Interactive Prototype Report

Project Title: ETA – Electronic Travel Assistance for Group 4.
Platform: Microsoft PocketPC (HP Jornada) – running Sun Microsystems’ Java Chai.
Online Doc.: http://ratbert.bmrc.berkeley.edu/ ... /Group4/assignment3.html

Problem and Solution Overview

Many young adults and college students budget travel to a large number of cities in a short period of time. They backpack in various cities of Europe and Asia on a limited budget, and this generally involves seeing and doing as many things in a city in only a few days. One of the most common resources they have at hand today is tour books such as *Let's Go Europe* and *Lonely Planet*. Although, these resources present a lot of information to the traveler, they are not tailored to their specific interests and budget nor are they very up-to-date. For example, a typical traveler might need days to research and organize information to create a personal itinerary. ETA bridges the gap left by conventional resources by managing travel information for the user. And, it does so in two ways. First, ETA gives users quick access to up-to-date travel information, such as accommodations, dining, and entertainment, through synchronization with the Internet. Secondly, it automatically creates travel itineraries based on the user’s preferences and interests, and ETA dynamically adjusts them to bring the user within budget.

Tasks

We wanted our tasks to be representative of the program features and also be able to string together a good mental model of project for the users. As a result, we have chosen three appropriate tasks with different levels of complexity for our participants to accomplish.

The first task is to create the user’s profile in which they entered in their preferences and interests. This task is considered “hard” because it requires the most steps to complete and leaves room for many errors. The second task is to find an accommodation in a specific city. This task is labeled “medium” since it requires less steps than the first but is more of a cognitive process than the next. The last task is to locate a particular destination on a map. This one is considered “easy” since it requires the fewest steps.

After careful consideration and analysis of our low fidelity prototyping experiments, we have kept our original tasks because they adequately cover the proposed functionality of our program. As mentioned in the Solutions section above, the itinerary is a major function that distinguishes our program from the conventional tour books, in that our program is intelligent enough to suggest points of interest for users who do not have the time to look everything up. And, filling the profile is one of the ways in which the computer can assess what the user desires and is inclined in doing. The accommodation and mapping tasks have been chosen simply because of their overwhelming popularity with the travelers whom we have interviewed at a San Francisco hostel, during the first phase project analysis.

Revised Interface Design

Differences As a Result of Low-Fi Testing & Why

The changes that we made to our interactive prototype came from solutions to the problems faced during the low fidelity testing in the first assignment. After the experiments were done, we collated and ranked
the critical incidents according to their severity and executed appropriate measures to fix them in our next version, the interactive prototype. Below, you will find the three tasks that our users faced, their major problems, and our formulated solutions to alleviate these problems via changes to ETA’s interface.

Task 1: Creating an Initial User Profile

Problem: Low fidelity testing told us that the users were quite perplexed on how to start the User Profile section. For example, they did not recognize the “start” arrow that was meant to lead them into the Profile Wizard. Some of them attempted to click on User Info to start the Profile Wizard, which was not correct. During the Profile stage, users were also confused on the Budget section. Originally, we intended the question to ask, “What is your desired budget goal for the whole stay?” However, they thought it was for a “day-to-day” basis. The To-Do menu system in the original hid a lot of function from our users in an attempt to make the system more efficient. However, this caused much confusion on the user’s part because they had a different mental model for what a “To-Do” was. Finally, the multiple selection menu used in the low fidelity model made it hard on the users because they were not familiar with Jornada’s click-and-drag type of operation, where you hold down the stylus and drag over your selections to take in multiple choices. Some users thought they could make only one choice of food, for example, in their profile.

Solution: We eliminated the confusing menu system from our Profile section and started the program on the general menu instead of making the user create profiles at the beginning. So, we allowed them to use ETA without creating an initial profile. In addition, we removed profile questions that were not relevant to the purpose of the program, such as the user’s address, number of people traveling, and rephrased questions to be more informative and less restrictive. The revamp of the budget section is a great example of this. There, we eliminated information that confused the user and that did not contribute to the functionality of ETA. We also removed the To-Do button and gave them a dropdown menu listed under Tools instead. We introduced a Home button for easy navigation and incorporated good suggestions from users regard the use of multiple selections. Here, we used check boxes instead of the old click-and-drag method to select choices, which is more intuitive.

Task 2: Finding an Accommodation

Problem: Almost on every trial, the users pressed the Accommodation button on the Profiles sub-menu, which was incorrect. What the system expected them to do was to go to the “To-Do” menu and choose Profile from there. They also were confused on “how” to reserve an accommodation once they have made their choice. As a result of this confusion, some of them attempted to bookmark their selection in hopes that this action would somehow lead them on the correct path. They were also confused on the rating system. Originally, they thought that the rating was similar to the type of system used to judge the “quality” of fine hotels, as in a “Five Star” hotel. However, this rating system was done by and for ETA users. Finally, they wanted to “see” their selection after they chose their accommodation, a functionality that was not present in the low fidelity version.

Solution: The very first thing we addressed was the sub-menu issue. We removed the sub-menu and added it to the main menu under View to avoid causing confusion. The Accommodations button was moved into the Tools section as suggested by our users. The bookmark option was removed because users equated this with reservations, which was not supported by our system. Instead, ETA now provides a quick way to get detailed information about a current selection and a mapping to go along with it. Users can now click Show Map, and ETA would display a zoomable diagram of where that particular location is. We also made improvements to our rating system by...
adding the typical cost of lodging, to further differentiate its “quality” (i.e. “Five-star,” “Four-star”) from the experience that it provides the users.

Task 3: Mapping a Location

Problem: This last task presented fewer problems than the ones above, partially because it was the simplest of the three. However, that is not to say that it was immune from difficulties. Although, the user did not initially describe any issues with the mapping feature, they did talk about it afterwards as a sidebar. They told us that they had minor irks with the main page for mapping, in that it was a bit disorienting because they were presented with question right away. They wanted some way of getting familiar with the ETA’s mapping abilities first before they plunged into finding a location. They also said that the address section was cumbersome because they had to enter a lot of information that was not necessary, the zip code. The section also did not address the issue of international destinations, an ability that the low fidelity version blatantly lacked.

Solution: To alleviate the issue of disorientation, we incorporated three options on the first main page, describing what ETA’s mapping functions can do. This quickly orients the user and gives them time to think about what to do, instead of barraging them with questions. From there, they can choose to quickly find the City Center, Find a Location, or get Bus/Driving Directions. If they choose Find a Location, they will move to a page where they can easily enter their destination address. The user can either enter just the city and state or just the zip code. They also have the option of choosing a location outside the United States.

Scenario For Three Tasks

Storyboard for Task 1: Creating a Profile
Storyboard for Task 2: Finding an Accommodation

Task 1 - Creating a Profile:
- Third page of profile section.
- Fourth and final page of profile section.

Task 2 - Finding a place to stay:
- Listings for lodgings.
- Details page for a specific lodging.
Storyboard for Task 3: Mapping a Specific Location

**Prototype Overview**

**Tools We Used**

- **Benefits From Using the Development Tools**
  - Electronic Travel Assistant for Hi-Fi prototype was written in Java to be run on Chai, the Java Virtual Machine on HP Jornada. We worked with WebGain's Visual Café 4.5 Standard Edition to generate Java Graphical User Interface code by a simple drag-and-drop. It allowed us for a faster development by minimizing the need for handwritten code. For example, using WYSWYG feature of Visual Café, we were able to generate many lines of Java code by drag-and-dropping various AWT features such as menu bar, radio buttons and check boxes onto the Frame. In addition, Visual Café provided us with a project manager, which for example listed all Java files that we were editing to allow us a better navigation capability among the source files in case we needed to refer to other files. In order to check for the incompatibility of Chai with Windows Virtual Machine, we used a software tool provided by HP that notified us for possible incompatibilities. Then, we compiled the Java code in Visual Café IDE to transfer the binary class files onto Jornada using ActiveSync. ActiveSync interface also gave us a complete control over our memory usage in Jornada.

- **Limitations of Using the Tools**
  - Although the Visual Café IDE allowed us to develop the AWT user interface relatively quick, it did not have the "Chai compatible" AWT mode. In other words, we did not know whether the generated code would be compatible with Chai Virtual Machine until we ran the compatibility checking software mentioned in previous paragraph. Once we spotted out any incompatibilities using the tool, we needed to go back and change or remove the code. On a same note, Visual Café did not provide us with the actual view of how it would look running on Chai because it was not tightly integrated with the IDE. In addition, it was sometimes impossible for us to tweak minor details of user interface using the WYSWYG capability of Visual Café. Thus, we needed to change the code manually in order to control the interface in details. It was a difficult task due to the sheer size of
Overview of UI Implemented

Electronic Travel Assistant starts from a page (aka homepage) that contains tool icons like shown in Figure A. The interface was revised since last prototype to guide novice users for easier and faster navigations. The previous prototype for ETA required the first time users to create profiles and it confused many users from our user testing. In addition, some users wanted to proceed without creating a user profile. We tried to pick the most suggestive and intuitive icons for the first time and novice users.

Coupled with the homepage is a global menu bar shown in Figure B that replicates all the features and capabilities of the homepage and allows users to quickly navigate through features and sections without having to return to the home page every time he/she wants to use a different section. This also has been revised since our last prototype, where each screen contained both a unique menu for that screen along with a global menu. The old global menu confused every single one of our testers. In this prototype, we grouped and simplified the global menu and completely removed the internal menu to bring consistency in design from one screen to another like shown in Screenshot 2, 3, and 4 in next section. For example, Screenshot 2 (Figure B) demonstrates the ability to quickly change the city using the toolbar. The global toolbar is accessible from every screen, which means users can quickly change the destination city from any screens without having to go back to the home screen.

The first icon listed in the homepage as seen on Figure A is CITY INFO, which gives a short overview of the history and culture of the city like shown in Screenshot 5. We decided to put the icon in the first row, leftmost column because if the user travels or plans to travel anywhere, one would try to get the information about the city. Next on the home page is the map section, when the user selects that feature, from the home page icon or the menu bar, he/she is taken to a screen where has the option to get a map of the city center, map a location, or get driving directions as shown in Task 3 screenshots.

Next are the most widely used features of ETA, lodging, dining, entertainment, and landmarks. We believed that the position of these icons in the central part of the screen is appropriate as most users will use them frequently and therefore easy access is important. To be consistent within the application, we decided to format these four sections in a similar fashion using a single template. Thus, it will help the novice users to easily familiarize themselves with four features by learning a single feature.
When a user selects anyone of these four features, the user is presented with suggested listings using the relevant data from user's profile like shown in Figure C, or screenshots 7 and 8 from the next section. In addition to the suggested listing, the user has the capability of getting a complete listing as well as to view user's bookmarked items.

When a user selects an item from the list, the user gets a new screen with more detailed information about selected item. For example, when a user clicked on a hotel Durant from a list of hotels like shown in first screen of Task2 section in next section, the user will see the details of the Hotel Durant. In addition, the user could map the selected place and possibly bookmark the place to easily refer to them later as shown in the Figure D. If the user wanted to map the Hotel Durant from last example, it would invoke the Map functionality with the address of Hotel Durant as a visible parameter as shown in the last screenshot of Task 2.

In order to have a consistent user interface, when the user chooses to map the item, it is displayed in the same format as in the "map a location" option of the map feature. The user will also be given the option to bookmark the item for later consideration or to just go back to the listing.

We also implemented creating a profile function. It takes the users through five screens to get specific information about the user as shown in Figure E below. The profile information collects user's preferences on lodging, meal, entertainment, and budget as shown in Figure E. This information is used by the application to make the most suitable suggestions based on the particular user's preferences as mentioned earlier. We changed some of the interfaces from this section from last prototype. For example, in our previous prototype they were presented with a list of preferences and a lot of users failed to recognize that one could pick more than one item from the list. More specifically, we had a list of many food types and many of the testers thought that they could only pick one of the food types. Thus, in this prototype we replaced the list with the checkboxes to avoid further confusion as shown in Figure E. We also added the capability of navigating back using a back button and a next button.

**What Was Left Out and Why**

Other features presented on the homepage will be implemented at a later stage of the prototype. For example, the search feature will allow users to search for accommodation, dining, entertainment or landmarks based on a set of constraints they provide. Thus, the particular feature will prevent users from being restricted by preset preferences on his or her profile. The feature was left out since we believed that the functions that it performs overlap with other features like finding accommodation from suggested listings.

Safety/emergency feature will come in handy in case of an emergency in foreign places. It will guide the user in seeking the
appropriate assistance, and possibly in solving the problem. The feature was left out because we thought that the functionality of this feature is not interconnected to any other features of the application. Thus, given the time constraint, we decided to focus on implementing features that are more critical in defining the user experience of the application.

Similarly, finding a driving direction and mapping the city center from the map section was left out. Mapping the city center function was very similar to simple mapping function such as mapping a particular hotel.

**Wizard of OZ Techniques Used**

We employed the particular technique many times to make the prototype work. For example, the application does not calculate the best-suited hotels from a large database using user's profile. Instead, it simply prints out the hard-coded preset data. In fact, we do not even save the user's preferences from profile. In addition, the map that we present is not a map that is calculated for every different place. It only presents one map that is hard-coded.

**Prototype Screen Shots**

We have eliminated redundancy by only including pictures that we have not displayed in the in-line screen shots. These screen shots help to illustrate the general, overall picture of ETA. They are annotated for your connivance.

![Screenshot 3: "Options" menu.](image1)

![Screenshot 4: "Help" menu.](image2)

![Screenshot 5: "City Info" page. The home link is in the upper right.](image3)

![Screenshot 7: "Sights" section.](image4)

![Screenshot 8: "Entertainment" section.](image5)