

Roller Interface for Mobile Device Applications

Lijian Wang, A.S.M. Sajeev

School of Mathematics Statistics and Computer Science
The University of New England
Armidale 2351, NSW, Australia

{lwang2, sajeev}@turing.une.edu.au

Abstract

Mobile devices generally have small screens. To display information that is well suited for larger screens (e.g. desktop computers), the information has to be segmented into many small presentation units that can fit into the small screen of mobile devices. This makes it difficult to effectively organize information and help users navigate to and from the information they want. This paper presents Roller, a new interface technique to present application interface on mobile devices. Roller helps alleviate the screen real estate limitations in user-interface design and provides rich contextual information to ease users' navigation tasks. Our primary trial study shows the effectiveness of using Roller to organize and present information on small screens.

Keywords: Roller, application data, application interface, presentation elements, presentation unit.

1 Introduction

For small devices such as mobile phones and palm computers, limited screen size makes it difficult to efficiently present information because the information that can be displayed on a desktop has to be split into many smaller blocks to fit into a small device. The problem even gets worse when presenting information of large volumes, for example, displaying a big city map or navigating a complex website. Besides, a small screen makes it hard to display the context of the information that is already on the screen. Users can easily feel being lost when navigating an information space that consists of a dozen or even hundreds of sub information blocks especially if they use devices in mobile situations where there could be "on the road" distractions to the user that may not be there on an office desk.

To address these problems, we propose Roller, a new object to organize application interface and present application data. Roller is a 3D oriented trigonal prism with ability to roll in and roll out. It can open to reveal inner contents and close to return to original outer state. Roller can present both linear and hierarchical data in a compact and flexible way, as well as provide rich context information for the information in focus.

Copyright © 2007, Australian Computer Society, Inc. This paper appeared at the Eighth Australasian User Interface Conference (AUIC2007), Ballarat, Australia. Conferences in Research and Practice in Information Technology (CRPIT), Vol. 64. Wayne Piekarski and Beryl Plimmer, Eds. Reproduction for academic, not-for profit purposes permitted provided this text is included.

We have used Roller to present the Yahoo mobile website as well as reorganize an Contact Book application interface, and achieved better-organized interfaces than the original ones. We have also developed a demonstrating program to show the feasibility to deploy Roller in real mobile devices.

This paper is organized as follows. Section 2 gives related work on small device interfaces. Section 3 describes the Roller model and how it is used to present hierarchical and linear data. Section 4 gives an example of using Roller to organize the Yahoo mobile web site. Section 5 shows how Roller is used to aggregate an application interface. Section 6 demonstrates a practical application developed with Roller. Section 7 discusses some design issues, and conclusions are given in section 8.

2 Related Work

To render web pages on to mobile devices, information generally is split into sub sections that are mostly organized in hierarchical structures. Different strategies have been proposed to display these smaller sections on mobile devices such as the Deck/Card metaphor, Zoomable user interface and 3D interface.

Hierarchical structures are presented in different forms. For example, the M link system (Trevor, Hilbert et al. 2001) organizes information into multiple hierarchy levels and use M links to connect different levels of the hierarchy. Buyukkokten et al. (2001) use a tree view to organize web pages in a tree structure and users click the tree nodes to expand the subsections and view details. Chen organizes web pages into a two level hierarchy with a thumbnail representation at the top level for providing a global view, and indexes at the bottom level for detailed information of sub pages (Chen, Ma et al. 2003). A common problem of these systems is that when viewing sub level information, users lose the context because its top level information disappears and that can cause user confusion.

The Deck/Card structure that originated from W3C WML (WapForum 1998) is another way to organize information. A card consists of a few pieces of information and is the smallest interface unit to display on mobile devices. Deck is a collection of cards that logically belong to a group. The "card" structure of WML, thought by some to be clumsy, will be replaced by XHTML Basic, which is a W3C recommendation for Web clients such as mobile phones, PDAs, pagers, and settop boxes (W3C 2006).

Zoomable user-interfaces (ZUIs) arrange information in an infinitely pannable space and at many nested levels of detail. Such interfaces, for example: Pad ++ (Bederson, J. Hollan et al. 1996), are used for information visualization and browsing. In small devices, as users pan and zoom, displayed content can easily disappear into the off-screen space. Some techniques have been proposed to deal with this issue. For example, rings that interlace with the border of the current display window are used to indicate that the off-screen objects are at the centre of the rings (Baudisch and Rosenholtz 2003). Zellwege et al. (2003), on the other hand, have suggested green thick lines along the borders of windows to indicate height/width and orthographic direction of unseen objects. However, with these techniques, the context information users can see is only the indicator (ring, or thick lines) and not the actual contextual objects.

In a 2D interface, most of the interface interactors such as menus and buttons are located in a rectangular display area. As 3D technology becomes more advanced, 3D objects, typically, 3D cubes, have been used in user interfaces. Linux allows users to open up to four desktops on a 3D cube that can rotate for program access (Eric Lai 2006). Mac OS (Apple 2006) uses a cube to manage user accounts. Each user's desktop is put on a side of the cube; when a new user wants to log into the computer, the current user's desktop rotates out of view and the new user's account on another side of the cube rotates into the view. 3D cube also has been used as navigating menus (Ciucci 1999) for getting access to different web pages where each side of the cube acts as a link to a web page. 3D cubes used in those works target desktop computers rather than mobile computing devices. They are used in a quite preliminary way because they are only able to display six different blocks of information corresponding to the total number of cube sides.

Distinct to the above works, we propose a new model termed Roller. It is a 3D object that can be used to present unlimited number of information blocks in both desktop and mobile computing devices. This widget not only can organize information in an efficient hierarchical structure, but also can display out-of-focus information (which provides the context) concurrently with in-focus information for easy user interaction.

3 The linear and hierarchical structure of Roller

Roller is a trigonal prism consisting of five sides that are identified as left, right, back, top and bottom if we face it straight (see Figure 1). We use Roller's body (left, right and back side) to represent elements that could be data units or application user interface units. The top and bottom sides of Roller are not used to present information but still drawn to give users a geographical sense of wholeness. Roller can provide *rolling functions*: rolling left or rolling right; and *revealing functions*: open up a door to expose things inside or close it to gain the outside view. The metaphor of *rolling functions* comes from wrapping a trigonal prism with a long cloth that can be considered as a display area to put presentation units; the wrapping cloth can be wrapped or unwrapped to provide

the required information. The metaphor of *revealing functions* comes from exploring a building with many secrete rooms that are interconnected, and a room can be opened to reveal what is inside and as a corridor to go to another room.

Linearity and hierarchy are the common ways to organize application data and present an application interface. In a linear structure, information elements are linked in a direct succession; the sequence of these elements can be ordered or unordered. The linear structure is widely used for web page navigation, where links to sub pages are put in a row and can be accessed in sequence or at random. In a hierarchical structure, the elements are put at different levels in the form of a tree. One element can be a parent or child element of other elements. Application menus are usually organized in hierarchy; a menu contains sub menus, which in turn contains sub menus at lower levels.

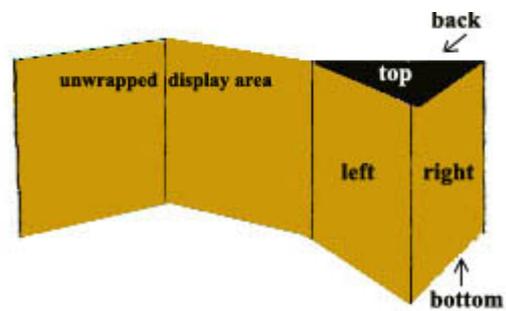


Figure 1: linear structure of Roller

Roller can represent linear elements. All the elements can be put on a presentation cloth, which is long enough to hold any number of elements, and then wrap this cloth around Roller. When we display a presentation unit, we can revolve Roller to left or right until the unit is revealed. For ordered elements, the elements are arranged in order from left to right. The first element is on left side, the second element on right side, the third element on back side, and the remaining elements wrapped inside. If we want to see Element 10, we need to roll Roller left 9 times to display it. For unordered elements, each element can be put on any Roller side randomly, whereas, some grouping constraints may still apply to group relative elements together.

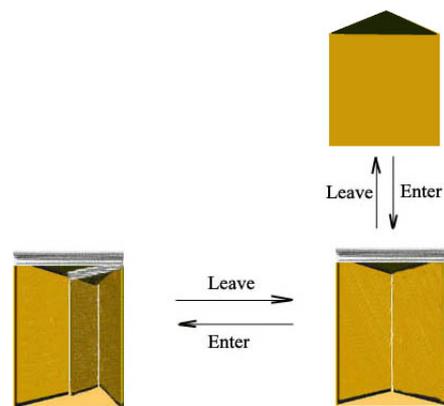


Figure 2: hierarchy structure of Roller

Roller also can be used to organize hierarchical presentation elements. By presentation elements we mean a user interface unit (e.g., File menu in Microsoft Word), or a data unit (e.g., a person's skills or working experience). Because it is actually a 3D object, the inner space of Roller also can be used to present data. If we consider Roller (a trigonal prism) as a building, the inside of this building also consist of 3 sides that can be used to present data. If we consider a side of Roller as a wall with a door on it, then this door can be opened to let users see what is inside. The data behind the door can be considered as child data of the data displayed on the door. In this way, the data can be organized in a hierarchical structure. When each door is open, it reveals two inner sides of the Roller. Each side can act as a data container to store either linear data or hierarchical data, if it is hierarchical data, then the side can be opened again to display the child data at this hierarchy. The hierarchical structure of the Roller is illustrated as in Figure 2.

4 Roller to present web data on mobile devices

To show how Roller works in practise, we use it to present Yahoo WAP website (<http://pg1.yahoo.com>) on a mobile device with a 240*270 resolution, as illustrated in Figure 3.

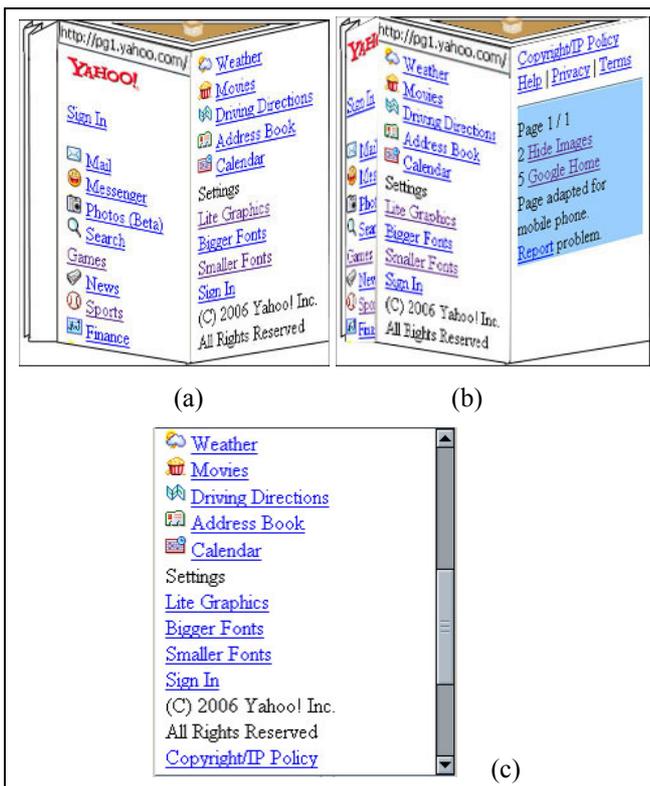


Figure3. Linear information display on a PDA screen (240*270 resolution). (a) The first two pages displayed on Roller. (b) The view after Roller rolling to left to reveal third page. (c) The original Yahoo web pages

The WAP site can be divided into three WAP pages and displayed on Roller in a left to right order as in Figure 3(a). The first two pages are arranged on the left and right sides of the Roller. When a user wants to display the third

page, he or she can roll Roller to left to get the view of the third page. If there are more than three pages, the Roller can further roll left, and that will make some pages fold up on the left edge just like folding up a curtain. If the user chooses to see a folded page, she can roll the Roller to right.

Figure 3(c) is the normal view of the Yahoo website, it shows the section from weather to copyright. The information before weather and after copyright cannot be displayed on one screen due to screen size limit. To see off-screen information, users need to scroll up and down. However, when organized on Roller, the whole web site can be put onto one screen, as shown in Figure 3(b). It gives users a view of the whole website while maintaining the visibility of the details; thus using Roller the context is not lost despite of smaller screen size of the medium.

As mentioned in Section 3, Roller can be imagined as a building. It has three sides that can be opened just like opening a door on the wall of a building. When the door is opened, the inside data will be revealed. Opening the door operation is programmed as follows: a side of Roller rotates anti-clockwise inwards to reveal two inside pane, left and right, and merge with the left pane inside. Closing the door operation is the opposite of the opening operation: the left pane inside rotates outwards to seal the Roller side being opened. Figure 4 (a) shows the result after the left side in Figure 3 (a) is opened.

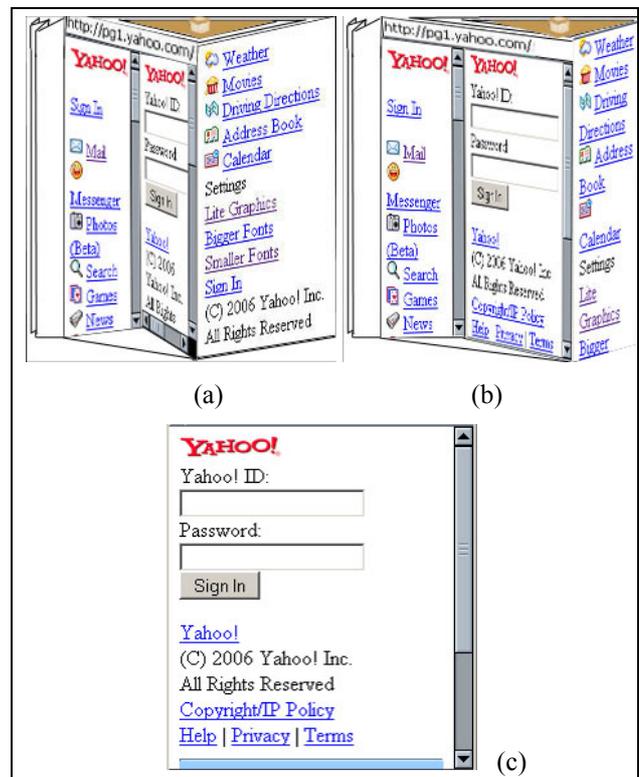


Figure 4. Display hierarchical information on a PDA screen. (a) The view after the left side of Roller is opened. (b) The Roller resizes to give more space to display mail account. (c) The original yahoo mail account

After the left side is opened, the left pane inside contains the same information as the side itself; the right pane

inside contains no information. We use left pane to display information that is higher in hierarchy, and right pane to display information that is lower in the hierarchy. As shown in Figure 4(a), after the user clicks the mail link, the right pane displays the child information (Mail account login page). The parent level information (Mail link) is displayed on the left pane of the inner Roller. If the information is greater than what the display area can display, a scroll bar is automatically formed to assist in displaying information that cannot be seen.

In Figure 4(a), the mail login screen is quite small. To help users to see the login information better, the Roller left side and right side can be resized. Figure 4 (b) shows the resized view to give mail login page more space for easy interaction, where the right side of the login page in Roller has shrunk. Even with the shrinking, the right side is still visible and users can interact with it directly. Figure 4(c) displays the original mail account login screen. It contains no context information such as the top level links of which the mail link is just one of the options. Comparing (b) and (c) in Figure 4, it is obvious that (b) presents users with more information and makes them clearly aware of their navigating locations in the website. In addition, it facilitates multiple screen interactions, which is handy if users want to view two different information spaces at the same time. For instance, a user may want to tell his friend the local weather conditions while writing an email. He can simply check the weather information on the right Roller side without losing his email context.

If we want to display the detailed information of **Copyright/IP Policy** in Figure 4(b), we need to open another inner Roller. For a desktop computer, we can do that easily because of its larger screen. However, on small devices, that will make the display area of **Copyright/IP Policy** too small to be seen. To solve this problem, rather than to open a new inner Roller, we use two inner panes of the Roller's left side to display that information. If we click **Copyright/IP Policy**, the mail login screen will shift onto left inner pane where currently the **Sign In** and **Mail** links are displayed, and the details of **Copyright/IP Policy** will be displayed on the right inner pane. The rules for displaying multiple hierarchical information are: The left pane displays the current information, the right pane displays child information of the current information. The inner Roller also has the rolling function. Rolling to left will move up one level to a higher hierarchy, and rolling right will go down one level to a lower hierarchy. The inner Roller can roll further to display information at deeper levels of the hierarchy.

5 Roller to organize application interface on small devices

In order to demonstrate how Roller is used to present application interface, we take a Contact Book application from Palm computer and reorganize it with Roller. The application provides users with address entering, editing, searching and categorizing functions. The original application on a Palm Emulator looks as in Figure 5(a).

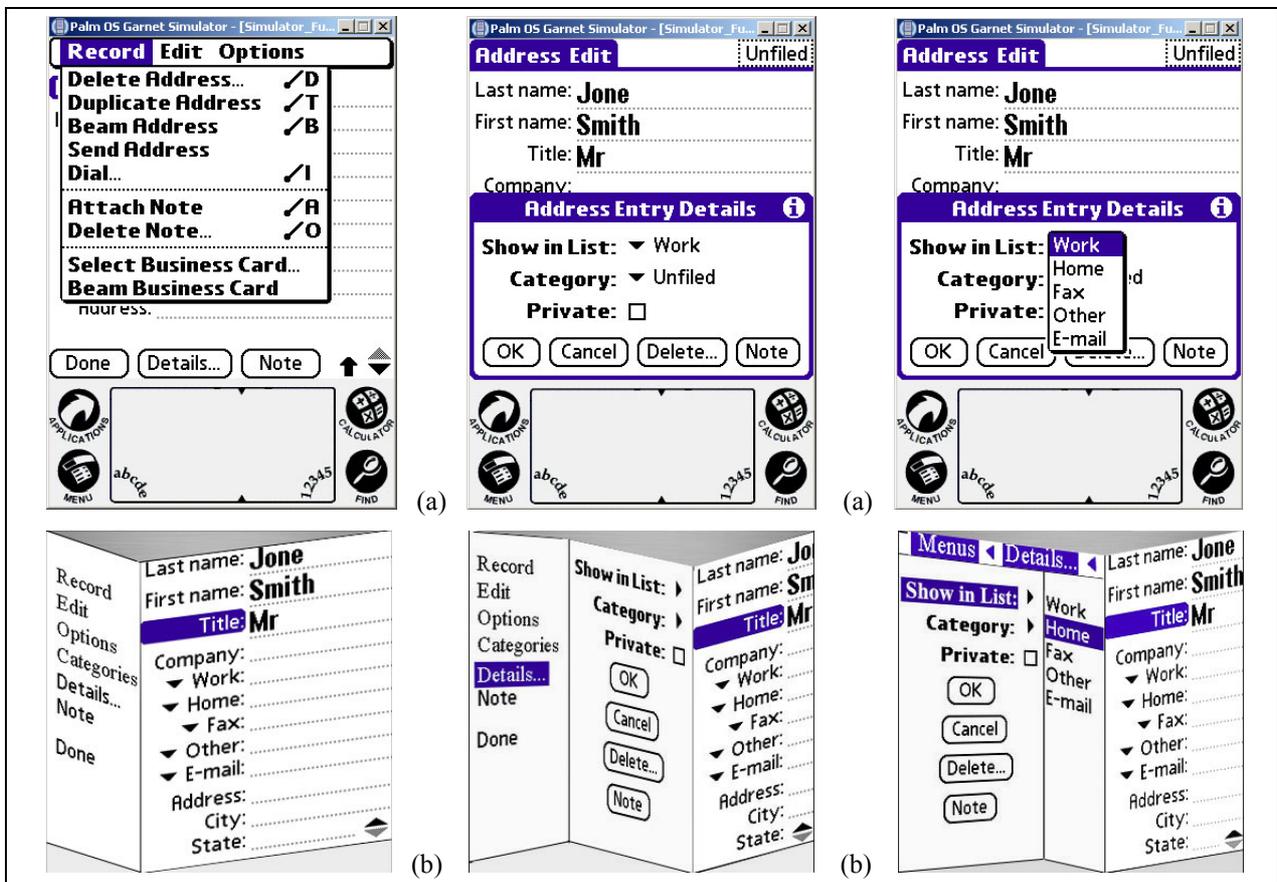


Figure 5. (a) The Contact Book application on palm computer. (b) The application displayed on Roller.

The application contains seven first level menus: **Record**, **Edit** and **Options** at upper left, a **Categories** selection at upper right, and **Done**, **Details**, and **Note** at the bottom left. At the bottom, **Details** contains submenus **Show in List**, **Category**, **Private** and four command buttons. From there, at a lower level, **Show in List** provides a few contact options such as **Work** and **Home** that can be selected to determine which contact to be seen in a record. The original menu system scatters at both the top and bottom area and looks a bit clumsy. Also, when lower level menus are evoked, the upper level menus are shadowed and become unseen. For example, when **Details** is clicked, the **Address Entry Details** pops up and shades the **Details** menu and user may get confused about where the pop-up originated from.

We rearranged the application interface on a Roller; we put all the menus on the left side and the record edit area on the right. The menus contain some submenus and thus the left side can be opened when users want to see detailed submenus. When the submenus display, the Roller is resized to give left pane more space to display the submenus, and right pane less space since it is not the activated pane at that moment.

When users further explore the submenus down the hierarchy, the parent menus roll to left and become invisible. That is undesirable, because users may lose track of their navigation path. To provide information about past menus, we add a header bar at the top of the Roller to show previous menus. The header bar is a narrow strip of display area and contains key information about what is inside just like a banner above a door does. Because the header bar has a tiny display area, it only shows the key word of context information, and may only show the first few letters if the key word is too long. Although the header bar does take some precious display area, it can give users an awareness of context information, and act as a quick way to switch back to parent level information when it is clicked.

The idea of header bar is illustrated as in Figure 5(b), when **Show in List** is clicked, it is displayed on left pane and its parent level menus disappear, and its child items are displayed on right pane. To give users some clues about the parent level menus of **Show in List**, **Details** and **Menus** are shown on the header bar. When users click **Details** on the header bar, the view will quickly switch to the middle view in Figure 5(b), or clicking **Menus** will return to left view in figure 5(b).

6 Practical application

In order to study the feasibility of implementing Roller on small devices and test its basic interaction functions we developed a city map application by using java Mobile 3D Graphics API (M3G) technology (Qusay H. Mahmoud 2004) and deployed it on a java WTK emulator phone. The map application shows the city rail map of Sydney (from <http://www.railmaps.com.au>) as in Figure 6. An overview of the map is displayed on the Roller to give users a panoramic view of the railway lines and how they are connected. Users can click the map on Roller to get the detailed view. The detailed view gives

user information such as the railway stations and their locations in a particular rail line. As illustrated in Figure 6, screen shot (a) is the original state of map application, (b) is the view after Roller roll one side further to left, and (c) is the view when a user clicks the map on the middle pane to see the full details of that map.

Although the map application is not very sophisticated, it shows the feasibility of the Roller interface technology on small devices. The application demonstrates basic Roller operations such as rolling left, rolling right and selecting a 3D object. Operations such as resizing Roller panes, opening and closing a Roller side are possible but rather difficult to implement because 3D object in Java M3G has to be manipulated for each pixel based on its coordinate values in 3D space, so these operations are not demonstrated here. We believe these operations can be done in a more efficient way as 3D technology on small devices gets more advanced.

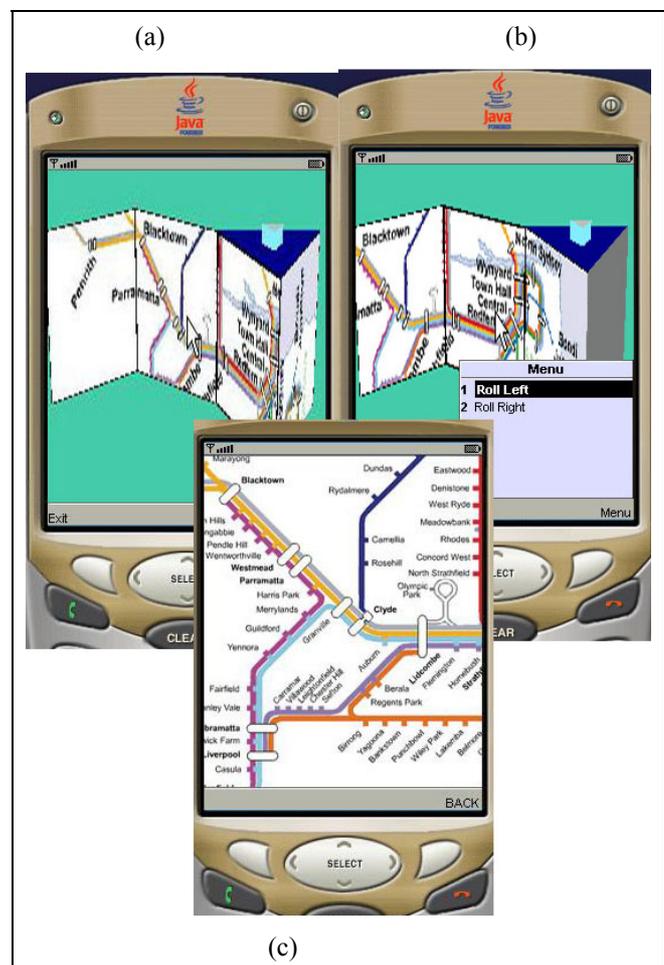


Figure 6. A map application on java emulator phone.

7 Discussion

7.1 Roller's interaction features

As an interaction widget, Roller provides ways to organize linear and hierarchical information, both application data and application user interface, and also provide basic functions to navigate through these data in responding to key events and pointer events.

Roller provides a rolling function. Roller can be positioned either vertically or horizontally. It can roll left or right horizontally, or up or down vertically. The function can be achieved through arrow keys, commands, or pointer devices that are able to rotate Roller by dragging and circling the pointer.

As for navigating hierarchical data, we imagine each side of Roller acting as a door that leads to the next level of data in the hierarchy. If a side contains hierarchical information, the Roller provides open and close operations to explore what is inside the left (and right) sides. If a side contains non-hierarchical data, then the side cannot be opened. For quickly switching a Roller's outside view to its inside view or vice versa, we can use quick buttons ◻▲, ◻ for outside view and ▲ for inside view. Alternatively, users can choose the "close door" or "open door" command. We use ► with an item to indicate it contains child items. When a pane contains more presentation units than what it can display, a scroll bar appears to allow user scroll to the content out the range of display area.

The presentation units can be organized on Roller in an ordered or unordered manner. For ordered ones, the presentation units are put from left to right based on their sequence that cannot be changed by users. For unordered ones, Roller provides a flexible mechanism to allow users to reshuffle the presentation units either by the drag and drop operation, or by typing the ID or name of a replacing presentation unit in an arbitrary way. That allows users to aggregate several interrelated presentation units together for easy interaction. An example of this behaviour is putting a client's history data adjacent to present data for easy reference.

7.2 Roller's Potentials and Limitations

Roller interface is a 3D oriented design. We found that Roller has clear advantages over traditional 2D interface. Firstly, Roller makes much more effective use of the screen real estate by displaying more information compared to a 2D interface. If we view Roller as a triangle, it uses two sides to display information rather than just one side, and the total length of two sides is greater than the third side so it can generate more display area. Secondly, Roller gives users clear space awareness during the interaction. The explorative inner space of Roller gives users a geographic sense of unity. The distinctive boundary between inner space and outer space makes them less likely to get lost during exploration. In addition, Roller provides users with good context clues and assists them to form a whole picture of navigation path. On 2D interfaces, the information in focus occasionally shade the information out of focus, whereas on Roller interface, these information are displayed jointly with varied display areas as demonstrated in Sections 4 and 5. Finally, Roller could be well suited for multiple user cooperation. For example, in playing a two participant game, Roller can display a player's state on the main side of Roller and his partner's on the auxiliary side simultaneously to assist better cooperation.

On the other hand, Roller divides a 2D screen into a few display areas. Although it provides users with resizing and full screen options that can change Roller into a single 2D screen, it increases the number of operations needed to get to an optimum display result. Furthermore, since Roller is a 3D oriented design, it consumes more memory and processing power on resource-constraint mobile devices. However, we do not consider this as a serious weakness since memory and processing power of mobile devices are always on the increase.

8 Conclusion

We have proposed Roller, a new technique to organize application interfaces on small devices. Roller is a 3D object and can present both linear and hierarchical information efficiently. The most important feature of Roller is that it corresponds to people's sense of physical space; thus it can generate a more natural and intuitive interface. Roller can present more information on the same display area compared to traditional 2D interfaces. Roller can display rich context information together with information of focus to give users an awareness of where they are in the information space.

The trial studies of using Roller to present web pages and software application interface show that Roller is an effective way to present information on small devices. The practical application developed in this study verifies the feasibility of deploying Roller application interface on small devices. As mentioned above, our implementation is on a simulator. Current interface development kits for mobile devices do not make it easy to develop three dimensional interfaces. Therefore, for usability testing, we are planning to develop a desktop computer application with full-featured Roller interface and deploy it onto a small window that has the same size as mobile devices.

9 References

- Apple (2006). Apple - Mac OS X - Fast User Switching, <http://www.apple.com/macosx/features/fastuserswitching/>.
- Baudisch, P. and R. Rosenholtz (2003). Halo: a technique for visualizing off-screen objects. Proceedings of the SIGCHI conference on Human factors in computing systems. Ft. Lauderdale, Florida, USA, ACM Press: 481-488.
- Bederson, B., J. Hollan, et al. (1996). "Pad++: A Zoomable Graphical Sketchpad for Exploring Alternate Interface Physics." Journal of Visual Languages and Computing 7: pp. 3 - 31.
- Buyukkokten, O., H. Garcia-Molina, et al. (2001). Accordion summarization for end-game browsing on PDAs and cellular phones. Proceedings of the SIGCHI conference on Human factors in computing systems. Seattle, Washington, United States, ACM Press: 213-220.

- Chen, Y., W.-Y. Ma, et al. (2003). Detecting web page structure for adaptive viewing on small form factor devices. Proceedings of the 12th international conference on World Wide Web. Budapest, Hungary, ACM Press: 225-233.
- Ciucci, F. (1999). Scorrimento automatico, <http://www.anfyteam.com/anj/cubemenu/cubemenu.html>.
- Eric Lai, F. (2006). Linux woos mainstream with graphic tweaks, <http://reseller.co.nz/news.nsf/0/F46D12D1C62E3469CC25710E007B2C84?OpenDocument>.
- Qusay H. Mahmoud (2004). Getting Started With the Mobile 3D Graphics API for J2ME (JSR 184), <http://developers.sun.com/techtopics/mobility/apis/articles/3dgraphics/>.
- Trevor, J., D. M. Hilbert, et al. (2001). From desktop to phonetop: a UI for web interaction on very small devices, Proceedings of the 14th annual ACM symposium on User interface software and technology Orlando, Florida ACM Press: 121-130
- W3C (2006). XHTML™ Basic 1.1, <http://www.w3.org/TR/xhtml-basic/>.
- WapForum (1998). www.wapforum.org/what/technical/wml-30-apr-98.pdf. 2005.
- Zellweger, P. T., J. D. Mackinlay, et al. (2003). City lights: contextual views in minimal space. CHI '03 extended abstracts on Human factors in computing systems. Ft. Lauderdale, Florida, USA, ACM Press: 838-839.