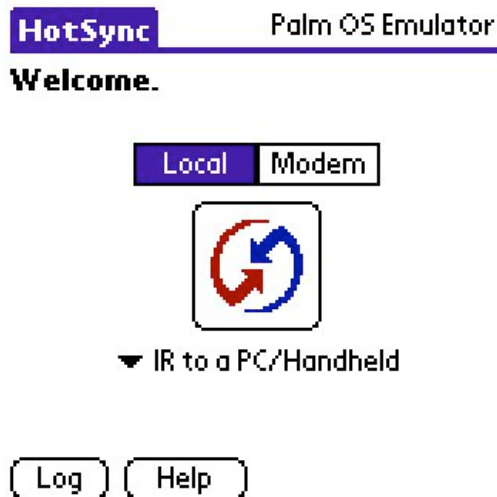


Figure A.7-1: HotSync

Ramifications of Synchronizing

MOTIVATION

B3.1 CI L.19 – Participant's old Palm Phone did not back up to Mac

B3.1 CI L.51 – Participant's old Palm Phone did not communicate with Mac but stored duplicate info

B3.1 CI L.82 – Participant's old Palm Phone lost all info a couple of times

B3.1 CI L.199 – Participant has error synchronizing with Mac when phone not set up

B3.2 CI Workflow – Participant uses excel to record contacts rather than an application designed specifically for this task

PROBLEM

Adding HotSync functionality to only the Date Book would violate consistency and standards for the Palm.

REDESIGN SOLUTION

HotSync functionality would be added to any application that transfers data to another device, thereby preserving consistency. These applications would be altered in the same manner as the Date Book application, with a new menu for syncing.

TRADEOFFS

Record	Edit	Options	Sync
Cho, Lily	to Nokia 6230	✓1	
Dunn, Hug	to My Powerbook	✓2	
Homer, D	to My iDisk	✓3	
Kesler, Jo	Add Device...	✓A	
Mellon, Jessica	412-563-2453 W		
Rosenberg, Jessica	320-568-3435 W		
Stadelmier, Steven	412-684-2489 W		
Tong, Calvin	412-243-5458 H		
Ulkucu, Emre	412-684-3910 H		
Yee, Eugene	707-482-4569 W		

Look Up:

Figure A.7-2: Sync menu in Address Book

Record	Edit	Options	Sync
<input type="checkbox"/> 1 Do u	to Nokia 6230	✓1	
<input type="checkbox"/> 1 Find	to My Powerbook	✓2	
<input type="checkbox"/> 1 Pos	to My iDisk	✓3	
6th	Add Device...	✓A	
<input type="checkbox"/> 2 Study for HCI Methods Final			
<input type="checkbox"/> 3 Buy milk at Giant Eagle			
<input type="checkbox"/> 3 Go talk to John at NSH 3537			
<input type="checkbox"/> 4 Sketches for Camera Phones			

Figure A.7-3: Sync menu in To Do

As with adding a new sync menu to the Date Book, adding a new menu to other applications will also mean taking up more screen space. Again, with more menu items, searching for a desired function will take more time since there are more items to search as well as increasing the number of distracters. Users familiar with the old Palm applications would not be immediately familiar with the new menu and would take time to adjust. Since the menu would be added to all applications that currently sync, Palm would also have to make an addition to their consistency and standards to include a sync menu in any application that syncs.

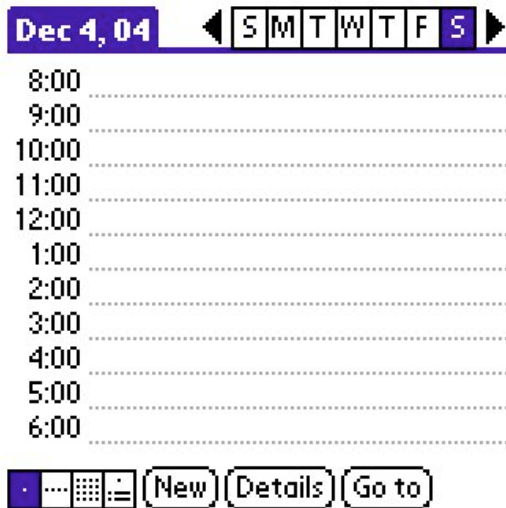


Figure A.8-1: Date Book without "Details" disabled

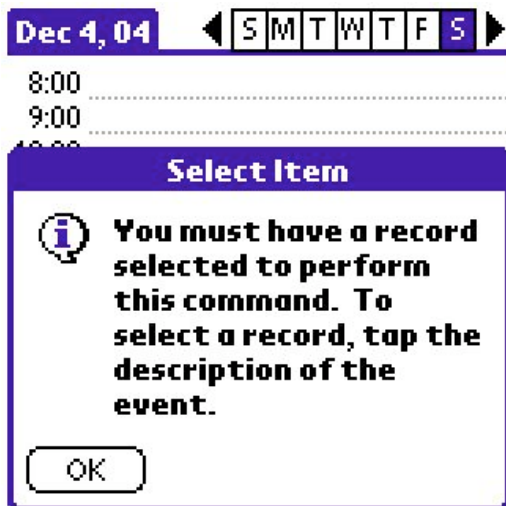


Figure A.8-2: The error message when one taps "Details"

Disabling Buttons & Menus

MOTIVATION

B.2.1 G3-HE-07 Date details - Details button in Date Book always enabled

B.2.1 G3-HE-16 Edit items enabled - Text "Edit" menu items are not disabled even when they do nothing

PROBLEM

If the user tries to tap on a button in the Date Book that does not have an active function, such as the "Details" button, the user is presented with an error dialog. This may confuse and frustrate many users, as well as waste time. This violates Heuristic 5: Error Prevention. When an inactive menu item is selected, such as "Undo" when nothing has been done yet, the system does not display any active feedback. The user cannot know if the system actually performed the action or not. This is a violation of Heuristic 1: Visibility of System Status.

REDESIGN SOLUTION

Any inactive button or menu item should be "grayed out", indicating to the user that the button or menu item cannot be used. This technique is commonly used in Windows, Mac OS X, and many other desktop environments. If the user does tap on the grayed out item, the user will be then presented with a dialog informing the user why the button or menu item was grayed out (<http://www.asktog.com/Bughouse/>)

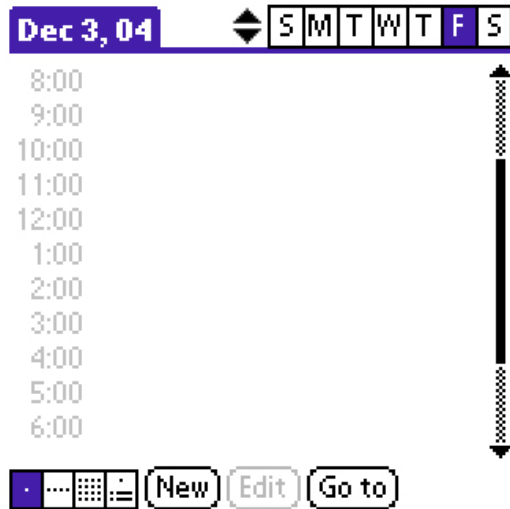


Figure A.8-3: "Edit" button disabled

10MostPersistentBugs.html, bug three). This will help the user understand why the button or menu is disabled. Once the user has performed an action or selected an item that would make the button or menu item active, the appropriate button or menu item would then turn black and become enabled.

TRADEOFFS

This will only be possible on Palms that can display more than two colors. However, since most Palms currently manufactured feature color displays, this should not be a problem. The dialog that is displayed when a user tries to tap on a grayed out item must also be created and implemented, but the current error dialogs that appear whenever a user taps on an inactive button can be used after a few minor changes in wording.

Security

Current

Privacy: ▼ Show Records

Password: -Unassigned-



Figure A.9-1: "Lock & Turn Off..." disabled

Preferences

▼ Connection

Available Configurations:

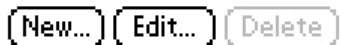


Figure A.9-2: "Delete" disabled

Ramifications of Disabling Buttons & Menus

MOTIVATION

B.2.1 G3-HE-07 Date details - Details button in Date Book always enabled
 B.2.1 G3-HE-08 Todo details - Details button in To Do List always enabled
 B.2.1 G3-HE-16 Edit items enabled - Text "Edit" menu items are not disabled even when they do nothing
 B.2.1 G3-HE-17 Duplicate address - "Duplicate Address" not disabled when nothing selected
 B.2.1 G3-HE-40 Lock error prevention - Clicking "Lock & Turn Off..." results in an error page
 B.2.1 G3-HE-51 Connections enabled - In "Preferences" under "Connection", "Delete" button is enabled when clicking it serves no function

PROBLEM

Besides the Date Book, there are other applications that contain buttons and menu items that should be disabled. Only changing buttons and menus in Date Book would be inconsistent.

REDESIGN SOLUTION

As with the Date Book, any other button or menu item in any application that is not disabled when it should be will be grayed out when inactive. Disabled buttons and menus that are tapped would present dialogs that explain why the item



Look Up:

Figure A.9-3: Text edit menu items disabled

is disabled (<http://www.asktog.com/Bughouse/10MostPersistentBugs.html>, bug three).

TRADEOFFS

This is only possible on Palms capable of displaying more than 2 colors, but since most Palms currently manufactured feature color screens, this should not be an issue. Dialogs for menu items would have to be written but the dialogs that display when buttons are tapped can be preserved. Palm would also need to add disabling buttons and menu items to their consistency and standards.

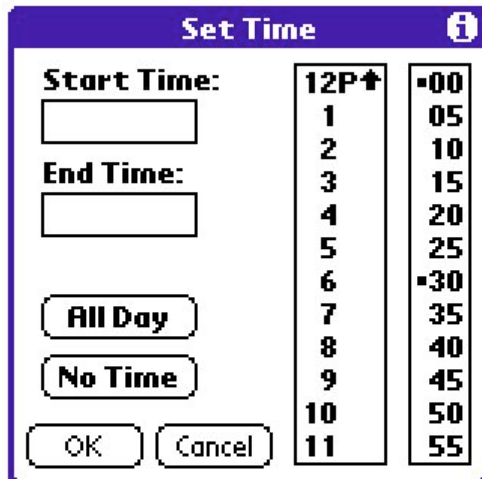


Figure A.10-1: Set event time



Figure A.10-2: Date Book Day View

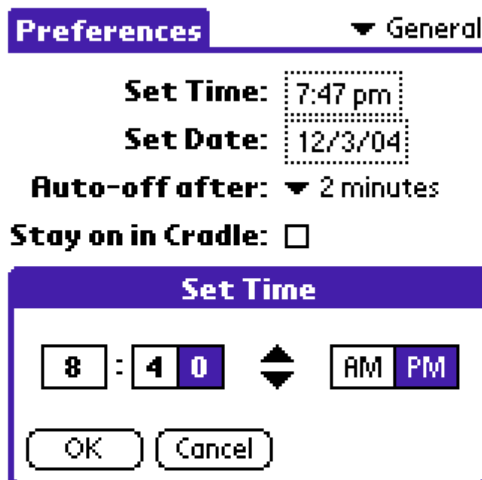


Figure A.10-3: Set Time

Scrolling

MOTIVATION

G3-HE-56: Setting time – In “Preferences,” setting time takes too long to scroll; can’t do
 G3-HE-56: Scroll button vs bar – Tips uses scroll buttons instead of scroll bar; scroll buttons are too sensitive
 G3-HE-30: Up/down physical buttons do not correspond to the ones on the screen
 G3-TA-12: User goes past the desired date of his event when navigating.

PROBLEM

Scrolling has proven to be a significant factor in the success and failure of many products and applications. As more and more information is translated in electronic form, the ability to efficiently and effectively navigate through all of this data is vital. The Palm is no exception, especially as it becomes more and more internet-based.

Scrolling appears in many screens throughout the Palm. However, many of the implemented methods of scrolling are suboptimal and can even hinder the user’s ability to perform the task. There is not much consistency with use of scroll buttons and scroll bars (see Figures A.10-1 – A.10-3). At times, scrolling is too slow. At other times, it is too fast. The user cannot control at what rate he wants to scroll either, even if he knows that he only needs to scroll a very little, or if he does a lot, he must scroll exactly as fast as

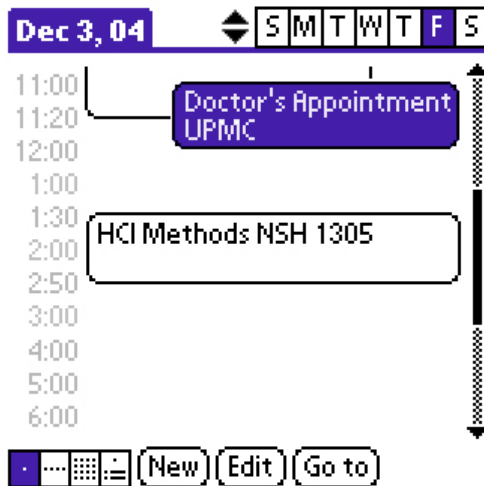


Figure A.10-4: New calendar view

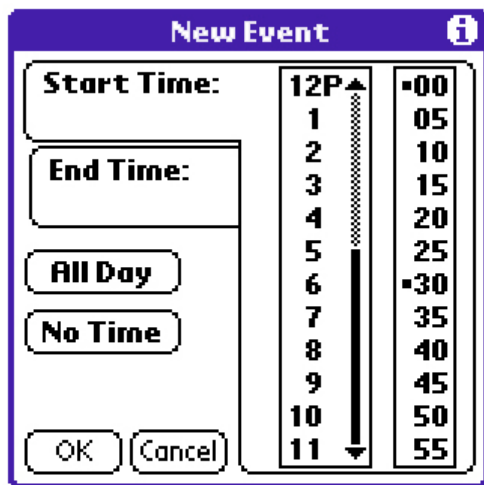


Figure A.10-5: New event setting

the developers allow them to. For many of the scroll bars, the user also cannot even tell where he is in terms of the list he is scrolling, whether in the beginning, middle or end. Also, the physical up/down buttons on the Palm do not correspond to their functions and can be confusing for many users.

REDESIGN SOLUTION

To alleviate these problems and inconsistencies we discussed quite a few possible solutions. We strongly considered standardizing the scroll bars but eventually decided that the scroll bars themselves were not completely impractical and that there was not sufficient evidence to call for a complete redesign of all scroll buttons. However, we did acknowledge that the efficiency and control of scrolling needs addressed.

To resolve this we propose to add a job wheel to the side of the Palm to give them the options to use this for more controlled scrolling. The wheel would allow users to scroll through screens very quickly or very slowly, at the user's complete control. It would also save the user's time because he would not have to tap on the screen or move their hand to the physical buttons.

In the Date Book we also revamped some of the scrolling methods throughout the application, but decided not to expand these changes in other applications. The format that they were based upon was greatly influenced by the very common scrolling bars implemented in Windows and Macintosh systems. This includes a bar that shows its status in terms of the whole page and dragging that bar allows for controlled scrolling, clicking away from the bar for fast scrolling of a single page, and arrows at the end for fine scrolling of single lines (see Figures A.10-4 and A.10-5).

TRADEOFFS

The only tradeoff to standardizing the scroll bars and making them more effective that we could find is that it would require more memory and processing power, which is often vital in systems with very limited resources such as the Palm. However, the consistency, control, and freedom given to the users due to this should be well

worth the cost.

Adding a jog wheel would also be more work for both the product developer and the Palm processor. Also, some users may not associate the jog wheel with scrolling. In instances where there are multiple things to scroll, the jog wheel would only have limited use and might even confuse some users. As a potential positive tradeoff, adding a jog wheel could eliminate the need for the two physical up/down buttons and allow room for other buttons, labels, or options.

Figure A.11.1-1: To Do Item Details

Figure A.11.1-2: To Do main view

Figure A.11.1-3: New priority drop-down menu

Labeling & Naming: To Do List Priority

MOTIVATION

G3-HE-23 – Unclear labeling of priorities in To Do List

PROBLEM

Users may mistake the priority which is on a scale of 1 to 5, as 1 being the worst in thinking of it as low priority, or 1 being the best as being 'number 1 priority'. Conversely, a person could interpret 5 as a high priority in comparison to 1 for example, or interpret it as 5th highest priority. This ambiguity could cause potential problems and frustration to users who wrongly assume that 5 is the highest priority (see Figure A.11.1-1 and Figure A.11.1-2).

REDESIGN SOLUTION

We feel that the best solution for this would be to specify whether it is of high or low priority by using specific words, which we chose as "highest" and "lowest" to indicate the ranges for the drop-down menu (see Figure A.11.1-3) and as "high" and "low" in the details box (see Figure A.11.1-4). With this, a user is immediately and explicitly told what the numbers mean without impeding on his work any more than before.

TRADEOFFS

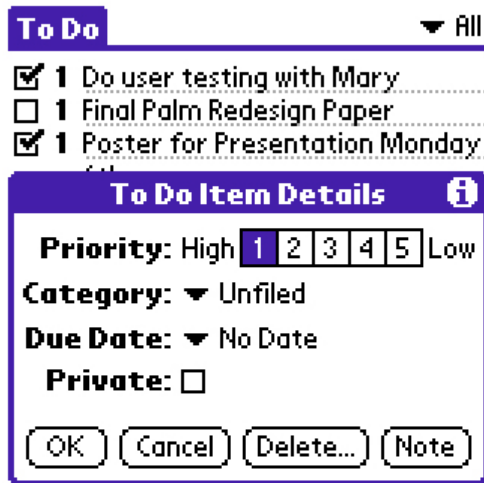


Figure A.11.1-4: New Details dialog

The only tradeoffs that we could think of for this is that more screen space is used in both the drop down menu and in the Details box (see Figure A.11-3 and Figure A.11.1-4). For the drop down menu, this is only temporary and should not affect a user's work at all since it will disappear once he makes a choice. In the Details box, the contents are slightly more cluttered, but the consistent format of the box still holds so it should not have an adverse effect on the user's experience.



Figure A.11.2-1: Original application screen

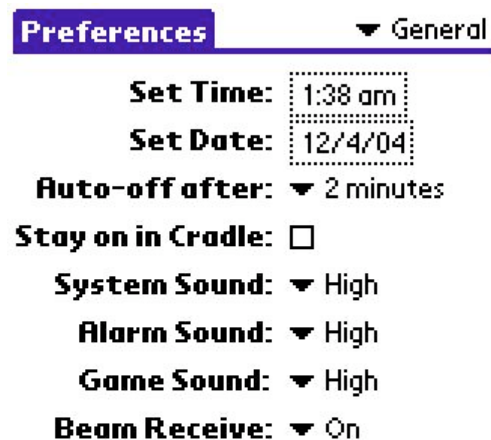


Figure A.11.2-2: Original Preferences screen

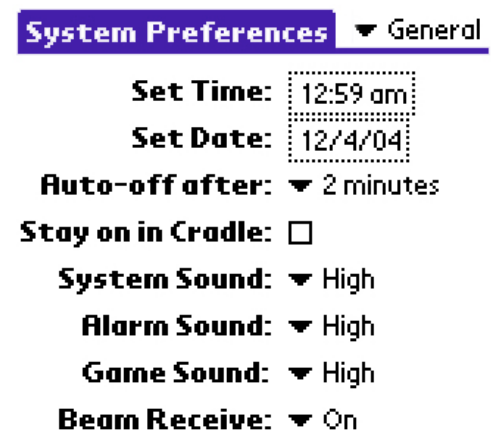


Figure A.11.2-3: New System Preferences screen

Labeling & Naming: Preferences

MOTIVATION

G3-TA-06 – User does not understand the function or context of “Preferences” items

PROBLEM

Users may not understand the use of the Preferences application, specifically what the application is changing preferences for, and therefore may misunderstand the context of the application. One user in our Think Aloud evaluation, for example, interpreted this as for setting preferences for an alarm application, taking it dangerously out of context of the real purpose. The Preferences application is very significant in that any changes to it can adversely affect all other applications. A person's time could be set forward to the wrong time, the date to the wrong date, system sounds turned off, any of which could have profound side effects.

REDESIGN SOLUTION

We feel that there should be a more appropriate and descriptive label for the Preferences application, specifying it as “System Preferences” in title bar (see Figure A.11.2-3) and “SysPrefs” in the Application screen (see Figure A.11.2-4) where less screen space is available. By specifying it as System Preferences, this avoids any possible confusion with other applications.



Figure A.11.2-4: New application screen

TRADEOFFS

Someone may not understand what Prefs is referring to, so one could potentially still have problems with the “SysPrefs” name, but that would be instantly resolved if they click on the application and see “System Preferences”. The name “System Preferences” is also significantly longer than any of the other application names, so the more descriptive labels would also take more screen space than their earlier counterparts. Furthermore, the tab-like title bar needed to be extended more than for other applications, conflicting with the small-tab consistency of the other programs. However, we believe that these are much less significant issues than not specifying the context in which the Preferences application exists.

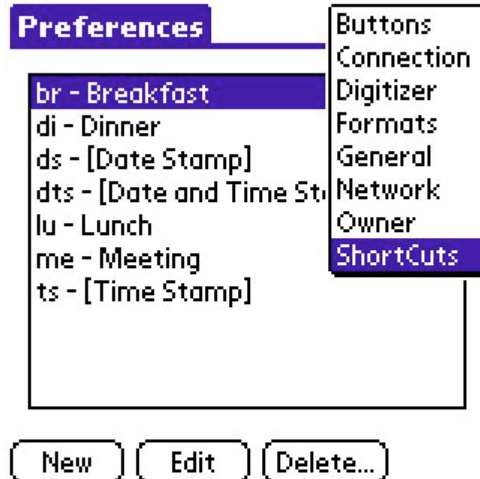


Figure A.11.3-1: Original Preferences listing

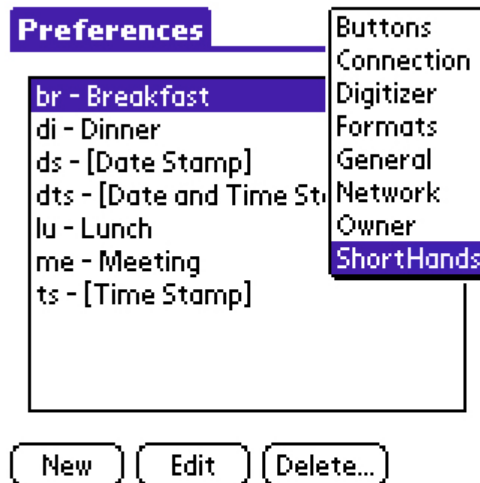


Figure A.11.3-2: Preferences listing with new label

Labeling & Naming: ShortCuts

MOTIVATION

G3-HE-13 ShortCuts unclear - Unclear what ShortCuts are or how to use them

PROBLEM

The Preferences application refers to functionality that they named “ShortCuts” (see Figure A.11.3-1). In modern Operating Systems, such as Windows and Macintosh, a “shortcut” primarily signifies to a link to a file or application. This connection is very strong as shortcuts are used daily by a large number of people. It also is used for providing a series of keystrokes to automate a task, for example in Windows to copy highlighted text, a shortcut would be to press Ctrl+C. The Palm developers probably tried to imply a parallel with this particular definition, but the former definition is much stronger. Furthermore, the latter definition of shortcuts are nearly always to perform some sort of task, whether to copy, paste, close, or otherwise. However, the ShortCut’s functionality is not to do this but instead provide a shorthand for writing a particular amount of text.

REDESIGN SOLUTION

We feel that there should be a more appropriate and descriptive label for the ShortCuts functionality. Since it is really there for

providing a shorthand for larger amounts of text, we felt that it was most fitting to use the term “ShortHands” (see Figure A.11.4-2). This much better describes what the functionality is really used for, and does not conflict with any established computer terms that could take it out of context as “ShortCuts” did.

TRADEOFFS

Someone previously accustomed to the “ShortCuts” label may have trouble associating the change from “ShortCuts” to “ShortHands.” They could assume that the functionality was therefore discontinued, though being able to make the connection from ShortCuts to ShortHands should be fairly obvious. Also, being one more character, it takes a slight amount more screen space though the change does not appear significant as the format and size of the surrounding box remained the same.

APPENDIX B: Evidence

B.1 Affinity Diagram

B.2 UARs

B.2.1 Heuristic Evaluation

B.2.2 Cognitive Walkthrough

B.2.3 Think Aloud

B.3 Contextual Inquiry Interview Transcript

Please see separate package