Menu Structuring for Mobile Devices

Katrin Sauerwein[1], Nathalie Prevost[2], Alexander De Luca[1]
[1]University of Munich [2] Simon Fraser University Surrey
sauerwei@cip.ifi.lmu.de, nathalie_prevost@sfu.ca, alexander.de.luca@ifi.lmu.de

Abstract: This project sets a discussion about possible improvements for mobile menu structuring. Navigation on mobile phones is supposed to get quicker and easier. To reach for better user overview and orientation, the relationship between the single menu items is visualized. To prevent wasting the expensive screen space, icons are used to represent the items.

Three different approaches are compared. *Image Embedding* arranges the content in its context by drawing according background images. The *Manhattan Lens Stairs* approach is based on human spatial memory abilities. It uses the Manhattan Lens to differ between important and unimportant items. Finally the *Dynamic Neighbors* approach represents the menu items' relationship by using color gradients. To enable a quicker navigation process, important items to display are calculated in every navigation step. Latter is implemented and tested in a user study afterwards.

1 Introduction

The increasing variety of mobile applications results in growing mobile menus. Due to the general limits, like the little screen size, standard mobile menus appear unclear. Users get lost navigating their mobile devices easily. As it is natural for humans, they still will not ask about the navigation path to find the searched item. They would rather just try. Poor overview, orientation and help functionalities activate such kind of behavior even more. Therefore, users often navigate along different paths, searching for their menu item. They first click and check afterwards. So more clicks and time than necessary are needed. The progress of execution is supposed to be different. Users need to know the right path before performing the navigation action itself. At least users shall be guided to their 'destination' item. To reach out for advancements, the key aspects for human perception like provision of orientation, overview and details-on-demand [Sc96] need to be considered. Additionally the minimal necessary navigation paths to one single item shall be shortened. If done so, less clicks are needed to find a menu item and the navigation process gets quicker. To save expensive screen space, icons are used to represent the single items. The data structure is supposed to be easy to understand by illustrating the relationship between the menu items.

As the range of mobile phones is huge, for the project it had been decided to specialize on one kind. So the developed approaches are focused on devices including a navigation-cross enabling users a 4-way directed menu handling. In the next section three different approaches are developed and compared.

2 Approaches

2.1 Image Embedding

The simplest approach utilizes the easiness for humans to remember items arranged in scenery [Le04]. The scenery has the power to implicate the items. So *Image Embedding* locates as many menu items as possible on one screen. The background consists of an image, which puts the items into their context and therefore relates them to each other. For instance, nine items are placed in their context, which shall be the organization of a hospital. So the hospital building is displayed in the back and the items mirror the available subareas. It is important to include all the items' content in the scenario context to understand their meaning and coherence.

As in Figure 1 (left) the items are placed according to the way they fit into the image. The start screen is kept general. It shall provide an overview and represent the whole application. A cursor highlights the selected part of the screen. By entering such a single part, the view is zooming in and this particular part is spread across the whole screen and therefore can be displayed in more detail. So a zoom and details-on-demand functionality is provided.

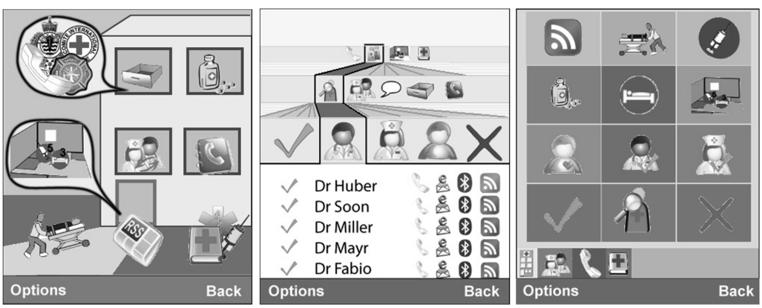

Figure 1: Image Embedding (left), Manhatten Lens Stairs (middle), Dynamic Neighbors (right)

2.2 Manhattan Lens Stairs

The *Manhattan Lens Stairs* approach is built on humans' spatial perception. Important items are displayed closer than unimportant ones [Ro98]. An item is declared as most important, if it is either selected or in the same level as the selected item. Therefore the data is available in a tree structure. Each depth of the tree equals one level. The importance of a special item decreases with the increasing amount of levels between that special item and the currently selected item. As the item size matters for perceiving spatial distances, important items are displayed comparatively bigger [KBS05].

The screen is arranged from top to bottom, as shown in Figure 1 (middle). In the first step just the first line is displayed. As soon as one item in that line is selected, the second one appears and shows the items according to the selected item of the first line and so on. Because more levels can be displayed at once, overview is provided. To generate the impression of spatial distances a variation of the Manhattan Lens is used [Ca99]. Items on the same navigation path can be related by using according colors. Therefore the path of navigation is clearly visible and provides orientation to users.

2.3 Dynamic Neighbors

The *Dynamic Neighbors* approach arranges items in a matrix-like way. Neighbors are adapted dynamically. Based on the selected item and the direction users click, the neighbors to display are calculated. Therefore the data is structured as an undirected graph. The distances between the items are known. While items representing similar content are located close to each other, items with different content are positioned further away.

Figure 1 (right) displays an example. Users shall be guided in a way they always know where to navigate. They know that the closest neighbors are displayed, but it is unclear which direction to click next to find an item. A solution is assigning each item a background color, which represents the position in the data graph and relates the items to each other. Close items have a high relationship and therefore a similar background color, but items connected through long distances get assigned different colors. With some knowledge about color gradients users can navigate without loosing their orientation.

3 Dynamic Neighbors

As the *Dynamic Neighbors* appeared promising during development, the approach got designed in more detail. The menu consists of the items and their colors and is brought alive by the dynamic neighbor calculation. This section is parted accordingly.

3.1 Menu Items

Each menu item is represented by an icon, which shall be as self-explanatory as possible. To display the relationship between the items, a color is assigned to each of them. As mentioned above the items need to be arranged in an undirected graph, within the items equal the vertexes. According to the content a few items representing the main navigation categories are chosen and assigned a color manually. Due to the graph structure the distance from every other item to each main item is known. With this principle a color value cv can be assigned to an item as follows:

$$cv = \frac{1}{\text{distance to item 1}} \cdot \text{color of item 1} + \ldots + \frac{1}{\text{distance to item n}} \cdot \text{color of item n}$$

Due to color intensity the calculation can be varied. The resulting color represents the position in the graph and therefore in the menu. As shown in Figure 2 the blue icon on the left represents a browser. The red one on the right mirrors a social community. These items visualize the main categories. The icon at the bottom is meant to represent a person's work. This person is member of the community, but it is also possible for everyone to browse his/her project work. Using the mentioned formula this pinkish color is assigned.

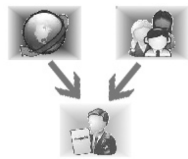

Figure 2: Color Arrangement

Acted on the assumption that users can estimate of which basic colors a color is mixed. So they are aware of their current position in the graph by perceiving the colors. Users looking at the bottom item in Figure 2 know they are close to the browsing and the community item. Instead, if they would look at a green item, they know that the current position is far away of those and they might need a longer navigation path to get there.

3.2 Menu Interaction

Twelve items are visible at the same time. Therefore, the display is split into a 3x4 matrix (t: top, m: middle, l: left, r: right, b: bottom):

tl	*tm*	*tr*
m1l	*m1m*	*m1r*
m2l	*m2m*	*m2r*
bl	*bm*	*br*

Each entity contains one menu item. To understand the calculation process, the terms *center of view* and *region of interest* are defined. The *center of view* consists of those matrix fields, which give information about the items currently displayed. These are the most important neighbors of the items in the center. The exact entities of the center are identified in each navigation step. The two middle lines are set while doing vertical-navigation. Horizontal navigation places the center to the middle column. The selected item is always positioned in the center, either on position *m1m* or *m2m*. Instead of moving a cursor across the screen, the items are visually moved to the *center of view* according to their current importance.

To decide which items to place for the next click, the *region of interest* is set. Items positioned in that region define the other neighbors to display in the following navigation step. With a 4-way navigation cross, four different *regions of interest* are possible. As

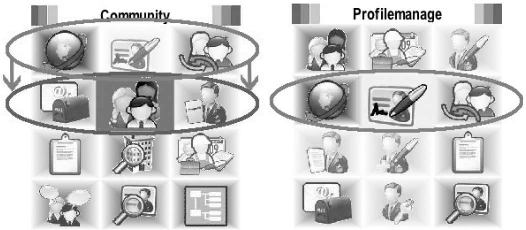

Figure 3: Menu Interaction: Before (left) and after (right) clicking up

example Figure 3 illustrates the necessary chain of reactions for a click 'up'. Position *m1m* contains the selected icon in the middle of the red marked center. If users click 'up' on the navigation cross, the *region of interest* is put on positions *tl*, *tm* and *tr*, which is outlined in blue (left). The containing items are visually moved to the new blue marked *center of view* on the right side of the Figure. All other items are faded out and dynamically replaced by the neighbors of the new items in the center. Filling up starts from the selected item. Due to the colors, which represent the position according to the whole menu and the dynamical arrangement of the items, users are guided to their demanded item.

To finally provide an understandable menu, an always callable top level is added [LB05]. It contains all main items. Each main item displays the number of underlying items. So users can always jump back to that overview level and start at one of the main categories.

4 Evaluation - User Study

The *Dynamic Neighbors* and an analogical static menu were implemented for the Nokia N80 using Flash Lite 2.1, before a user study was done. The participants were asked to look up the same items in each menu before filling a questionnaire. First times and number of clicks needed to find an item were measured. In a second step a questionnaire helped to answer further aspects. Seven people within an age group of 18 and 35 years, who are using their mobile devices regularly, took part in the study.

71. 4% of the participants preferred the dynamic menu, but it took some time of familiarization to understand the dynamic data arrangement and the color model. The evaluation is as follows: The average time needed to find an item was 30 seconds and could not be reduced compared to the static menu (25 seconds). The longer the minimal necessary path to an item is, the less the difference in time appeared. Reason is the fading effect used for the *Dynamic Neighbors*. So each single step done in the dynamic menu takes 5 seconds longer than in the static one. In contrary the number of needed clicks in average was one click less in dynamic menu. As each item can be reached from different directions via different ways, the minimal necessary navigation path is shortened. Therefore reducing the time for one single navigation step would advance the *Dynamic Neighbors* to be the faster navigation possibility.

The success of the color model was neither confirmed nor declined. Users were definitely supported by the colors, because they needed less clicks. They used cross-links, which are offered of the colored neighborhood representation. But participants were not aware of using the model for orientation, as the according question just scored 2.9 out of 5 points. A possible reason is the wide range of used colors. So even neighbored items could have relatively different colors. Additional the fully colored icons take a lot of accentuation off the background color.

The additional top level provides the desired overview. Mainly at the beginning of navigating, participants used it to get hold of the different categories' content. In addition the overview level provided a lot of orientation. So it was called in case a person got lost in the menu.

5 Conclusion and Future Work

The introduced approaches of the *Image Embedding*, *Manhattan Stairs Lens* and *Dynamic Neighbors* offer a good basis for discussing mobile menu structuring. The implemented *Dynamic Neighbors* menu proved to achieve the basic goals of providing overview, orientation and details-on-demand functionality. Still future work has to be done to reach out for a fully operational version. By reducing the time needed for each single navigation step, it verifies itself to be the faster menu. An additional advancement of the color model can enable users to find even shorter navigation paths and therefore reduce navigation times as well. Therefore the starting point for future work is set.

References

[Sc96] Shneiderman, B: The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. In Proc. of the IEEE Symposium on Visual Languages '96, Los Alamitos, 1996.

[Le04] Lewis, J.; Rosenstolz, R.; Fong, N.; Neumann, U.: VisualIDs: Automatic Distinctive Icons for Desktop Interfaces. In Proc. of the 2004 SIGGRAPH Conference, ACM Transactions on Graphics (TOG), 2004.

[Ro98] Robertson, G.; Czerwinski, M.; Larson, K.; Robbins, D., C.; Thiel, D.; Dantzich van, M.: Data Mountain: Using Spatial Memory for Document Management. In: Proc. UIST'98, 1998.

[KBS05] Karlson, K.; Benderson, B.; SanGiovanni, J.: AppLens and LaunchTile: Two Designs for One-Handed Thumb Use on Small Devices. In Proc. CHI'05, 2005.

[Ca99] Carpendale, M., S., T.: A Frameword for Elastic Presentation Space. Ph.D. thesis, Simon Fraser University at Vancouver, 1999.

[LB05] Lam, H.; Baudisch, P.: Summary Thumbnails: Readable Overviews for Small Screen Web Browsers. In CHI '05: Proc. of the SIGCHI conference on Human factors in computing systems, New York, 2005.