E-TRAVEL Human Interface Analysis

The goal in designing the human interface is to improve the interface such that:

- the product is integrated with the user's work.
- the product adheres to accepted human interface guidelines.
- the amount of user documentation, training, and support can be reduced.

If the interface is optimized, it will require fewer add-ons, such as documentation and support. This increases user satisfaction and user productivity. It is important to remember that the system exists to help the user accomplish set goals. Providing a lot of features and functionality will not make products more successful; as products do more things, they become more complex and harder to use. Thus, the interface must make the product as intuitive as possible.

Integration and Productivity

Systems fail because they're not integrated into peoples' work. Effective interfaces attempt to anticipate what the user wants and needs and bring to the user the tools needed for each step. It is necessary to look at the user's productivity, not at the system's. The system should perform a maximum of work while requiring a minimum of user information. The system should help the user accomplish his or her goals as quickly as possible. Not only should the system itself be made faster, but it should incorporate good metaphors, enabling users to learn more quickly. The user should be kept occupied. Time spent waiting is time spent not working. If a user *must* wait, the system should show evidence of work progression.

Time spent learning a system is time that a user is not actively working. The system should incorporate instructions on the interface. The system should provide clear, understandable, and accurate information regarding the task that it is performing. The system should not concern the user with inner workings of the system. How the system is performing a task is not as important to the user as what task is being performed and how long that task will take to complete.

Web Traveler should be designed with frequent travelers in mind. A "frequent traveler" is defined as a person who makes five or more trips per year. Thus, some of the users will only use the system five times a year. Some users may use it even less. It is unclear how often Web Admin will be used. The less frequently the system is used, the more instructive the interface must be. Knowledge needs to be embedded in the interface. Thus, support can be provided without breaking the context of the task to be completed.

To further increase productivity, the system should track state—users should be able to log off, go home, and pick up exactly where they left off.

Human Interface Guidelines and Productivity

Most importantly, an effective interface must employ consistent use of visual conventions, language, navigation, and other system behavior.

An effective interface is:

- safe and supportive for the naïve, but not condescending to the experienced. First time and sporadic users of the system will need more support than power-users. It is important to make the interface understandable by all users.

- visually apparent and forgiving.
  Users must be able to figure out what to do by looking at the interface. If they make a mistake, they must be able to undo what they last did.
- controlled by the user.
  Users must feel as though they are in the same place, with information being brought to them.
- efficient in the architecture and the front end.
  Efficiency breakthroughs are in the architecture, not in the front end design. The system itself must be designed with the user in mind.
- readable.
  Font and font size must be carefully chosen to maximize readability. In addition, there must be effective contrast between background and text.
- frames-free.
  Frames make printing difficult, are slower to load, and make users guess where things will appear.

Further, effective interfaces must:

- allow undo.
  Users must always be able to undo their last action. If they cannot undo the last action, then the interface must warn them that what they are about to do is irreversible.
- save work.
  Users must be able to come back to the system and see what they have done. In the event of a system or computer crash, as much information as possible must be saved. If any information is lost, this fact must be communicated to the user.
- make objects that do different things look different.
  Links should produce different results than buttons, which should produce different results than icons. Thus, users can construct a model of the system and how it behaves.
- give users a path through the system, but always allow a way out.
  If there is a preferred path through the system, the system should provide that path. While it is important that users be able to control their experience, it is also important that they are efficient. If one path is more efficient than any other, that path should be stressed, while allowing the user to go back or skip ahead. If a path will cause the user to lose information or may cause undesired events, then that should be communicated to the user.
- employ visible, natural navigation.
  Users must be able to see where they are going. The steps to accomplish a goal must be direct.
- set context.
  The system must structure work process and progression through tasks and logic.

Decrease the Amount of Extrinsic Support

Much of the documentation work that is currently done is explaining poor design. Performance-based design is driven by tasks that are performed by users. This results in:

- no need for training.
- a decrease in the need for users to memorize.
- an interface that is congruent with the way that work is done, making it familiar to users.

- an increase in time saved.
- an increase in employees accepting the system.

Power-users side-step instructions; therefore, the interface should separate "what" instructions from "how" instructions. For example, all users need to know that they can book a business trip. However, power-users may not need or want any instructions. Some basic instructions can be incorporated into the interface, simply by giving buttons and links intuitive names. These are "what" instructions—"What do I click? What do I do next?" Some users will need some more handholding, such as "To book a flight, first click the Business Trip link." These are "how" instructions—"How do I do this?" Brief "how" instructions can be provided on-screen, in a position that will not interfere with the power-users' ability to use the system. Context-sensitive Help can fulfill the further requirements for "how" instructions.

While completely eliminating Help is not possible, nor desirable, a well-designed interface can reduce the need for Help. It can also help to eliminate the need for large user manuals, online documentation, support calls, and training.

Current Status

The current interfaces do not meet any of the aforementioned guidelines.

- The system has been designed around the database and the CRS, rather than around user tasks.
- The system does not make clear what information is required and what information is optional. Nor does it make clear what can be entered (letters only, numbers only, etc.).
- Neither Web Admin nor Web Traveler employ any visible metaphors.
- Web Admin depends on Java applets that can take up to two minutes to load. No system progression is shown.
- Very little in terms of instruction is provided on-screen. Often, the text that is provided is inconsistent and difficult to read.
- The system does not track state as well as it might. A user cannot pick up where he or she left off when making a reservation, for example.
- The system does not employ a consistent use of visual conventions, language, navigation, and other system behavior. Some portions of Web Admin are Java-based, some are form-based. Adding and deleting items differ from one page to the next. The same options may be named differently both between and within modules (for example "Service Class").
- The system is not safe if a person doesn't know exactly what he or she is doing. In Web Admin, it is all too easy to corrupt the database. In Web Traveler, a user may think he or she has booked a reservation when that reservation has not been sent, or vice versa.
- Users of Web Traveler do not have any method of undo for preferences information or canceling a trip. Few warnings are provided. When canceling a trip, the only option in the given warning is **OK**.
- Each product looks different from page to page.
- Web Traveler employs a very large font size that may cause the **Navigation** menu to be cut off. Instructional text is small and italicized. Web Admin employs a lot of italicized text, which is harder to read.
- Web Traveler employs frames, which are used inconsistently.

- If the system crashes or a PNR is modified outside of Web Traveler, any changes are not saved and a user cannot open his or her itinerary in Web Traveler.
- Links, buttons, and icons behave inconsistently. In Web Admin, buttons may appear to be different colors from page to page, depending on whether the page is HTML- or Java-based.
- Users are not given a path through the system. In Web Admin, this could cause companies or agencies to be set up incorrectly. In Web Traveler, this could result in reservations being made or changed without the user's knowledge.
- The system allows the user to complete tasks that could harm them. For example, if a user makes a Car reservation before an Air reservation, the Car reservation could be changed or cancelled without the user's knowledge.

Proposed Changes

- Determine what users do with both products. Web Traveler tasks are relatively transparent, but it is important to know who does which tasks and how often in Web Admin.
- Employ a consistent format for required information. Perhaps all required fields are marked with an asterisk, or the field is pre-filled with the word "required." If a field can only contain letters, numbers, or a combination thereof, this information should be communicated to users. Perhaps these fields could be pre-filled with the word "alphanumeric" or "1" for numbers.
- Create consistent metaphors for both Web Admin and Web Traveler. In Web Traveler, it is possible to create a travel agent-type metaphor that will not get in the way. Web Admin should employ a metaphor that is equally as recognizable to its audiences.
- Eliminate Java.
- Provide on-screen instructions, written by a documentation specialist.
- Track state as much as possible.
- Ensure that buttons are the same color from page to page, and that clicking a button produces the same type of result. (The system does something with the information that has been given.)
- Ensure that clicking links produces the same kind of result. (The system takes a user to another page.)
- Ensure that all buttons and options that produce the same results are consistent in name.
- Warn users before they take an action that may harm them. Make sure that that action can be cancelled before anything happens.
- All options should be undoable, and if they aren't, a clear warning message that allows **Cancel** should be displayed.
- Each page should employ a consistent look.
- Sans-serif fonts, which are easier to read online, should be used. Fonts should be at least 10 point. Nothing should be italicized.
- Web Traveler should use tables instead of frames.
- Users should be provided with a clear path through the system, especially in Web Traveler, since performing tasks out of order (or not at all) can greatly affect a user's reservations.