MORE IN MY FACE, LESS IN MY WAY

Palm Pilot Redesign and Retrospective

Group B:
Michele Clarke
Jason Chalecki
Michael Coblenz
Bryan Crowe

Submitted for:
Chris Neuwirth
John Zimmerman
Intro to HCI Methods
Carnegie Mellon University
Fall 2005

TABLE OF CONTENTS

Executive Summary

Interfaces should be designed such that users think only of their task, and not about the interface itself. This frees users to do their tasks in their own way, rather than in the system's way. In this project, we present a redesign of the Palm interface that has the objective of allowing users to focus on their tasks, rather than on complexities of the interface, with a twist: the Palm should help users remember their tasks. In this respect, notably opposite the previous goal, the system must come to the surface. The challenge we addressed was how to improve the Palm interface so that it is less in the user's way, but more "in her face", i.e. brings important facts to the surface. This change, motivated by the contextual inquiry, resulted in significant design innovations.

A team of four computer scientists, including heavy math backgrounds with touches of design and business experience, tackled this problem. First, we addressed a number of usability problems that were either predicted by HCI methods or directly experienced in user tests. Second, motivated by data, we invented novel interaction techniques, by which the system can actively help the user remember her tasks.

We redesigned the event details window and the To Do details window to address usability problems found in a think-aloud study, heuristic evaluation, and cognitive walkthrough. Accompanying this change was a new alarm settings window and alarms for To Dos. This change was motivated by cognitive walkthrough and heuristic evaluation, and supported by data in a contextual inquiry.

We invented two interaction techniques for helping users remember their tasks:

· We extended the existing Palm To Do interface by adding a To Do visualization. This visualization contains every To Do, and is always visible on the screen. Users can sort the list in several ways and also adjust the order manually. Each item is shown as an abbreviation, which can be expanded to the full title or the complete Details screen.

· We added a plugin to the user's email client. The plugin adds a button to the header region of each email. When clicked, the button allows the user to create a new To Do, which is immediately sent to the Palm. This new To Do is linked to the original email, so in the Palm interface, users can easily see the original source for each To Do that came from an email. This improves visibility of To Dos, helping the user keep important items 'in their face,' as a user expressed was necessary in a contextual inquiry.

By fixing several serious usability problems with the Palm, we make the system less in the user's way, empowering them to complete tasks more easily. Adding a new To Do visualization and linking emails to To Do items helps ensure that users do not forget their tasks. These changes represent a significant improvement over the existing Palm interface, in which tasks can easily be forgotten.

Redesign Discussion

Introduction

Our path to a Palm redesign revealed numerous design problems with the Palm. We generated ideas for improving the interface from two perspectives:

· Contextual Inquiry inspired us to create significant innovations that help users remember their tasks.

· UARs described numerous design problems. Rather than addressing various unrelated UARs, we chose a few key areas of the interface to thoroughly redesign. This approach resulted in a coherent redesign and fixed several serious problems, at the cost of not addressing all of the most serious problems. Nonetheless, we addressed 67% of the 12 problems of severity 3 or 4.

Following this philosophy, we redesigned the alarm dialog, To Do Details window, and Event Details window, since we felt we could improve those core areas of the Palm significantly (based on the data we collected). We also added an Alarm button to the To Do and Calendar views to address user difficulties in finding the alarm.

We discarded several design changes because they did not relate to our focus or were impractical. For example, we invented a calendar visualization for easily viewing several weeks of events at a time. We also designed a visualization for alarm settings for To Do items, but our best design was not practical for the Palm's small screen. (Refer to Appendix D.)

Contextual Inquiry gave us insight into the needs of real users: ensuring that tasks are not forgotten. To that end, we added a To Do visualization to help users remember their tasks, and an email client plugin to help users track email inspired tasks.

After creating our new design, we evaluated it with a heuristic evaluation. We found several problems, documented as TD-HE-1 through TD-HE-4, and iterated on our design to fix them. We also iterated several times on our other design changes to improve their usability.

Success Story

Leonard is a recently tenured professor at a state university. He is passionate about his work and the football team. He is very sociable, with great sense of humor, making even department meetings fun. And Leonard cannot say no to anybody. He claims that it is his one healthy addiction, but sometimes it can be too much.

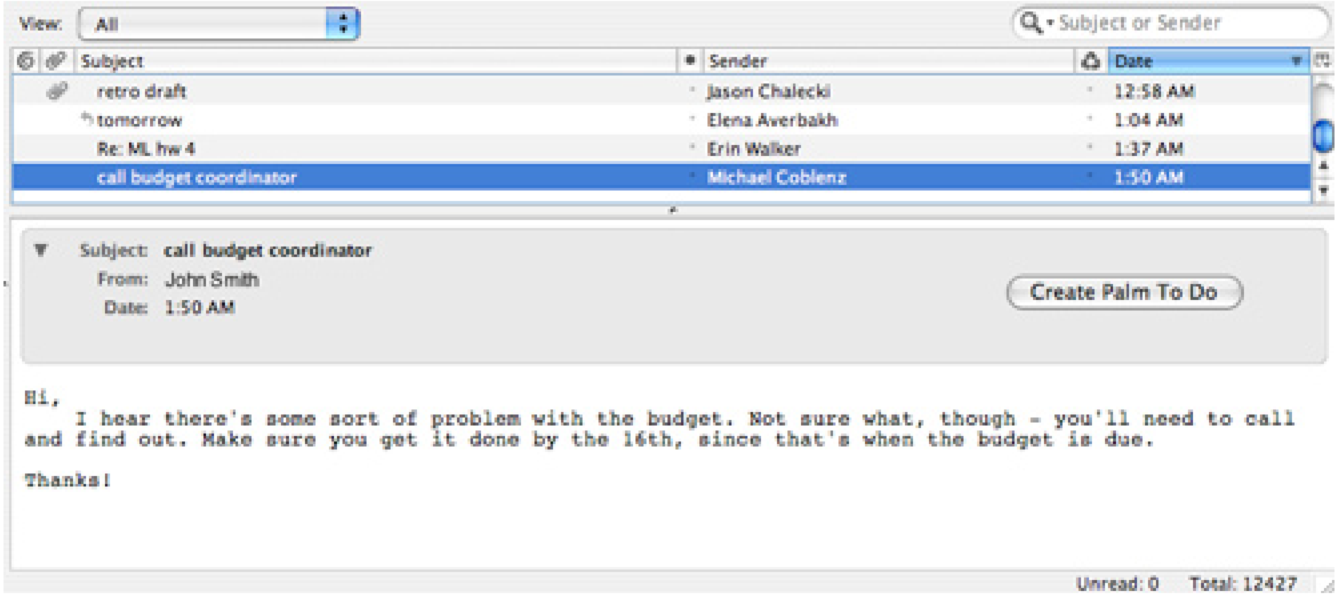
What he really loves is kicking back in the mornings with a cup of coffee and the newspaper. This morning is no exception. He takes his time, enjoying the relaxation. When he gets to the sports section, the dog starts jumping at his heels to be let out and his Palm buzzes with an alert that his first meeting is in two hours. Thankful that his Palm and dog keep better track of time than he does, he quickly skims the BCS polls before heading upstairs to get ready for what will certainly be another busy day.

Step 1

Upon arriving to work, Leonard says hi to some of his fellow co-workers and heads into his office to check his email before the meeting. He notices that some emails involve tasks for him. He selects an important email from his budget coordinator and taps the ‘Create Palm To Do’ button.

Redesign Aspects:

To Do Email Plugin



Subject	Sender	Date
retro draft	Jason Chalecki	12:58 AM
tomorrow	Elena Averbakh	1:04 AM
Re: ML hw 4	Erin Walker	1:37 AM
call budget coordinator	Michael Coblenz	1:50 AM

Subject: call budget coordinator
From: John Smith
Date: 1:50 AM

```
Hi,
    I hear there's some sort of problem with the budget. Not sure what, though - you'll need to call
and find out. Make sure you get it done by the 16th, since that's when the budget is due.

Thanks!
```

Unread: 0   Total: 12427

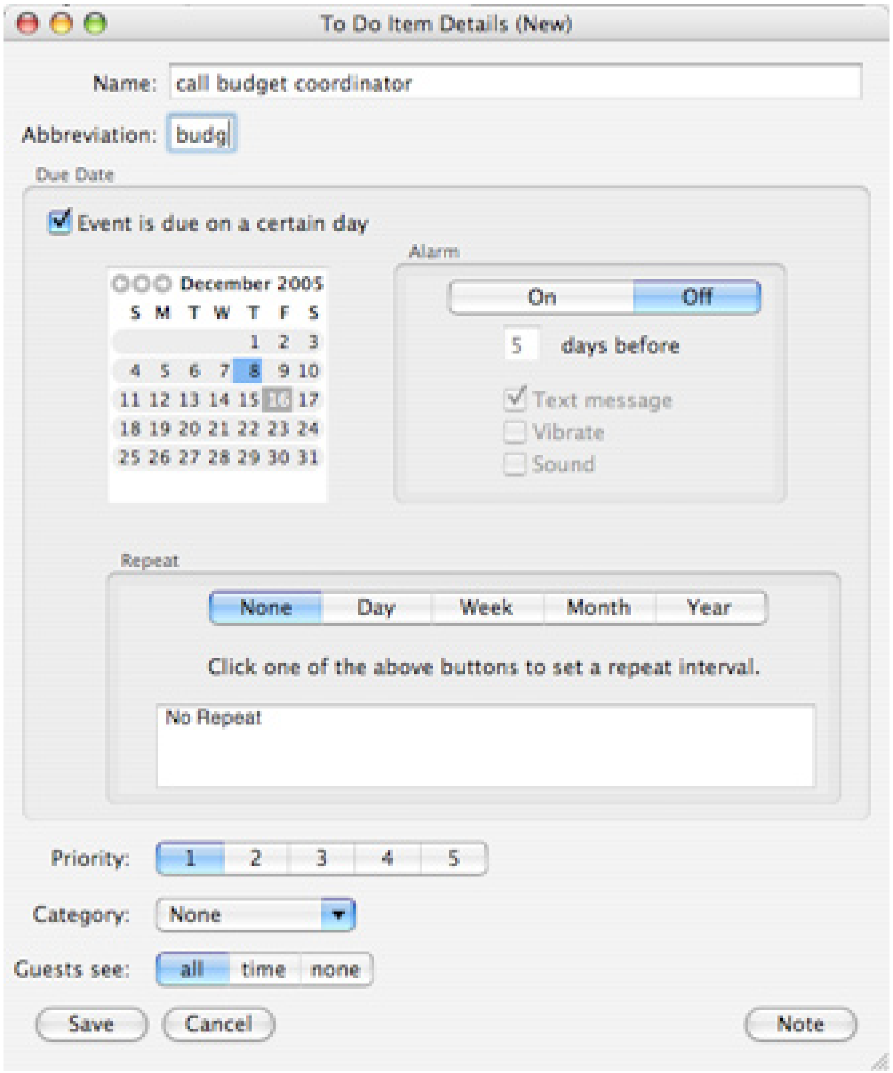
Figure 1

Step 2

Leonard's budget coordinator needs him to call her on the 16th with the latest update of the proposal. Leonard types in a name for the to-do and gives it his usual abbreviation, 'budg'. He quickly sets the date to the 16th and taps 'Save'. With twenty minutes left before his meeting, Leonard creates to-do's for all the necessary emails.

Redesign Aspects:

To Do Email Plugin



To Do Item Details (New)

Name: call budget coordinator

Abbreviation: budg

Due Date

☑ Event is due on a certain day

December 2005

S	M	T	W	T	F	S
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

Alarm

On [✓ Off]

5 days before

☑ Text message

□ Vibrate

□ Sound

Repeat

[✓ None] Day Week Month Year

Click one of the above buttons to set a repeat interval.

No Repeat

Priority: [✓ 1] 2 3 4 5

Category: None

Guests see: [✓ all] time none

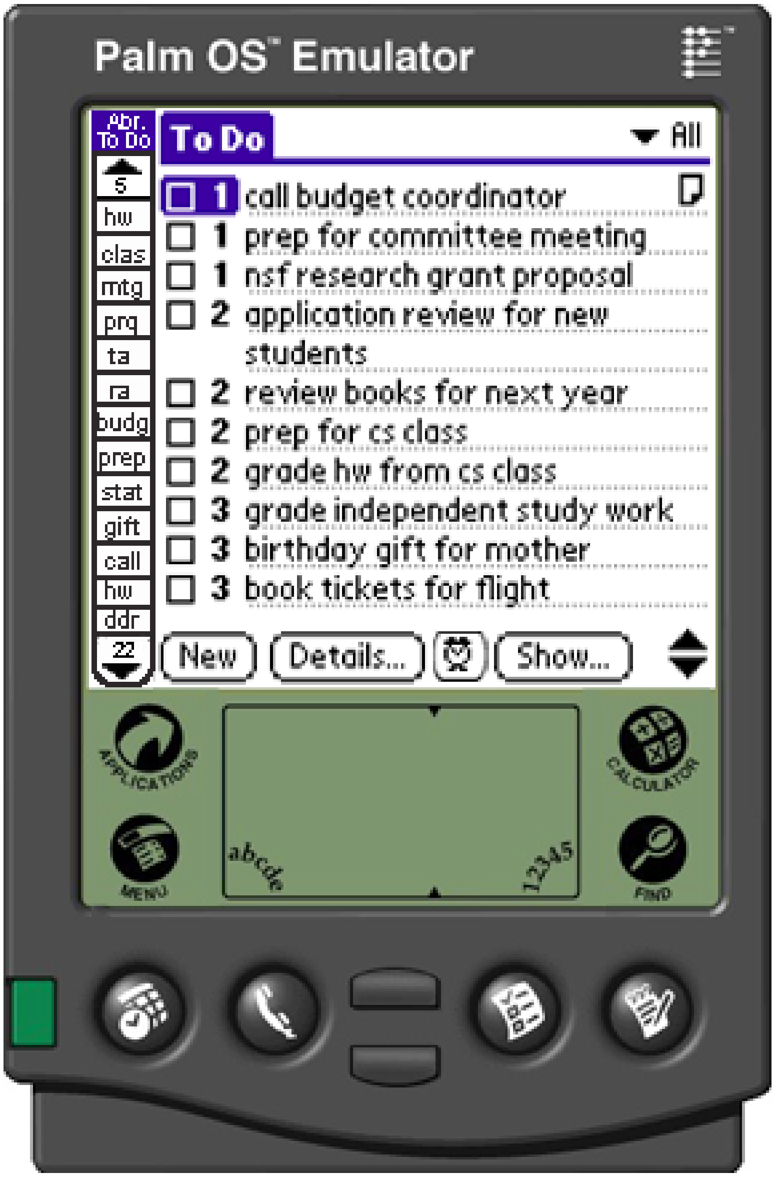
Figure 2

Step 3

Leonard checks how he is doing on time and sees that the meeting starts in five minutes. He quickly grabs his Palm and heads out the door. On the way to the meeting, Leonard decides that he needs two days' warning before he calls his coordinator to ensure that the budget is finalized. He turns the Palm on, using the ToDo button, to set an alarm.

Redesign Aspects:

Alarm Button



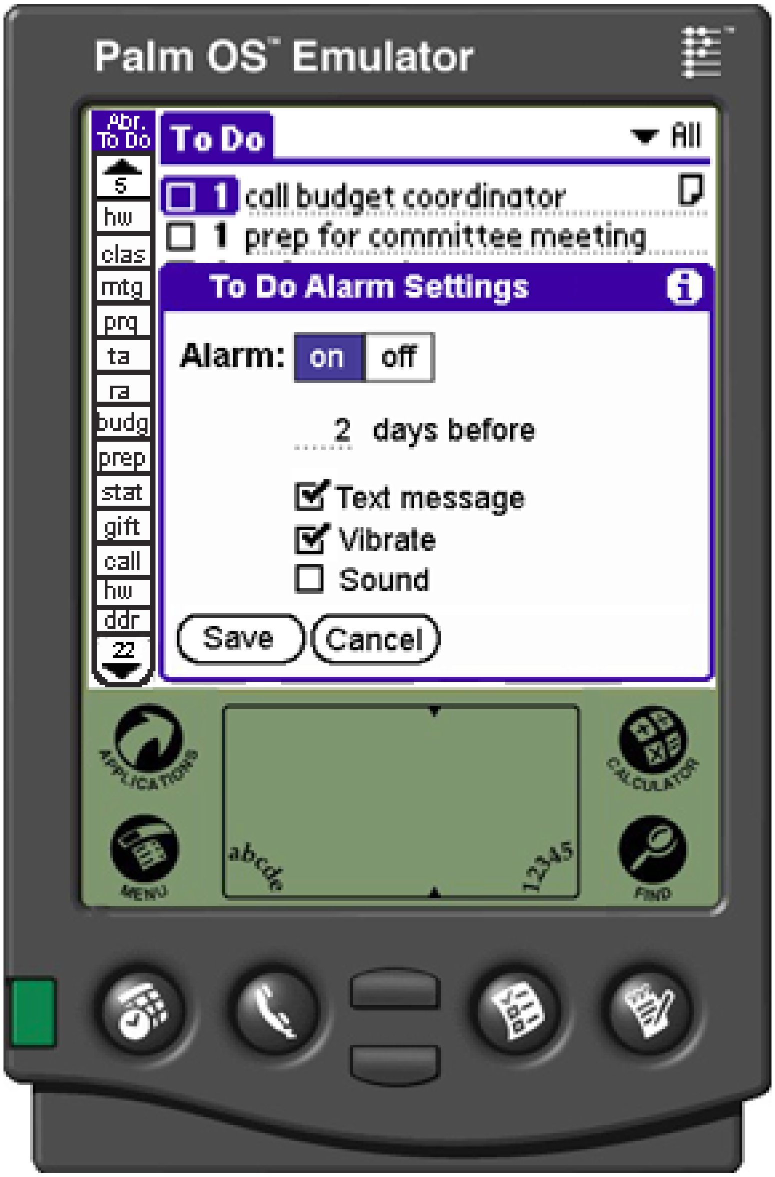
□ 1 call budget coordinator
□ 1 prep for committee meeting
□ 1 nsf research grant proposal
□ 2 application review for new students
□ 2 review books for next year
□ 2 prep for cs class
□ 2 grade hw from cs class
□ 3 grade independent study work
□ 3 birthday gift for mother
□ 3 book tickets for flight

Figure 3

Step 4

Leonard uses Grafitti to indicate 2 days and sets it to notify him with both a text message and a vibration. He taps 'Save' and sees the alarm icon, confirming that his alarm has been saved.

Redesign Aspects:
Alarm Window



□ 1 call budget coordinator
□ 1 prep for committee meeting

To Do Alarm Settings

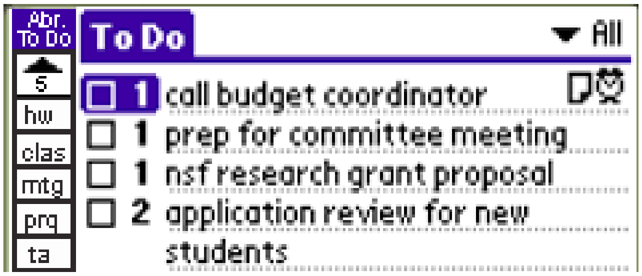
Alarm: on off

2 days before

☑ Text message
☑ Vibrate
□ Sound

Save Cancel

Figure 4



□ 1 call budget coordinator
□ 1 prep for committee meeting
□ 1 nsf research grant proposal
□ 2 application review for new students

Figure 5

Step 5

During the meeting, Leonard is about to use the calculator function. First, however, he glances at the abbreviation todo list and sees an unfamiliar one, labeled 'ta'. He taps it and sees the full title, telling him to hire a new teaching assistant. This description quickly jogs his memory and he is glad this todo was 'in his face'. Curious about the details, he taps '+'.

Redesign Aspects:

Abbreviation To Do List

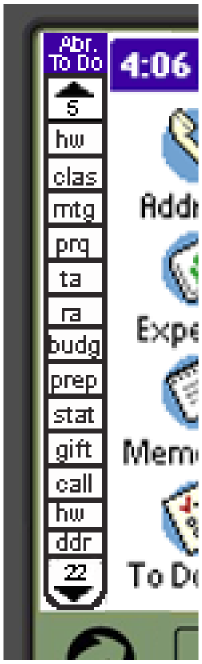


Figure 6

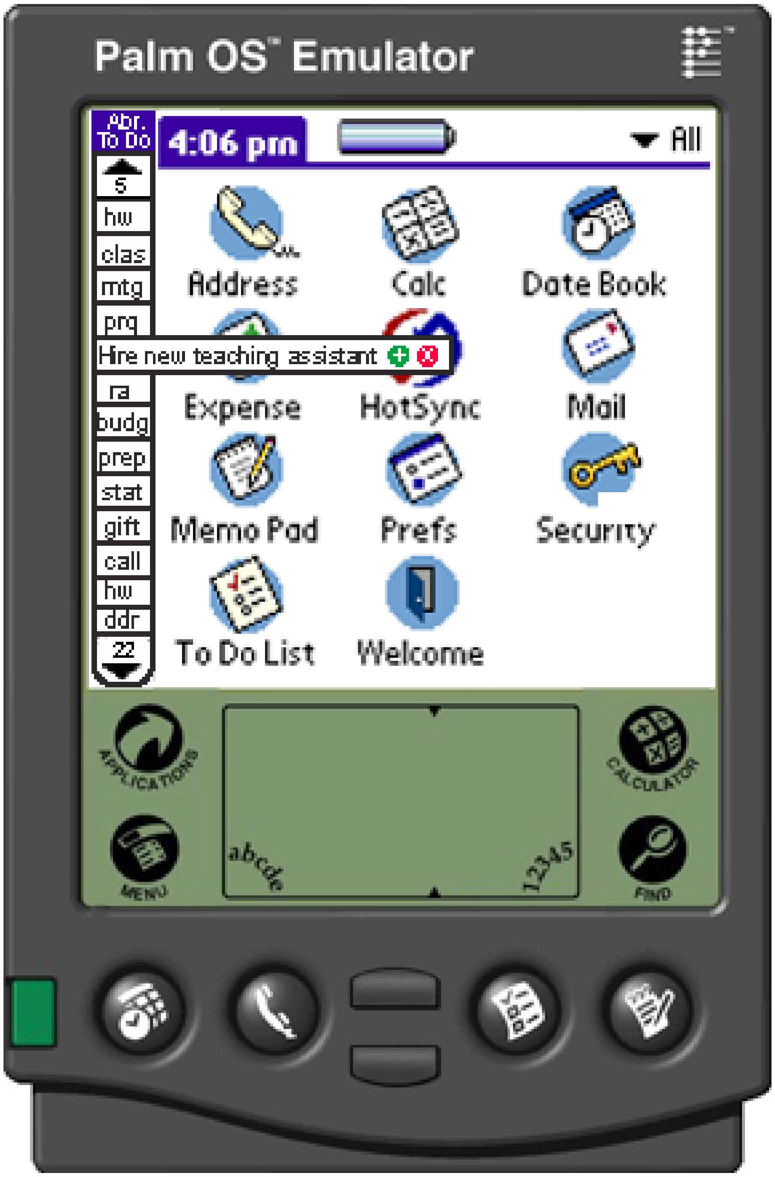


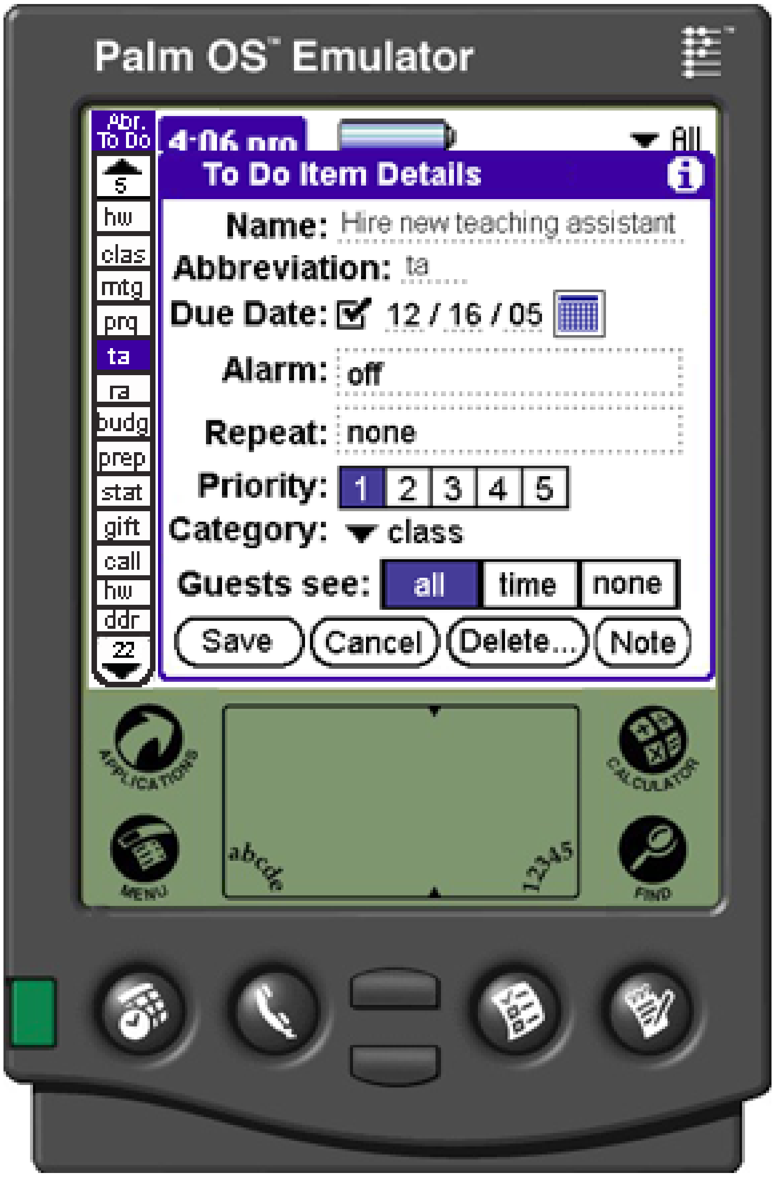
Figure 7

Step 6

Leonard sees that he needs to hire a TA by the 16th, but he forgets who he needs to coordinate with to do this. He taps 'Note' to see more information and access the original email.

Redesign Aspects

Abbreviation To Do List
To Do Item Details



To Do Item Details

Name: Hire new teaching assistant
Abbreviation: ta
Due Date: ☑ 12 / 16 / 05
Alarm: off
Repeat: none
Priority: [✓ 1] 2 3 4 5
Category: class
Guests see: [✓ all] time none

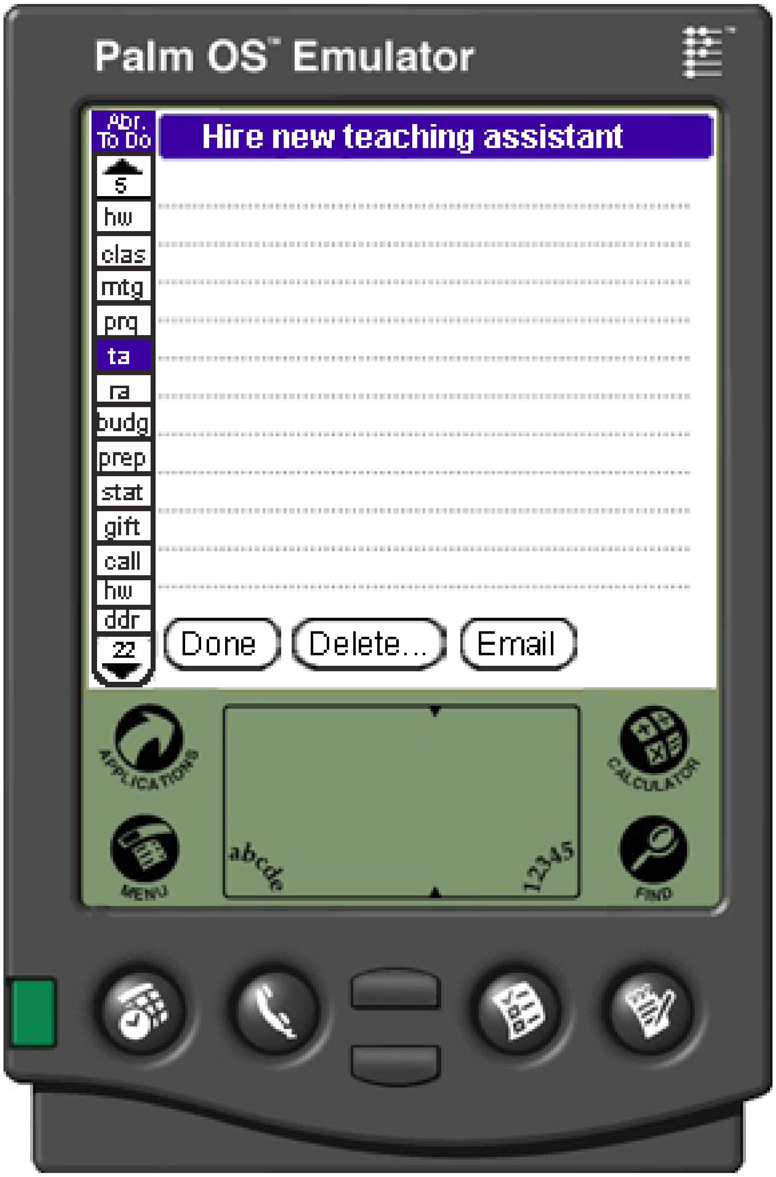
Figure 8

Step 7

The Palm automatically associates to-dos with the originating email, so Leonard taps ‘Email’.

Redesign Aspects:

Abbreviation To Do List
To Do Email Plugin



Hire new teaching assistant

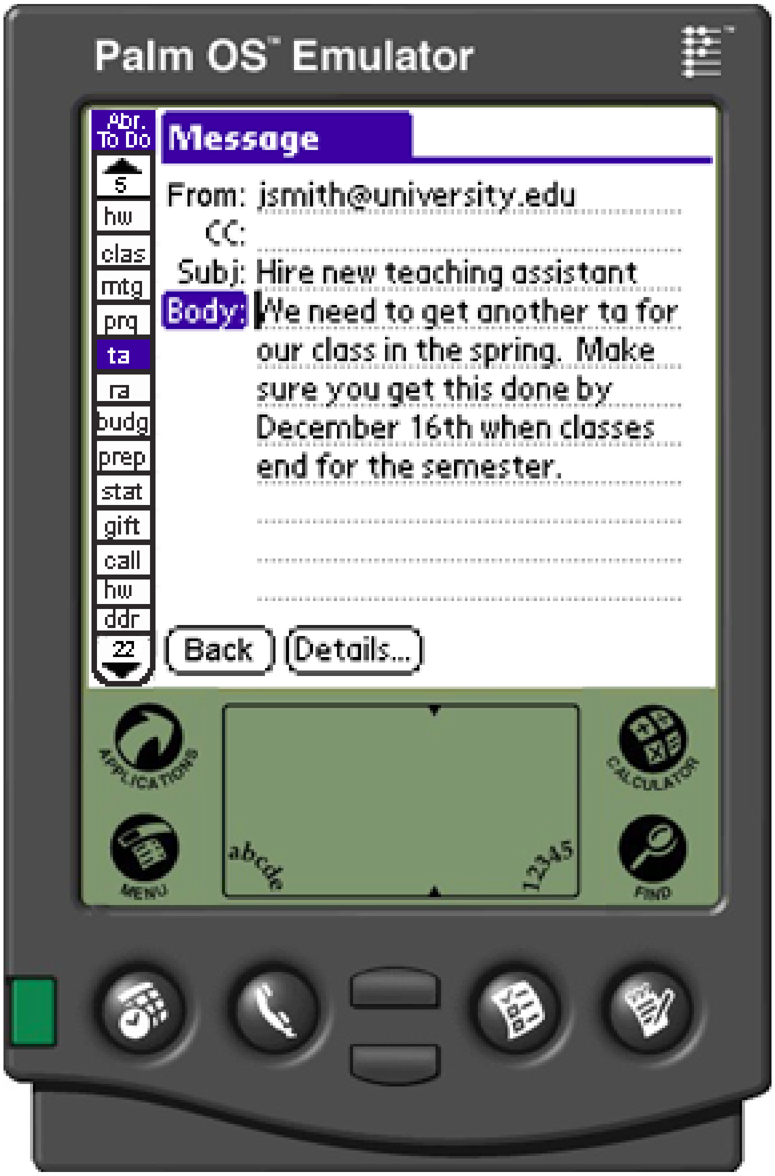
Figure 9

Step 8

Leonard reads the email and sees that it is from John. Fortunately John is in the same meeting. During the break they decide to meet the Monday before the 16th to consider their options.

Redesign Aspects:

Abbreviation To Do List
To Do Email Plugin



From: jsmith@university.edu
CC:
Subj: Hire new teaching assistant
Body: We need to get another ta for our class in the spring. Make sure you get this done by December 16th when classes end for the semester.

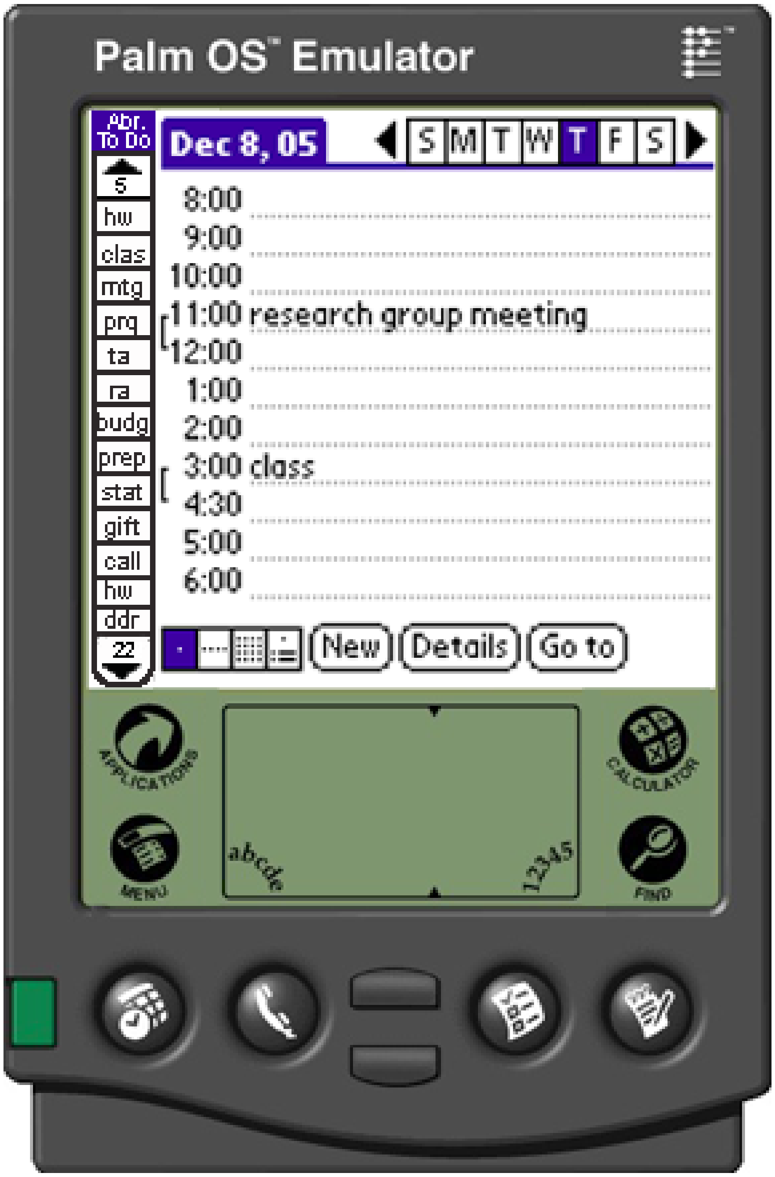
Figure 10

Step 9

He taps the calendar button followed by 'New' to schedule the meeting. He needs to make a new event for this meeting, so he taps 'New.'

Redesign Aspects:

Event Details Window



Dec 8, 05

8:00
9:00
10:00
11:00 research group meeting
12:00
1:00
2:00
3:00 class
4:30
5:00
6:00

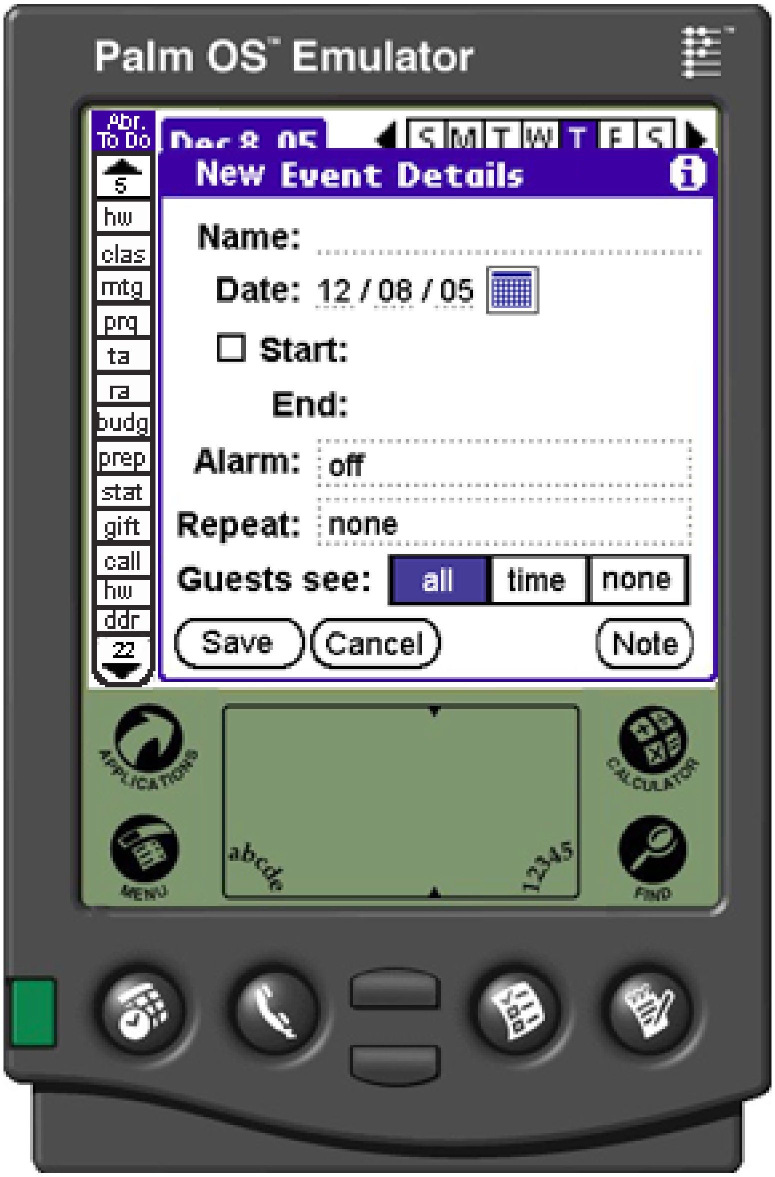
Figure 11

Step 10

Leonard uses Grafitti to enter the name. He is not sure what the exact date is, so he taps the calendar icon to choose the date.

Redesign Aspects:

Event Details Window



New Event Details

Name:

Date: 12 / 08 / 05

☐ Start:

End:

Alarm: off

Repeat: none

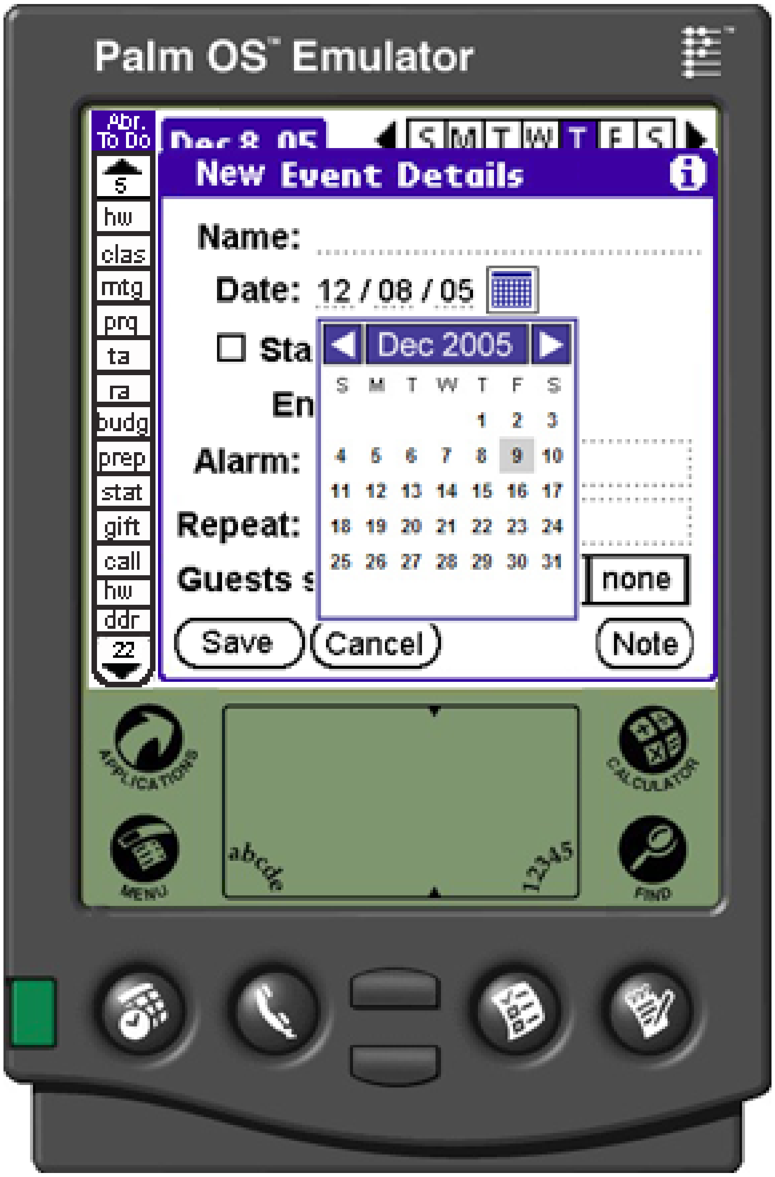
Guests see: all time none

Figure 12

Step 11

Shocked as always that end of semester has come so quickly, he looks at the calendar and sees that the 12th is the day he wants. He taps it and the calendar popup disappears, as the date is set to 12/12/05.

Redesign Aspects:
Event Details Window



New Event Details

Name:

Date: 12 / 08 / 05

□ Sta

En

Alarm:

Repeat:

Guests s

none

Dec 2005

S	M	T	W	T	F	S
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

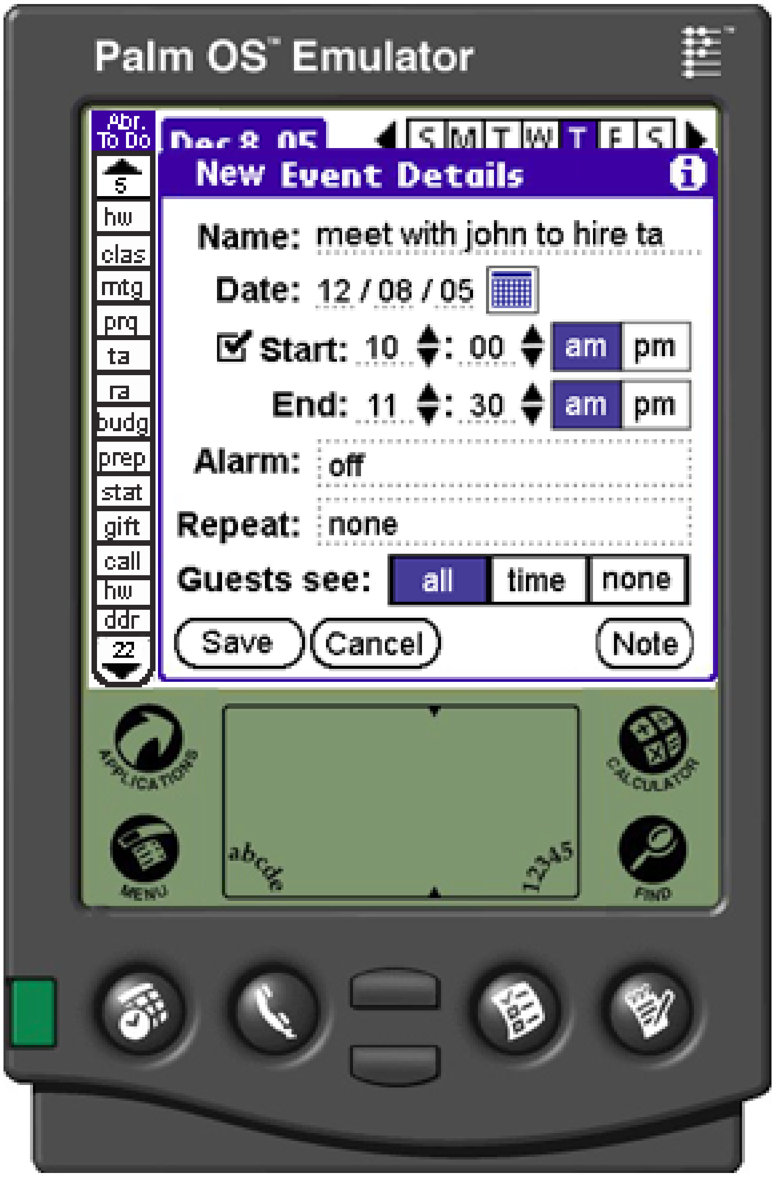
Figure 13

Step 12

Leonard agreed with John to meet from 10-11:30 am, so he enters these values using the arrows.   With everything entered he taps 'Save'.

Redesign Aspects:

Event Details Window



New Event Details

Name: meet with john to hire ta

Date: 12 / 08 / 05

☑ Start: 10 : 00 [✓ am] pm

End: 11 : 30 [✓ am] pm

Alarm: off

Repeat: none

Guests see: [✓ all] time none

Figure 14

Redesign In Depth

Abbreviated To Do List

Design

On the left side of the screen is a list with one item for each To Do (Figure 15). Each item shows an abbreviated form of a To Do. For example, a To Do that says "ONR Proposal" might be abbreviated "ONR." If more To Dos exist than can be displayed on the screen, the list becomes scrollable, and the scroll arrows indicate how many To Do items are in each direction.

Users can sort the list in ascending or descending order by priority, due date, category, recent access time, or name by clicking on the Abr. To Do header at the top of the list and selecting the desired option (Figure 18). Users can also manually move items around in the list by dragging them.

There are a couple trade-offs. The label "Abr. To Do" is probably not sufficient to communicate its purpose when first seen. Also, Not every user may want to use the list. They will likely be frustrated with the inability to close the list so they could reclaim the screen space. However, we felt the ability to close allows the list to not be in the user's face.

There are two views:

- Abbreviation view, the default, shows only the abbreviation for each item.
- Title view displays the entire title of an item, using multiple lines if needed. (Figure 16)

In abbreviation view:

- To open title view, single-tap the abbreviation.
- To open details view, double-tap the abbreviation.

In title view:

- To return to abbreviation view, single-tap minus.
- To open item Details (Figure 17), single-tap plus.

Rationale

In my face: During the CI, the user frequently complained that her reminders were not "in her face". One cause was that the blue emails sometimes scrolled off-screen because they were interleaved with non-blue emails. Also, it was essential to this user that she not forget anything. Therefore, we created a To Do view that concisely displays all (and only) To Do items.

By including every To Do, we ensure that items are organized and not forgotten. Combined with the email client plugin, it supports the long lists of To Do items that she frequently maintains in email. Furthermore, it enables the user to quickly glance at To Do items.

Sorting: The ordering of the items in the list is extremely important. The user in the CI sometimes sorts by priority, but also by subject. We allow users to sort by various characteristics but still manually customize the resulting order.

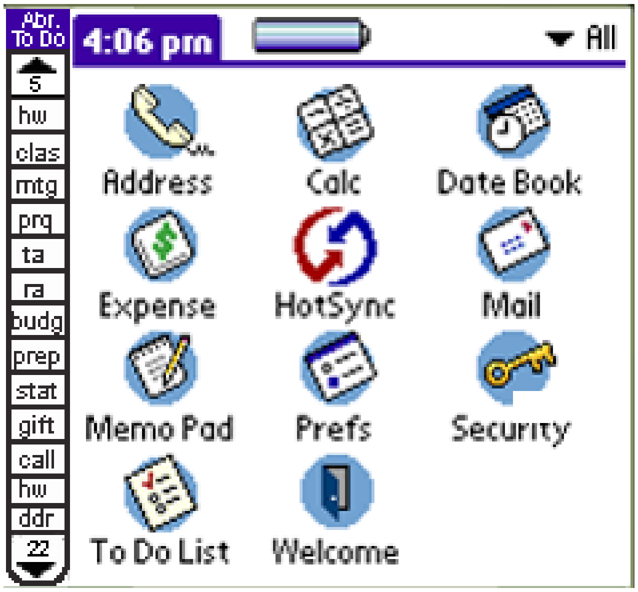


Figure 15 - Redesign

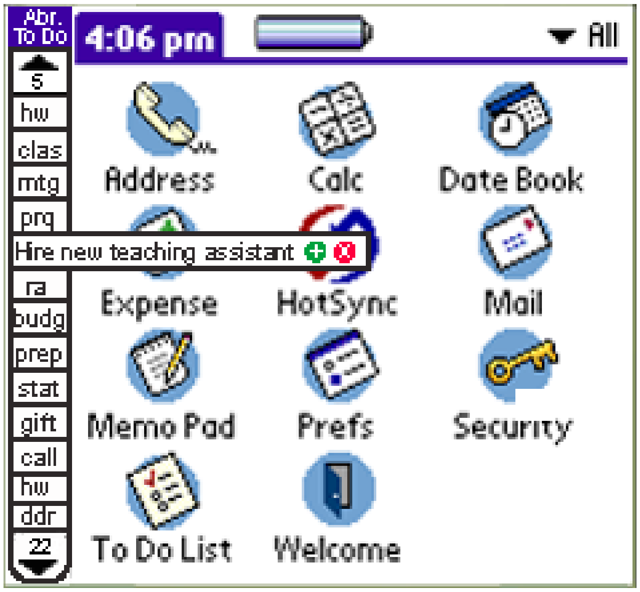


Figure 16 - Redesign

Abbreviated To Do List

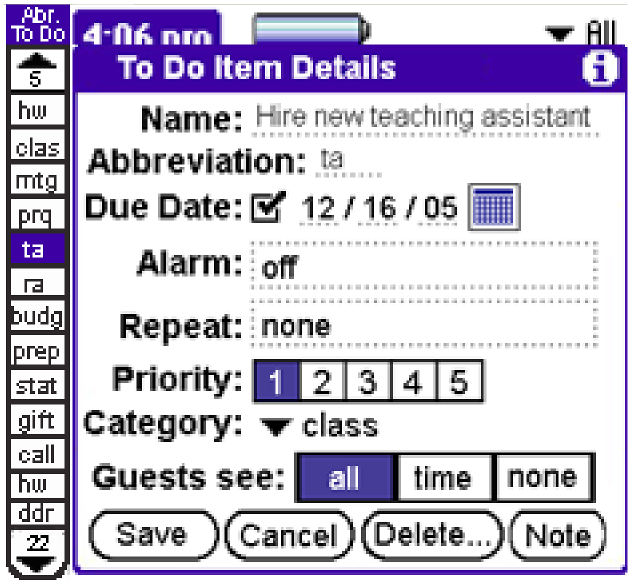
The "recently accessed" sort option helps ensure that users can easily view to-do's that they have not seen lately.

Abbreviations: Users enter abbreviations in the Details window. We reason that this results in the most recognizable abbreviations for users (the recognition rather than recall heuristic) without overly burdening users for data entry because abbreviations are limited to four characters. Furthermore, our CI showed that users are already accustomed to representing to-do items with abbreviations. If a user does not specify an abbreviation, the system generates one.

Scrolling: Near the scroll arrows is the number of To Do items remaining in each direction. This improves visibility because the users know there are more To Do items than what they can see.

Interaction: According to our CI, users frequently need to glance at To Do items, so the title is displayed by merely single-tapping its abbreviation. Double-tapping a To Do in the list reveals the Details window so that users can quickly access all related information. We added plus and minus buttons in title view so that the available options are visible. This argument could have resulted in icons in abbreviation view, as well, but we felt the trade-off of taking up extra screen space was not worth the benefit. Also, our list is on the left side of the screen so that it cannot visually interfere with scroll bars.

Design Change Aspect - Motivating CI Evidence	
In my face	L2-142; L2-258 - L2-262, L2-142, L066 – L068, L3-300, L2-142, L2-168
Sorting	L2-161, L2-165, L2-189, L2-184, L2-193, L2-165, L2-184
Abbreviations	L2-180, L2-204, L2-244
Interaction	L2-168, L2-213



Name: Hire new teaching assistant
Abbreviation: ta
Due Date: ☑ 12 / 16 / 05
Alarm: off
Repeat: none
Priority: [✓ 1] 2 3 4 5
Category: ▼ class
Guests see: [✓ all] time none

Figure 17 - Redesign

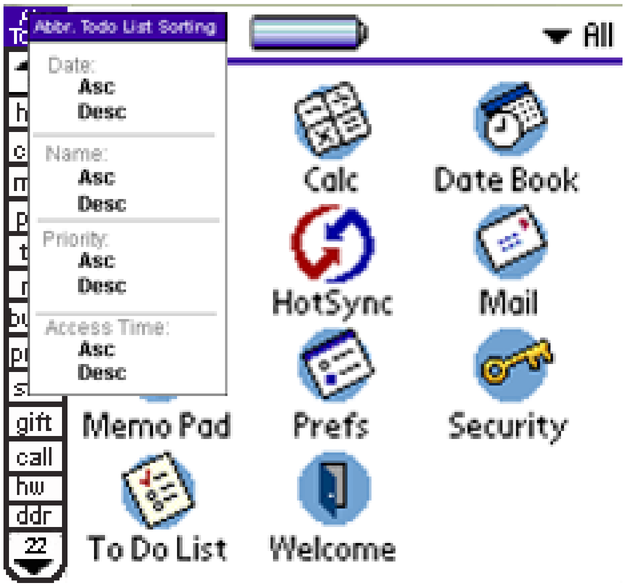


Figure 18 - Redesign

Design

On the right side of the header section of each email is a new button, "Create Palm To Do" (Figure 19). The button displays a window in which users specify details for a To Do corresponding to this email (Figure 20). This window allows users to specify the same settings as are available on the Palm. When the To Do is saved, it is automatically copied to the Palm. A user preference allows users to choose a color, if any, to label the email as a To Do item.

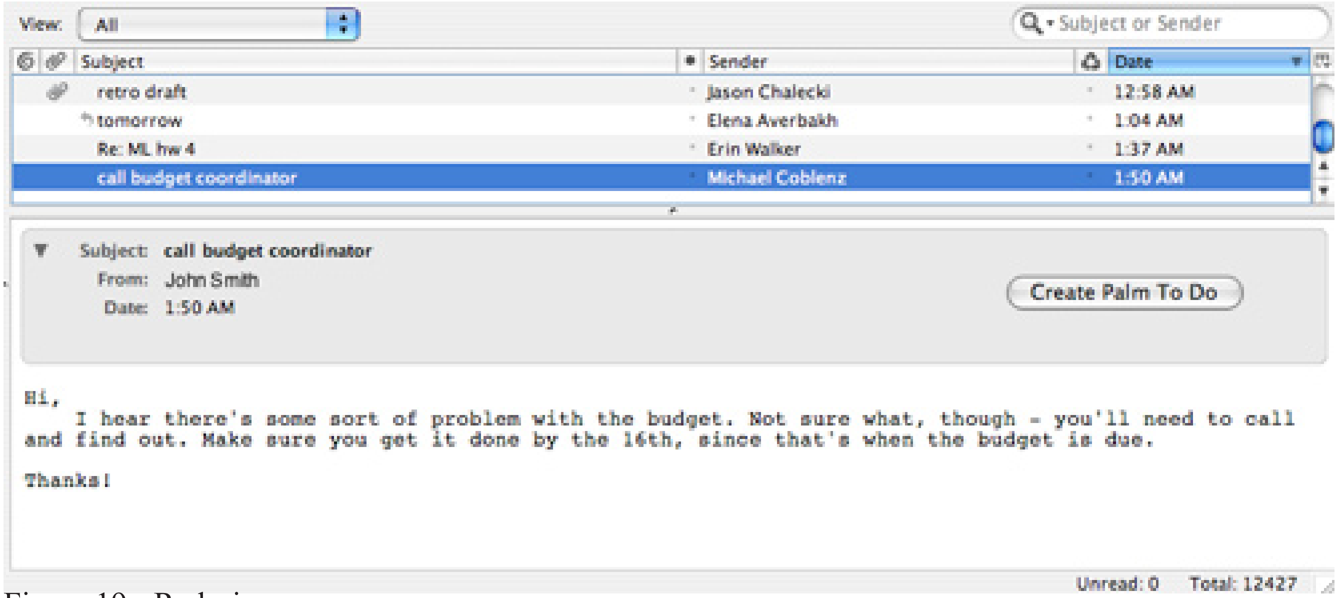
When the To Do is copied, a reference is created from the To Do to its email. This helps users see the original context, and view additional information not already available from the To Do. This reference is shown by a button in the To Do's Note.

Rationale

Button in email client: In the CI, the user frequently receives emails that represent To Do items and marks them as such by setting their labels blue. However, the Palm has a To Do application; we hypothesize that any user would want to take advantage of this to track To Do items. Therefore, we added this button to the client so that users can make To Dos from emails. With the button always visible, it is convenient for users to create To Do items for emails. Also, it is a reminder for users that they should create To Do items when necessary. In our CI, we found that users may forget to mark emails.

Details Window: The window used to enter To Do information (Figure 20) closely resembles the corresponding window on the Palm. The Consistency and Standards heuristic suggests, therefore, that this will be a successful interface. However, we made a few adaptations that make the window more appropriate in a computer context. For example, rather than needing to put the Alarm and Repeat options in separate windows to reduce size, we included them in this window.
Referencing: Associating To Do items on the Palm with the original emails helps users find relevant information.

Design Change Aspect - Motivating CI Evidence	
Button in email client	interview1.mov 16:00, L2-142, L2-162, L2-163, L2-256, L2-250, L2-254
Referencing	L3-296, L3-309



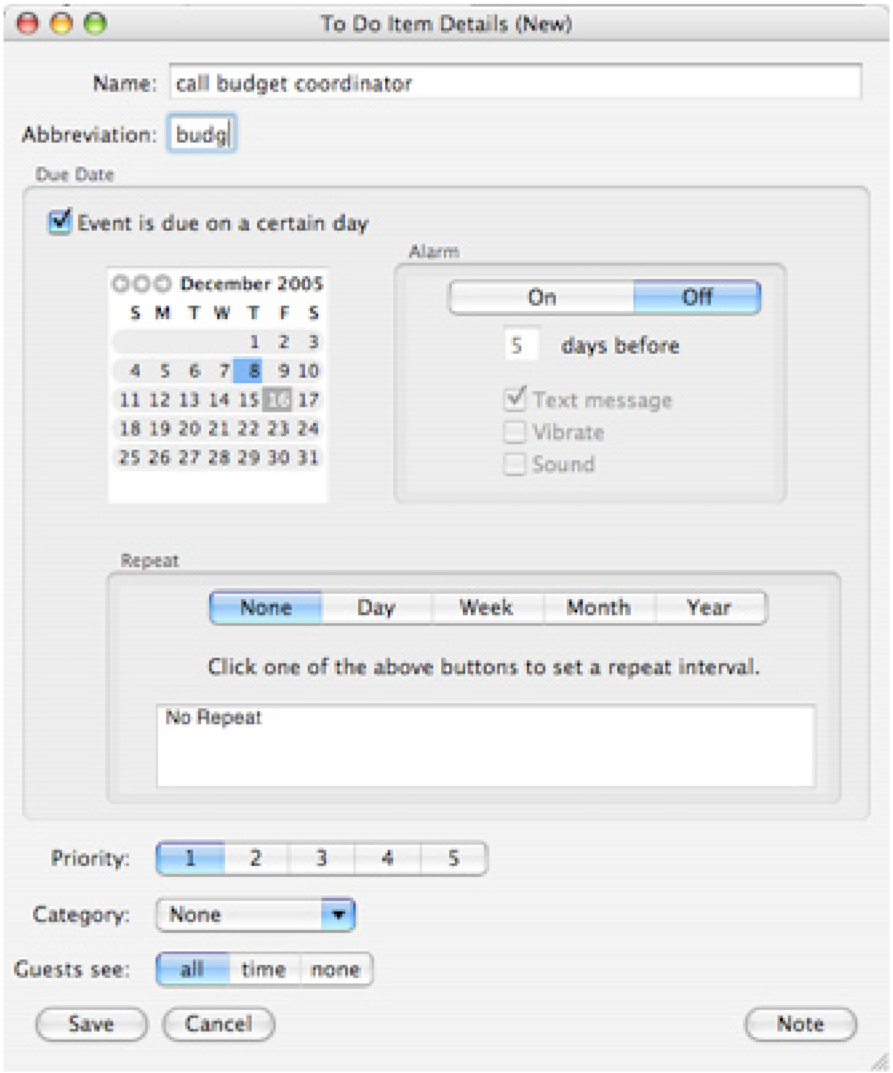
Subject	Sender	Date
retro draft	Jason Chalecki	12:58 AM
tomorrow	Elena Averbakh	1:04 AM
Re: ML hw 4	Erin Walker	1:37 AM
call budget coordinator	Michael Coblenz	1:50 AM

Subject: call budget coordinator
From: John Smith
Date: 1:50 AM

```
Hi,
  I hear there's some sort of problem with the budget. Not sure what, though - you'll need to call
and find out. Make sure you get it done by the 16th, since that's when the budget is due.

Thanks!
```

Figure 19 - Redesign



To Do Item Details (New)

Name: call budget coordinator

Abbreviation: budg

Due Date

☑ Event is due on a certain day

December 2005

S	M	T	W	T	F	S
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

Alarm

On [✓ Off]

5 days before

☑ Text message

□ Vibrate

□ Sound

Repeat

[✓ None] Day Week Month Year

Click one of the above buttons to set a repeat interval.

No Repeat

Priority: [✓ 1] 2 3 4 5

Category: None

Guests see: [✓ all] time none

Figure 20 - Redesign

Event Details Window

Design

We made extensive changes to the Event Details window (Figure 21). This window is now used for creation as well as editing. The window shown handles creation; the editing window is similar, but without "(New)" in the title, and has a Delete button at the bottom.

Our design adds a date picker, allowing users to enter dates with Graffiti or a calendar. We consolidated the name field and all date and time choices in one window. We modified the alarm setting interface and improved the interface for choosing privacy settings.

Rationale

This design change was created to address the fact that the set time dialog box, which was used to create new events in the old interface, was found to be confusing because the content of the dialog box was not indicative of creating a new event (Figure 23).

Name: With the name in Details, the replacement for the "Set Time" window, we ensure that the user enters a name (by displaying an error if the user tries to save without a name). Additionally, this is a better match with the real world, where users are likely to name events before considering details.

Date Setting: Using Details for event creation adds context for the user by providing the date. Users now easily see when they are creating an event for. This may alleviate problems where users were unable to understand the context after tapping "New". We added a calendar popup to aid users in choosing a relative date, e.g. the third Monday of November.

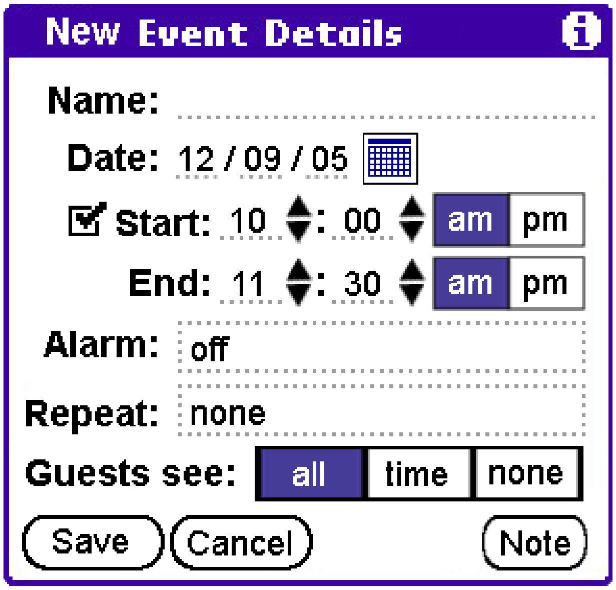
Time Setting: Consolidating the time into this window will make it easier for users to find the alarm, since the alarm is now on the same screen as the event creation interface.

Alarm: This is required to implement the alarm in a dialog. (Design Change: Alarm Window)

Privacy: The new privacy options clarify what the effects of the various choices are by renaming them and collecting them in this dialog. One trade off, however, is that users may not understand that "all" refers to all details not all events.

Save instead of OK: Novices can be unsure whether clicking OK saves their data. Changing the button to say "save" resolves this.

"New" in title: One window is used now for both for creating and editing events. "New" in the title distinguishes new events from old, improving consistency.



New Event Details

Name:

Date: 12 / 09 / 05

☑ Start: 10 : 00 am

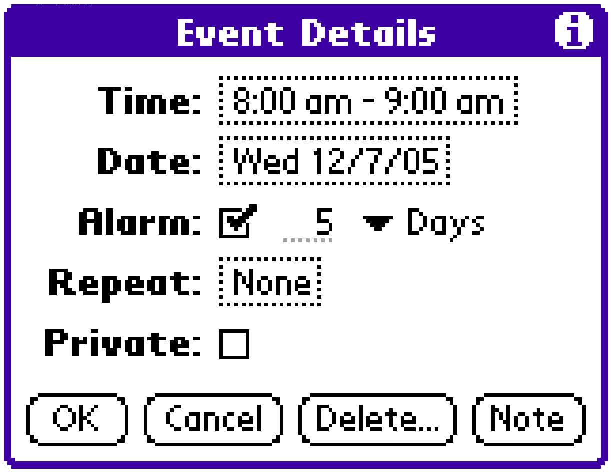
End: 11 : 30 am

Alarm: off

Repeat: none

Guests see: all

Figure 21 - Redesign



Event Details

Time: 8:00 am - 9:00 am

Date: Wed 12/7/05

Alarm: ☑ 5 ▾ Days

Repeat: None

Private: □

Figure 22 - Old interface

Event Details Window

Design Change Aspect - Associated UARs	
Name	B-HE-05
Date Setting	MEC-TA-09, BJC-HE-01
Time Setting	B-CW-03, MEC-TA-09, JPC-HE-08
Privacy	MJC-HE-07, MJC-HE-04
Save instead of OK	JPC-TA-05
"New" in title	B-CW-03

Design Change Aspect - Motivating CI Evidence	
Date Setting	L074

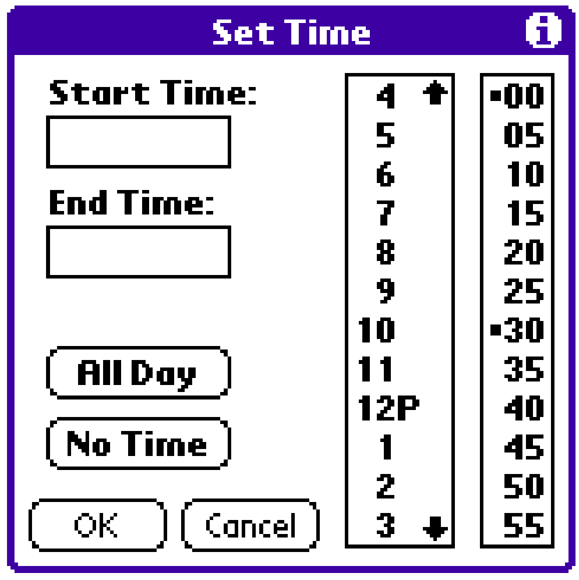


Figure 23- Old Interface

Alarm Window

Design

Our new Alarm window makes it easier for novices to understand the alarm feature and improves the usefulness of alarms. The checkbox to enable an alarm was replaced with radio buttons. We added the word "before" to indicate that the given time is an amount before the event. We changed the pull-down list to radio buttons for the "mins/hours/days" choice. Additionally, we added different alert types for users to choose from. A trade-off is that a new dialog increases complexity. Additionally, other relevant details, such as the event time, are now obscured.

Rationale

On/Off radio buttons: With the extra screen space provided by a dedicated dialog, the check box is replaced with radio buttons because they provide clearer labeling for the action.

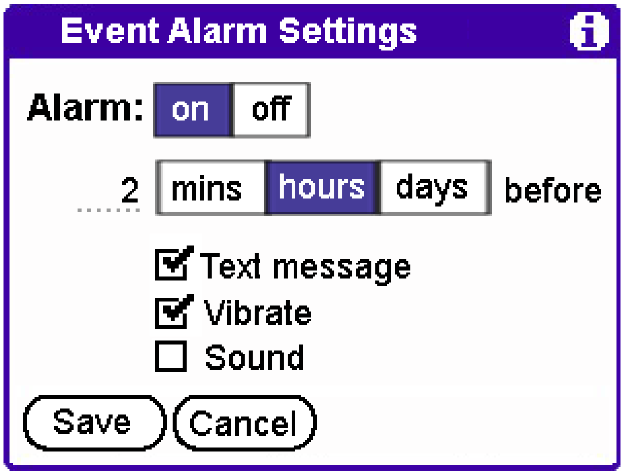
Units: Users need to know the units before they enter a quantity, as demonstrated in the think-aloud study where the user examined the pull-down menu before entering a quantity. Making the choices visible saves the user time. KLM analysis indicates that task time decreases using radio buttons. The unit 'days' is the only available option for a To Do since to-do's do not have time associated with them.

"before" label: Users may not understand whether the time specified is before or after the event. Adding the word "before" disambiguates this.

Multiple alarm modalities: During the CI, users wanted to be sure that the system would remind them of their events. These modalities are also an improvement according to the Flexibility and Efficiency of Use heuristic.

Design Change Aspect - Associated UARs	
Alarm Window	B-CW-04, BJC-HE-06, B-HE-04

Design Change Aspect - Motivating CI Evidence	
Alarm Window	L2-142, L2-258, L2-262



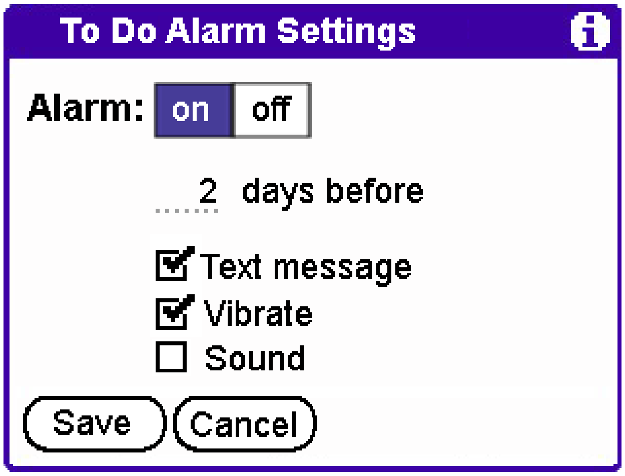
Event Alarm Settings

Alarm: [✓ on] off

2 mins [✓ hours] days before

☑ Text message
☑ Vibrate
□ Sound

Figure 24 - Redesign



To Do Alarm Settings

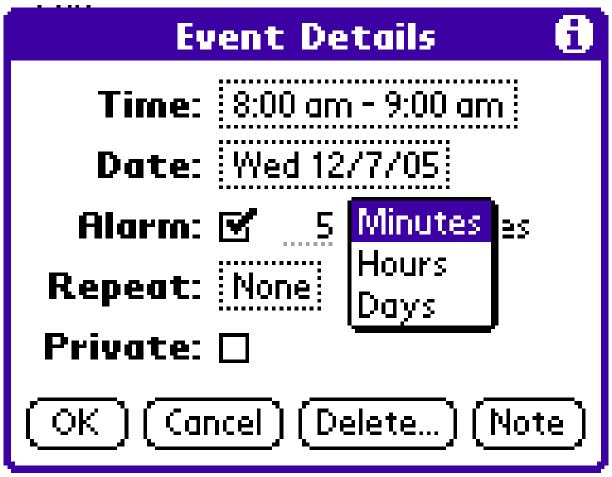
Alarm: [✓ on] off

2 days before

☑ Text message
☑ Vibrate
□ Sound

Figure 25 - Redesign

Alarm Window



Event Details

Time: 8:00 am - 9:00 am

Date: Wed 12/7/05

Alarm: ☑ 5 Minutes

Repeat: None

Private: □

Figure 26 - Old Interface

To Do Item Details Window

Design

We changed the Details window similarly to how we changed the Event Details window. This window is now used for creation as well as editing. The window shown is used for creating a new To Do (Figure 27). The ToDo details window is similar, but without "(New)" in the title, and has a Delete button at the bottom.

The item's name is included in the Details window, and a calendar picker is available in addition to text for setting due dates. We added Repeat and alarm features. We changed the button name from "OK" to "Save", clarifying its purpose. A trade-off is the user is forced through the dialog, even if they only want to provide a name.

Rationale

Name: This is required to maintain consistency with the Event Details window.

Due date: In collecting the information for a To Do into the Details window, we also added the due date.

Alarm: The user in the CI said that her reminders were not sufficiently "in her face". By adding an alarm, the system can actively remind the user before items must be completed, making the user more aware of the To Do items.

Repeat: In the CI, the user has items that recur or must be attended to periodically, such as "Boeing monthly report".

"Save" instead of "OK": Identical to corresponding change in Event Details window.

Design Change Aspect - Associated UARs	
Due date	B-CW-04, MEC-TA-07, JPC-TA-08

Design Change Aspect - Motivating CI Evidence	
Alarm	L2-142, L2-258, L2-262
Repeat	L3-311, L3-318, L022, L2-263, L074

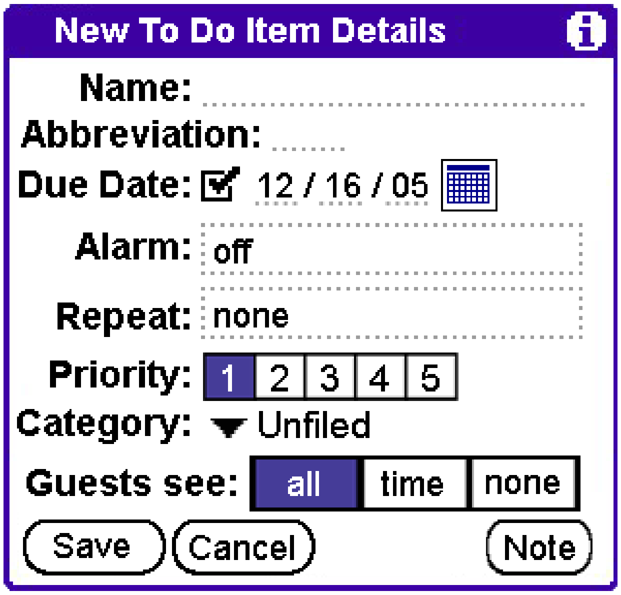


Figure 27 - Redesign

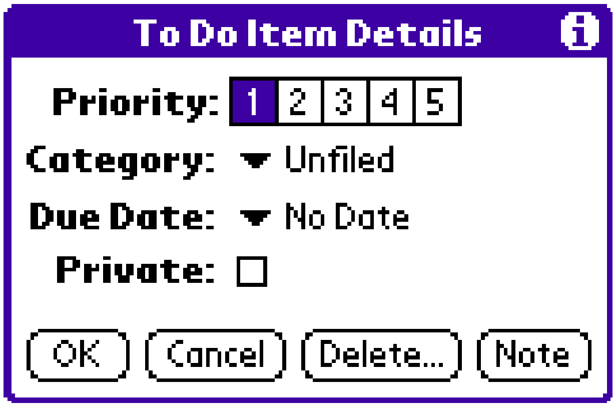


Figure 28 - Old Interface

Alarm Button

Design

On both the Day view and the To Do view, there is a new button labeled with the Palm alarm icon (Figures 29, 30). When tapped, the alarm dialog box displays for the selected to-do. (If no to-do is selected, the system displays the same warning that occurs when 'Details' is tapped without an event selected.)

Rationale

The main cause of error and frustration for the user in the TA was that the alarm function could not be located without great difficulty since it was embedded in Details. We solve this problem by adding an alarm button on the main screen so that it is clearly visible in the common views. Rather than label the button with 'Alarm,' we chose to use the alarm icon to save screen space. TA data shows that users will easily understand the meaning of the data.

Design Change Aspect - Motivating TA Evidence	
Alarm Button	think_aloud_task_segment.mov 07:16, MEC-TA-05, MEC-TA-08, JPC-TA-11, JPC-TA-12, MJC-TA-5, JPC-TA-8

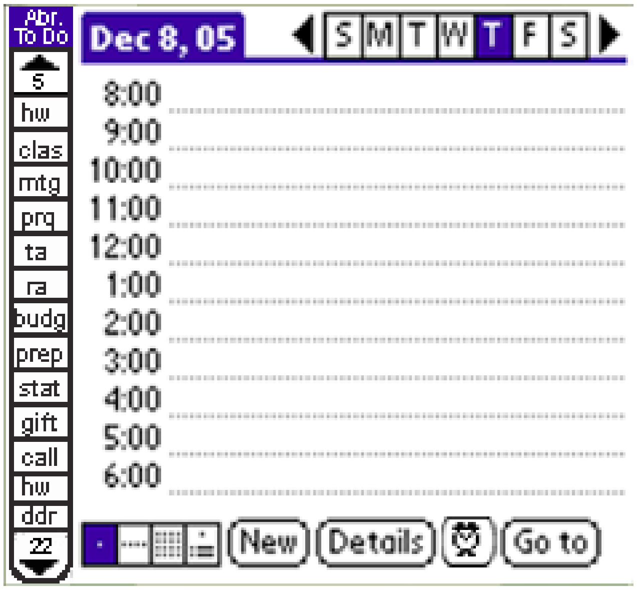
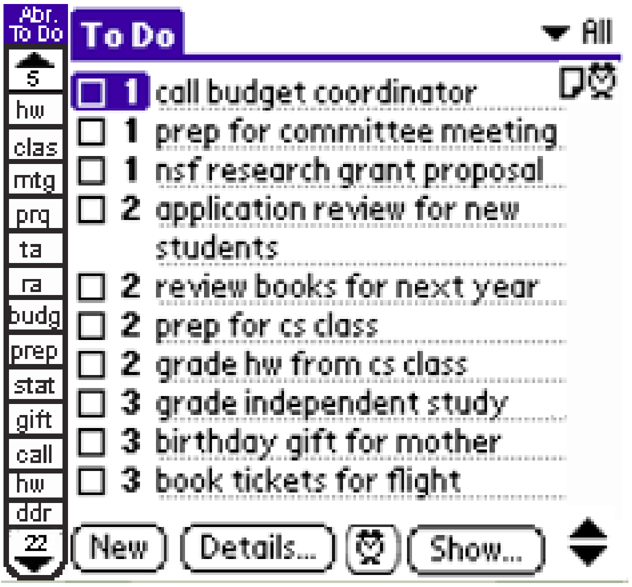


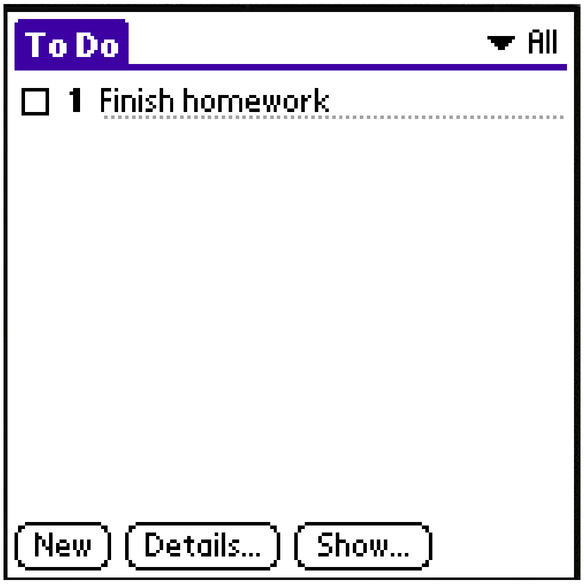
Figure 29 - Redesign



☐ 1 call budget coordinator
☐ 1 prep for committee meeting
☐ 1 nsf research grant proposal
☐ 2 application review for new students
☐ 2 review books for next year
☐ 2 prep for cs class
☐ 2 grade hw from cs class
☐ 3 grade independent study
☐ 3 birthday gift for mother
☐ 3 book tickets for flight

Figure 30 - Redesign

Alarm Button



☐ 1 Finish homework

Figure 31 - Old Interface

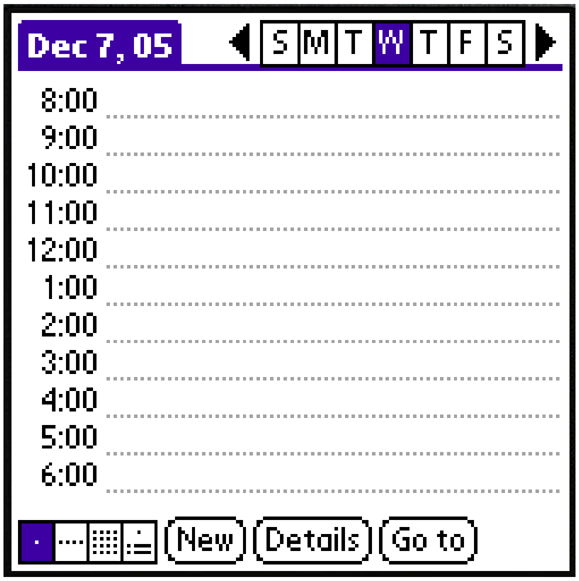


Figure 32 - Old Interface

Retrospective

Focus Setting

It appears that this technique would be very useful in practice. Brainstorming many questions about what information could be discovered provides much better breadth of the topic than just trying to pick foci directly. The affinity diagramming process makes it easier to see where the emphasis probably should be through the clustering. It also makes it easier to see where gaps in the information might be, allowing more questions to be generated before even the foci are chosen. Being explicit about what is not in the foci makes it much easier to avoid wasting time by unintentionally pursuing irrelevant investigations.

Marketing departments typically determine product goals. This technique may be perceived as performing a similar purpose, which marketing may interpret as a threat to their authority. Usability analysts would need to clearly communicate the purposes of focus setting to get buy-in in this case.

Contextual Inquiry & Design

The first techniques we learned in this course were contextual inquiry and contextual design (CI/CD). The goal of the former is to gather data from users in the context of the work that they do. The latter provides a mechanism for representing this data in a salient way through the creation of different models. Gathering data in context allows us to gain insights into things such as the many tasks users do, the people they interact with, the environment in which they work, and cultural influences.

Overall we found this method to be very useful. Observing how someone does To Do management and modeling this data allowed us to come up with novel design ideas. For example, something we encountered in the data was that current To Do mechanisms were not “in the user’s face” enough. This prompted us to invent the abbreviated To Do list, in which the content and number of To Do’s are always visible. If we had not done the CI/CD, we would likely not have seen this need.

This method definitely provides many benefits, but a certain amount of balancing has to be done to ensure that these benefits outweigh the relatively high costs. The process of interviewing, creating models, and designing from data is very time-consuming, as we observed and experienced during our first two assignments. However, the time required to perform these methods is not the only associated cost, since the process of obtaining users is also expensive. When performing CI/CD, care must be taken when selecting the number and type(s) of users. Enough users are required to obtain reliably representative data, but not so many that the high cost outweighs the gain.

We also found that data from CI/CD is not only useful for inspiring new design ideas, but also for validating otherwise inspired design ideas. Thus, for sudden design ideas, we could then look back to the models to see if a design idea was supported by data. For example, we had an idea to create a zooming interface, which would allow the user to dynamically see To Do’s in a hierarchical fashion based on the due date. We drew up sketches as to how this would work and were intrigued by its possibilities, but we did not include it in our final design, since it was not supported by the CI/CD data.

The type of data provided by this method is very different from that provided by the other methods. The other methods have, as a primary goal, evaluating existing interfaces. However, CI/CD data is not necessarily tied to any particular interface, and is not primarily intended for evaluation. Instead, it focuses on building models of work so that experts can design new systems to support it. Gathering data in the context of the user’s work environment provides data that no other method we learned can emulate.

Keystroke Level Model

The next technique we learned was the keystroke level model (KLM), which is from the GOMS family. KLM is used to predict the time a skilled user would require to perform a task. The model takes as input an interface design and a series of actions to be performed, which constitutes a certain benchmark task. Based on this input, the model outputs the execution time for a skilled user. In our case, modeling was done with the CogTool software, which automates the process of creating a KLM. KLM is analytical, i.e. its data is based on theory and does not require real users.

Since this method requires an interface, working or prototyped, for which a benchmark task can be constructed, it is used in the detailed design/implementation phase of the software lifecycle. KLM provides information about how long users need to complete tasks in an interface. As such, it is useful for comparing task completion times of alternative interfaces. The model provides justification for its judgments, which can be very valuable in improving efficiency of an existing interface. In particular, KLM calculates a time for each step in the task.

Unlike all the other methods we have learned, KLM provides objective data whose validity is backed by scientific studies. This is in contrast to HE and CW, where evaluators must subjectively determine where the user will encounter problems. It is also in contrast to CI and TA where the data may be slightly unreliable because their particular data may be representative of only one user and not a larger class of people.

While using this method, we noticed that it leads to significantly less contention among evaluators than the other methods, since it is objective. When using the other usability evaluation methods, we frequently disagreed about severities of problems. One change we made in our redesign was to use radio buttons instead of a pull down list for setting alarm units. Several of us found, on the KLM homework, that radio buttons are faster and do not require much additional screen space, so we all quickly agreed to make this design change.

We did not focus on KLM for our redesign because our primary goals were unrelated to the efficiency of expert users. Also, although CogTool makes the process of creating a KLM faster and easier, it is still a time consuming process. We found that CogTool makes the process of creating KLM models much more feasible than it would be otherwise, but it does not make the process so easy that it will be used casually for tasks that are not very time intensive

Heuristic Evaluation

Heuristic evaluation (HE) is a usability inspection method in which a team of evaluators combs through an interface in order to see whether it meets or violates a predetermined list of principles of good design. It is task independent, so evaluators are not constrained to one particular user task. We found that TA and CW encouraged us to focus on only a small subset of the interface that corresponded to a certain task. Heuristic evaluation allowed us to freely navigate the interface. However, even though HE is task independent, when applying the method, we found it helpful to informally consider tasks while analyzing the interface. This approach helped us perform complete, careful evaluations of the most frequently used areas.

We found HE generated the most usability problems in the shortest amount of time. We also found it easier to learn and apply than CW and TA. Another strength of this method is that it can be applied to paper prototypes as well as software prototypes and functioning systems. CW also has this strength, but for a TA, one needs at least a partially functioning prototype.

The ease of use, ability to quickly find many problems, and flexibility of applying it prompted us to use this method when evaluating some of our redesigns. In particular, we needed a quick way to test our abbreviated To Do list, to make sure it did not have any major problems. Upon doing the HE we found 4 heuristic violations, one a major usability problem. This fact exhibits two important lessons. First, even when designers are familiar with the HE heuristics, their designs may still contain violations. Second, it is possible, while perhaps not optimal, for designers to do an HE on their own work.

One criticism of this method is that evaluators may generate many minor problems. However, we found that applying severities to all the issues found helped mitigate this problem. With severity ratings, we could consider the most serious issues first, and the smaller problems (some of which may not really be problems when considering the context) last or not at all.
Another potential objection to the value of this method is that it is not always straightforward as to how to apply

the heuristics to novel program spaces. We found that it is highly effective for traditional GUI interfaces, but suspect that it is not as useful for other types of interfaces. For example, a device with speech input and no actual visual display would be very hard to do an HE on; this context may necessitate new or modified heuristics. It seems CW or TA may be more appropriate to perform in these types of situations.

Cognitive Walkthrough

A cognitive walkthrough evaluates how well an interface supports learning by exploration. In a cognitive walkthrough, a group of evaluators proceed through a task, answering four questions about the interface and a hypothetical user at each step in the task. This could also be done by an individual, but better results are typically produced by a group. Researchers developed the questions based on a theory of learning.

Unfortunately, we found the negative aspects of this technique to be the most prominent, if not necessarily the most important. The questions are nuanced and complex, making this a rather difficult technique to use correctly. It appeared to us that CW has the least benefit for a given amount of effort. The obvious products are the Usability Aspect Reports for failures, of which there were few. However, since the technique involves thoroughly going through a task, the lack of UAR's is as valid an acclamation of the parts that succeeded as the UAR's are a criticism of the parts that failed.

This success information is captured by the success stories that are part of the method, but because of their lack of structure, they do not seem to be as accessible to other product team members as the UAR's are. This is unfortunate for at least two reasons. It is harder to communicate the value of something without having concrete deliverables. Also, it becomes harder to ensure that the good aspects, i.e. the reasons for the success stories, are not broken or removed. This is especially unfortunate because one of the major benefits is that this technique can be used very early in the product cycle, requiring only paper prototypes or storyboards.

Another aspect we felt was a shortcoming of this technique was the absolute way in which questions had to be answered. In several cases, reasonable theories were found that supported both “yes” and “no” answers. Being forced to pick one ignores the subtle but relevant ways in which even very similar users may differ. These situations can be argued through to eventually resolve one case as being more salient, but this can require a protracted process. It seems that it would be more cost effective to let both answers stand: both may provide value. Future investigations may resolve one answer to be “correct” at little additional expense.

The importance of task selection may also be considered somewhat of a shortcoming, though cognitive walkthrough is hardly unique in this respect. With the size of most applications and the duration of many product cycles, let alone the availability of usability expertise, there will hardly be the chance of analyzing the entire product. More than likely, only a few tasks could be analyzed. Unfortunately, one of the benefits of this technique -- the ability to use it very early in the product cycle -- aggravates this issue. For real products, as the cycle proceeds, requirements and users change. It is quite possible that tasks that were very important at the beginning of the cycle are much less significant at the end.

Despite the strong recommendation to use this technique in a group setting, bending this “rule” may mitigate the cost of time and expertise. The most important tasks can be done first, in a group setting. This can help establish an agreed context for the CW. More importantly, it will generate a good user description and set of assumptions. Then, less important tasks can be analyzed by individuals using the user description and assumptions generated from the group work. Probably, subtle points will be missed by the individual analysis, but the major failures should still be found. And since the work is distributed, more tasks could be analyzed. If desired, success stories could be verified by other individuals to increase reliability.

This modification of the method briefly touched on one of the other valuable, but possibly still subtle, aspects of the technique: the user description and assumptions. Making them explicit has two primary benefits. First, they can be verified against the real world. Second, they can be more clearly communicated. Having the correct assumptions does no good if the rest of the team has other assumptions. The ability to use this technique early compounds this value since potentially wasteful work can be avoided early.

There is a negative aspect tightly coupled with the user description and assumptions. An actual user is not completing the task so it is the analysts who are applying the assumptions to determine what a user would do. This subjectivity opens the results to debate about validity in cases. This can be mitigated by using analysts with more expertise, but that, of course, comes at a cost. The subjectivity of CW is in contrast to KLM, which provides an objective measure.

Cognitive walkthrough is certainly a beneficial technique. For some application spaces, especially "walk up and use" applications like kiosks, it may be indispensable. But in general, it is unclear whether its perceived and actual benefits outweigh its expense of time and required expertise.

Think Aloud

In a think aloud study, users are given a task to complete while continually talking about what they are doing and why, i.e. what they are thinking. The analyst observes this, noting important events based on criteria that determine when something definitely good or bad happened. One justification for the validity of the results of this study lies in cognitive science. Since the user is simply verbalizing her current information state, she is able to accurately express her thoughts. Of course, there is a significant time cost to verbalizing thoughts, so this method is less directly applicable to time-sensitive systems.

We felt that there were several major benefits present with this method. First, the experience of a real person actually encountering a problem (or good aspect) is very concrete and compelling. It can make it much easier to convince others that something is a problem. The criteria aid this by being fairly clear and definite, resulting in much less debate about the validity of the findings. In cases of failure, seeing what the user tried to do can provide valuable insight into improving the design. It also seemed like more of the problems found were of a higher severity.

Probably the most important benefit, however, is that this technique addresses the issue of the "user is not like me". No matter how much planning and research is done, the user will always do something unanticipated. This is in contrast to the other methods, which, being inspection methods, cannot provide the benefit of a real user.

The specific example we experienced highlighted some of the negative aspects of this technique. Sometimes a user behaves so unexpectedly that one is reticent to accept the evidence at face value and rather considers the data an outlier. We suspect that careful selection of users will increase confidence in the validity of users' actions, regardless of the particular behavior. Unfortunately, this only exacerbates the difficulty of getting users to participate in the first place. Evaluating the interface with more users can also improve this confidence; of course, this also increases expenses.

Once users are found, there may still be difficulties. Thinking aloud in the level of detail that is required is not especially natural. Other experiences with this technique have shown that some users have trouble thinking aloud and either continually fall silent or may simply read aloud whatever is on the screen. Multiple users can actually increase subjectivity in this method in the case of determining the salience of an issue encountered by only a small proportion of the users.

This technique shares a negative aspect with the cognitive walkthrough: the importance of task selection. Due to the expense of this technique, a limited number of tasks can be evaluated. Conversely to the cognitive walkthrough, the negative aspect of this technique not being applicable until later in the product cycle mitigates this effect. This is because later in the cycle, it is likely to be clearer what the important tasks are.

Despite difficulties involving obtaining and testing against real users, think-aloud may be the most valuable evaluation method. In particular, it finds many unanticipated problems; also, it is more likely to find higher severity problems. Furthermore, it provides qualitative data that can inspire design improvements. Therefore, we appreciate why TA is considered to be the gold standard of evaluation methods.