New Approaches to Help Users Get Started with Visual Interfaces: Multi-Layered Interfaces and Integrated Initial Guidance

Hyunmo Kang, Catherine Plaisant and Ben Shneiderman
Department of Computer Science & Human Computer Interaction Laboratory
University of Maryland, College Park, MD 20742
{kang, plaisant, ben}@cs.umd.edu

Abstract

We are investigating new ways to help users learn to use public access interactive tools, in particular for the visual exploration of government statistics. Our work led to a series of interfaces using multi-layered design and a new help method called Integrated Initial Guidance. Multi-layer designs structure an interface so that a simpler interface is available for users to get started and more complex features are accessed as users move through the more advanced layers. Integrated Initial Guidance provides help within the working interface, right at the start of the application. Using the metaphor of "sticky notes" overlaid on top of the functional interface locates the main widgets, demonstrates their manipulation, and explains the resulting actions using preset animation of the interface. Additional sticky notes lead to example tasks, also being executed step by step within the interface itself. Usability testing with 12 participants led to refined designs and guidelines for the design of Integrated Initial Guidance interfaces.

1   Introduction

The help system of computer applications is design to provide useful and usable information on their usage. While standardization and innovations in user interfaces have made computer applications easier to use, it is clear that getting started with unfamiliar interfaces, learning advanced features and understanding application domain concepts remains a challenge for many users. Although the help system is typically used only when users cannot proceed with their task and most of the help information is never consulted at all, there is no doubt about the necessity of providing help. Different applications will benefit from different styles of help: for example a complex CAD system to be used daily by an engineer will require elaborate tutorials and detailed online reference manuals. On the other hand a web application used once or twice by novice users may benefit from a video demonstration or a one-page set of "get started" directions. Such public access applications are likely to be abandoned by users who cannot accomplish their goal within the first few seconds or minutes. Designing an easy to learn interface is designers' first goal. The next challenge is to choose an effective method to help users learn that interface.

In the context of our NSF funded project aiming to improve citizen access to statistical information, we are exploring which type of help might be best suited to improve the learnability of tools to find, browse or manipulate statistical information. Public access information systems imply that most users will be first time users of the interface, and that they will have limited time and interest in learning the system. Users will want an answer to their question but not necessarily learn all the features of the tool. We chose to focus our work on the investigation of multi-layered application design and a new help method called Integrated Initial Guidance (IIG or "sticky note" approach). A prototype of Integrated Initial Guidance has been implemented in Ymap application, which is a relatively simple interactive tool for visualizing statistical Census data. The help interface was revised iteratively while we collected feedback from users. Our current results highlight the benefits of the Integrated Initial Guidance approach but also the challenges in designing and implementing this new approach.

2   Related Work

The multi-layered approach, initially called level-structured approach (Shneiderman, 1998), advocates the use of two or more interfaces each containing a pre-determined set of features of growing complexity.

McGrenere, Baecker and Booth (2002) allowed users to specify which features would be available in each layer, and conducted an experiment showing that users were able to learn to use a commercial word processor when first using the simplified version of the software.

There has been limited recent research activity on online help. Early work in the 80's showed that avoiding jargon, keeping sentences short, emphasizing tasks was helpful, but the philosophy of integrated help is primarily guided by many of the principles of the minimal manual (Carroll, 1997), which promotes shorter manuals providing procedural information. Most books on help cover the design of extensive online help manuals. Palmiter (Palmiter, Elkerton & Baggett, 1991) showed that users were faster and more accurate when performing tasks after being shown animated demonstrations than textual explanations. Harrison (Harrison 1995) showed that using help using visuals was more effective than text only explanations. (Wiedenbeck & Zila 1997) showed the importance of providing exercises. There are many commercial products that provide animated demo help, especially with Macromedia Flash (www.macromedia.com). Recently, text documentations are replaced or reinforced with animated demo help, but there has been very few research attempts on how to integrate those new kinds of help and the actual applications in seamless way.

Previous research (Knabe 1995.) showed the problems of non-integrated online documentation and help.
- Being switched out of the application they need to use to complete their task.
- Confusion with the help illustrations easily confused with the functioning interfaces.
- Overlapping of windows, with likely hiding of important parts of the interface and or help.
- Because of the above-mentioned problems, users have to remember series of steps before they can try to perform them.

These problems guided us toward the use of integrated help.

3   The Case of Ymap

We chose Ymap as a case study. (Note that Ymap was originally called Dynamap.) Ymap is a relatively simple interactive visualization tool developed by the Human-Computer Interaction Laboratory and to be soon released by the Census Bureau with Census information on CDs. Users can click on a map to display in a table facts about the area selected. They can select multiple areas and zoom on the map. They can use dynamic queries to filter the map according to a list of criteria. Finally they can use a scatter plot, tightly coupled to the map and table, to see relationships between two criteria (see http://www.cs.umd.edu/hcil/census). Our numerous demonstrations of Ymap made us confident that users could understand and use Ymap after a minute of two of demonstration, but usability tests at Census with novice users receiving no training revealed that users had difficulties getting started with Ymap. They have problems with zooming and multiple selections, but also with the sliders, and are often puzzled by scatter plots. We experimented with Integrated Initial Help with a single layer version of Ymap, and with a multiple-layer version of Ymap.

3.1   Creating a multi-layered design of Ymap

Ymap is well suited for a 3 layer design (Figure 1) using
- level 1: map and table only
- level 2: map and table, plus dynamic query filters
- level 3: map and table and dynamic query, plus scatter plot

This could be accomplished easily by hiding the appropriate subwindows of the interface. When users go up one level, the appropriate parts of the interface are added with animated transitions, which help users see what is changing.
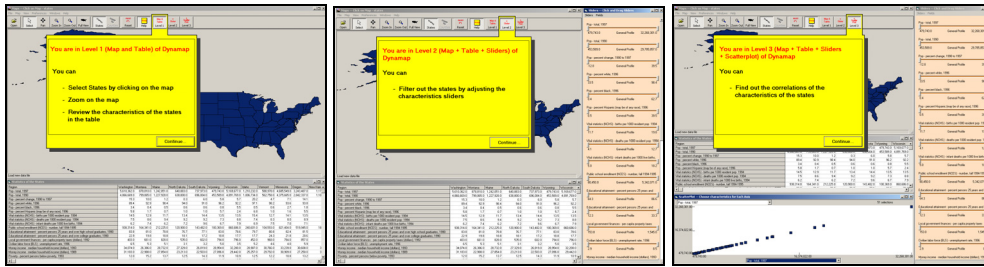
Figure 1: Three different levels of the multi-layered approach, (a) the first level shows only the map and the data table (b) second level shows slider bar along with map and table (c) the third level adds scatter plot. A yellow introduction panel lists the features of the level and indicates the location of the level buttons.

3.2   Integrated Initial Guidance

"Sticky note" help was first implemented in the PhotoFinder kiosk (Shneiderman et al. 2002.) Integrated Initial Guidance (IIG) implements sticky notes inside the interface itself, thereby allowing users to use the interface or run automated demonstrations while reading the sticky notes overlaid on the interface.

Integrated Initial Help:
1) Highlights the *main* functions of the interface;
2) Shows the location of the *main* interface widgets whose use can be explained or demonstrated with a "show me" button which
   o Introduces procedures as series of steps
   o Provides explanation of the effect of actions
   o Hints at alternative (generally more advance procedures);
3) Provides lists of simple to complex example tasks (e.g. find the state with the higher average income), which lead to demonstrations of advanced interface widget functions.

Our first IIG implementation (see examples in Figure 2) was improved iteratively while we collected feedback from 12 users. An informal think-aloud procedure was followed. Users were given neither demonstration nor specific instructions but only some context for the situation: "Imagine that you are considering moving elsewhere in the US and are looking for information about states to decide where you might want to live. You went on the web and found this site at the census bureau". After they explored the site for about 20 minutes they were interviewed in more detail, then shown the other interface and asked to comment on what they liked and didn't like about the two interfaces.
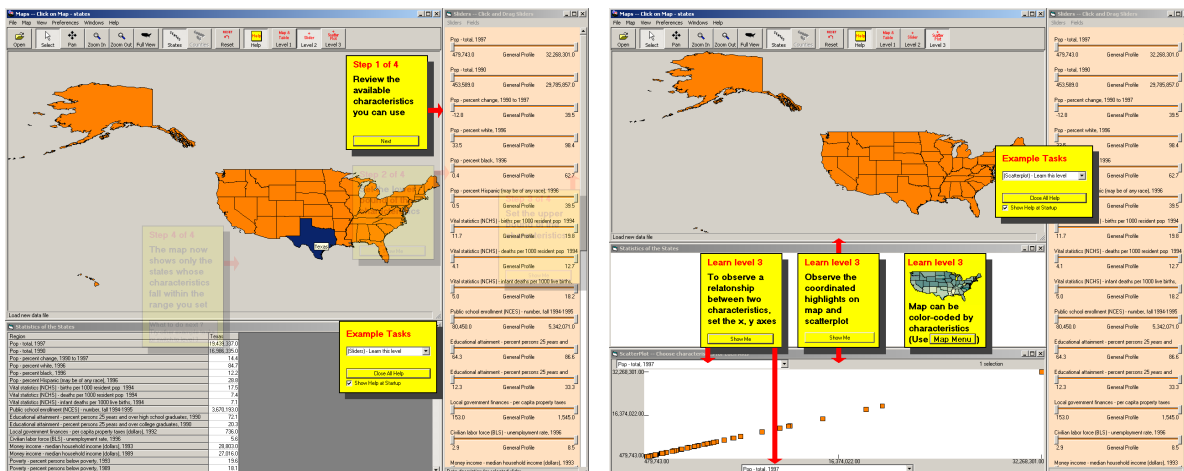

Figure 2: Two examples of IIG sticky notes. The left example (a) corresponds to a serial grouping of sticky notes. It shows the series of steps to accomplish a task. The right example (b) shows the sticky notes representing separate or "parallel" features in a level of the interface.

The most striking observation was that users used the IIG features in very different ways, and that together the first six users used every feature we had designed in the IIG. One user focused his use of the help mostly on trying out the example tasks, while several others never used them. Some users used the "show me" feature; others did not and executed the steps themselves; others started by doing it themselves but then used the "show me" function when they failed to guess the procedure (e.g. to select multiple states). All users found something of interest in the data about the states and became engaged in the application. The help sticky notes were never explored systematically. Most users indicated that they preferred the multi-layered interface to the interface presenting all features at once.

3.3   Design Guidelines

While designing the IIG we were confronted with the conflicting goals of trying to keep the help simple while addressing as many interface aspects as possible.

Completeness

The Integrated Initial Guidance *is likely to always be incomplete* (unlike the more traditional reference manuals or tutorials which are supposed to cover all aspects of the system, exceptions, etc). The designer's role is to choose which aspects of the interface need to be addressed in the IIG and how to structure the help information by controlling the grouping, location and ordering of the notes. To refine the original goal of focusing on the major functions of the interface we suggest the following guidelines to select features to be explained with the IIG:

- Highlight the unconventional features (e.g. the buttons leading to the advanced interface layer) and ignore standard features (e.g. open and save files, or reset)
- Demonstrate the use of widgets that lack affordances (e.g. show that you can click on the map or on points of the scatter plot)
- Prioritize the importance of features and eliminate help for less important ones in order to avoid cluttering the screen with IIG notes.
- If you still have room, remind users of the features people are known to forget or not notice (e.g. scroll the table)

Grouping

Once designers have selected which aspect of the interface to provide help for with IIG, it is important to structure the help information in meaningful chunks, when each chunk corresponds to a single sticky note. Small chunks will minimize the need to remember instructions as they are executed with the "show me" or as users execute them themselves, but small chunks will lead to more sticky notes, cluttering the screen and adding confusion.

In our exploration, we found that chunks could be grouped either in series or in parallel. A series of sticky notes corresponds to a natural series of steps associated with a procedure (e.g. select a dynamic query slider, adjust the minimum value, the maximum value, and explain the effect on the map.) It was important to mark the sticky notes with progress indicators (e.g. 1 of 4), and we found it useful at any given point in the series to show the past and future steps as translucent notes, allowing users to go back and repeat certain steps, or skip others. All the task examples corresponded to series as they demonstrated a particular way of performing a task. (Figure 2(a))

Parallel groupings of chunks correspond to choices of interface features. For example, parallel notes can show that users could either zoom, or select on the map, or change level. All notes are shown as equivalent, and remain visible at all time even during the "show me" demonstrations. (Figure 2(b))

We often found that the information we needed to provide corresponded to parallel series. We attempted to create mixed serial and parallel structures (e.g. a parallel of 2 alternatives: "zoom on the map" or "select states", and each represents a series of step. For example, "zoom on the map" is composed of a series of steps "click on zoom button", "click on the map", and "click on zoomout button"), but our user testing indicated that users felt disoriented by the complexity of this hybrid flow.

After several attempts we concluded that there was a strong benefit in choosing one or the other (parallel or series) but avoiding combining the two. This implies that designers have to force alternatives into a series (i.e. explaining/demonstrating first the several zooming steps, then the steps to selecting the map), or force series into single demonstration by aggregating steps into a single "show me" demonstration.

Location

Another challenge was to place the sticky notes at their best possible location. We used the following rules:

- Place sticky notes close to the related widget
- Avoid hiding important components of the interface at the start, but also later when "show me" animations take place.
- Layout sticky notes so that their layout implies the order in which they should be read. Of course we assumed that in general users would read from top to bottom and left to right, but this is not always the case. For example users tended to read the sticky note that was the closest to where their cursor was (i.e. where they had just clicked before).

We found this part of the design particularly challenging, as those three rules were often not compatible. Because the placement was so delicate, we chose to fix the location of the sticky notes. Because users could use the interface in a normal open ended way sticky notes sometime got in the way of users' interaction with the interface despite our attempts to avoid it. Users tried to move the notes but could not, leading to some frustration. On the other hand they all understood the rational for keeping the notes at a fixed location when it was explained to them.
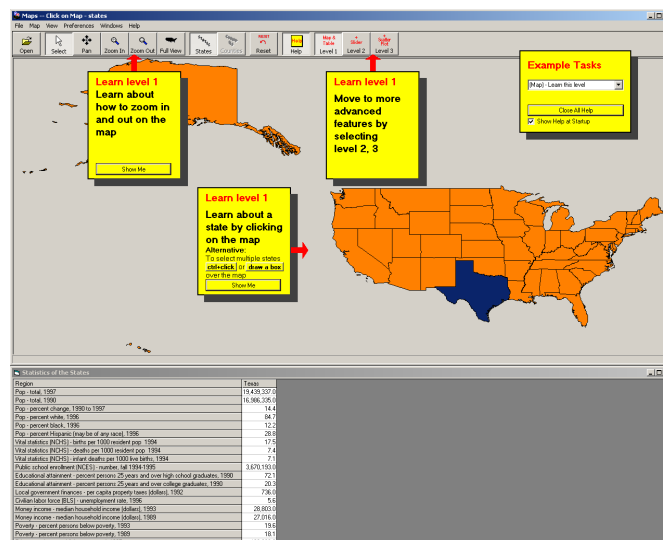

Figure 3: Sticky notes are placed close to the related widgets and the red arrow attached to them represents which widget is related to the sticky note instruction. The example task sticky note is placed at the upper right corner of the interface to minimize the occlusion.

Interactive Demonstrations

By definition, the "show me" demonstrations were executed in the working interface and allowed users to either perform the step themselves or see each step being executed for them. This has the potential to create situations where the interface and the automated demonstrations are not synchronized (e.g. the step might instruct users to select Texas but they select another state). When a user fails to perform an instruction as instructed, we chose to have the help send the user back to that same instruction or to do the step automatically depending on the instruction. When an instruction has already been completed by the user, the help system also skips to the next sticky note instruction automatically.

Examples

Most questions asked by the first time users of Ymap seemed to be either goal questions or procedural questions. Therefore we added an "example tasks" sticky so that users could see what kind of tasks was possible and consequently learn by following real example tasks. Ordering the examples from simple to complex was helpful, and for consistency we added a "Learn the interface" example task used as the default menu item when the initial introductory sticky notes are visible.

4   Conclusions

As the length of the guideline section suggests, we found that it was not easy to design and effective set of sticky notes explanations for Ymap. Nevertheless it was clear that this process was made much easier by the use of a multi-layered approach. The multi-layered approach naturally simplified the task of structuring the help information. Each level was explained with a smaller set of sticky notes and it was easier to organize them in series or parallel groups. We attempted to create an equivalent set of explanations for the single-layer version of the interface but were quickly discouraged to do so by users who found the already complex all-at-once interface even more overwhelming with all the sticky notes overlapping it. The multi-layered approach seems promising. It facilitates the discovery of the simpler features, bringing more satisfaction to more users who may never seek the more advanced features. Our current results highlight the benefits of the Integrated Initial Guidance approach but also the challenges in designing and implementing this new approach.

References

Carroll, J.M. (1997). *Minimalism beyond the Nurnberg Funnel*, MIT Press, Cambridge.

Harrison, S.M. (1995). A comparison of still, animated, or nonillustrated on-line help with written or spoken instructions in a graphical user interface, *In proceedings on Human factors in computing systems*, CHI'95, 82-89, ACM,, New York.

Knabe, K. (1995). Apple guide: a case study in user-aided design of online help, *Conference companion on Human factors in computing systems*, May 1995.

McGrenere, J., Baecker, R. & Booth, K. (2002). An evaluation of a multiple interface design solution for bloated software, *In Proceedings of CHI'02 Human Factors in Computing Systems*, 163-170, ACM, New York.

Palmiter, S., Elkerton, J. & Baggett, P. (1991). Animated demonstrations vs. written instructions for learning procedural tasks: A preliminary investigation, *International Journal of Man-Machine Studies*, 34, 687-701.

Shneiderman, B. (1998). *Designing the User Interface: Strategies for Effective Human-Computer Interaction (third edition)*, Addison-Wesley Publishers, Reading, MA.

Shneiderman, B. (2000). Universal Usability: Pushing human-computer interaction research to empower every citizen, *Communications of the ACM*, 43(5) (May 2000), 84-91.

Shneiderman, B., Kang, H., Kules,B., Plaisant, C., Rose, A. & Rucheir, R. (2002). Design: A photo history of SIGCHI, evolution of design from personal to public, *Interactions,* 9(3), 17-23.

Wiedenbeck, S. & Zila, P. L. (1997). Hands-on practice in learning to use software: a comparison of exercise, exploration, and combined formats, *ACM Transactions on Computer-Human Interaction (TOCHI)*, 4(2), 169-196.