Application and device effects on the usability of mobile phones and PDAs

Paul Vickers
Jiraporn Buranatrived
Paul Clement

Northumbria University
School of Informatics

Submitted for publication in *IEEE Pervasive Computing*

Running Title: Application and device effects

Topic areas: HCI, usability, PDA, mobile phone, migration

Areas addressed:

- user impact of mobile or ubiquitous computing applications
- quantitative and/or qualitative evaluation techniques
- case studies of deployed systems (commercial or experimental), emphasizing interaction design and usability evaluation

Contact

Dr Paul Vickers
Northumbria University
School of Informatics
Pandon Building, Camden Street
Newcastle-upon-Tyne NE2 1XE
paul.vickers@northumbria.ac.uk

Tel +44 (0)191 243-7614
Fax +44 (0)870 133-9127

Version #1
13 January, 2003

Application and device effects on the usability of mobile phones and PDAs

Version #1

13 January, 2003

1. Introduction

Mobile computing is one of the fastest growing areas within the industry (as evidenced by the growing consumption rate of mobile devices that are on the market today). However, despite their popularity these devices are not without usability problems. Mobile devices are starting to replace older forms of communication and computation and developers are faced with new issues that need to be addressed throughout the development process [1]. Compared with desktop computers they have a number of substantial limitations (mainly associated with their memory, processing power, and their interfaces which are typically less sophisticated and relatively small). Increasing the physical size of a palm-top/mobile phone device is not an acceptable solution given that their fundamental purpose is to be worn or carried by their users [2]. As mobile device market penetration increases, service providers and software developers will need to find ways to migrate existing function to these new platforms. There are many challenges here and it is not as simple as 'shoe-horning' a desktop-based application into the smaller interface environment of a mobile devices. The emergence of wireless devices and mobile networks has opened up new business opportunities as e-commerce now extends into the mobile realm to become m-commerce (mobile commerce).

To exploit the technical opportunities that mobile computing offers (such as instant connectivity, localization and the capability to receive information and conduct transactions anywhere, at any time, in a real-time environment) companies must develop effective and efficient applications with friendly and usable interfaces. Designing for mobility, a widespread population, limited input and output capabilities, and for users' increased multitasking with more interruptions is a challenge that is coming to the fore of developers' and researchers' attentions [3]. Success will be affected by finding the right mix of applications that fit within constraints of limited screen size, memory and processing power. Good interface design requires more than just squeezing information into a little screen.

There is a dichotomy between the desire to write an application and deploy it without modification across multiple platforms and the benefits of customising an application to exploit the particular interface characteristics of individual devices. Furthermore, there are at least two broad categories of software that will be implemented on mobile devices: new systems designed especially for mobile environments and versions of existing desktop/server applications that have had some or all of their functionality migrated to mobile devices. In an effort to explore these issues we describe two pilot studies. In the first we investigated whether unmodified applications could be deployed on both a mobile phone and PDA without any significant difference in usability. In the second

investigation we explored some of the issues involved in existing migrating server-based functionality to a mobile platform. While the first study concentrated on using common interface components for two different devices (a mobile phone and a Palm OS machine), the second focused on exploiting the specific abilities of another mobile device (a Compaq iPaq) .

Our findings indicate that usability can be maintained through multi-platform deployment, but that there are also usability advantages if the specific interaction paradigms of different mobile platforms are taken into account.

2. Multi-platform deployment

Chittaro and Dal Cin [4] discovered significant differences in the way navigation and item selection techniques affect interaction on a single mobile phone platform. Because there are several types of mobile device capable of supporting m-commerce, each with its own user interface, there will be different usability issues for an application running on different devices. Thus, we wished to see if we could implement common applications on different devices and yet maintain a broadly equivalent user experience.

Two prototype applications were built in Java 2 micro edition (J2ME), one to simulate mobile stock broking and the other to simulate the on-line purchasing of cinema tickets. J2ME allows applications to be compiled for both a Palm OS computer and a mobile phone without changing any of the code. There are obvious advantages to being able to write an application once and deploy it across different platforms. However, the benefits of this approach would be lessened if the usability of the application varied across the different devices, hence the reason for this study.

2.1. Movie ticket purchasing

Chittaro and Dal Cin [4] used a movie ticket purchasing scenario for their work. Movie ticket purchasing is a customer-driven activity in which the user requests information from a server and responds accordingly. We developed a simple prototype system that displays a list of available movie titles, a list of cinemas based on the user’s location and, for a chosen movie and cinema, the list of showing times. Users select their seat position and specify the type and quantity of tickets required (e.g. adult, student, child, etc).

2.2. Stock broking

Whilst stock broking supports the same transaction model, it also has a real-time event-driven aspect. Stock prices are served at regular intervals to the mobile device. Price thresholds are set for certain stocks and the device signals an alert when chosen stocks hit these thresholds. The user then

decides whether to buy or sell the shares.

2.3. Experiment

16 subjects (8 male, 8 female, all students on an MSc Computing course) carried out four tasks using the two applications and the two mobile devices (see Table 1).

Table 1 Experimental tasks

Task		Description
1	Mobile broking on the mobile phone	Subjects were asked to monitor the prices of various stocks via a stock ticker and a series of alert messages that were triggered by the application when particular stocks hit pre-programmed price thresholds. The main task was to buy and sell a number of certain stocks when their prices were between pre-specified lower and upper limits.
2	Mobile broking on the PDA	This was the same as Task 1 except that it was performed on the PDA rather than the phone. The details of the stocks to be traded were also changed.
3	Ticket purchasing on the mobile phone	Subjects were required to buy a range of adult, student, and child tickets for specified films showing at specified times at certain cinemas. Subjects were also required to find out information about showing times of certain films.
4	Ticket purchasing on the PDA	This task was the same as task 3 except that it was performed on the PDA and with different film, time, and cinema ticket requirements.

Each subject's performance (such as time taken, error rate, etc.) was logged automatically. Subjects' workload for each task was measured using the NASA Task Load Index (TLX) method [5, 6]. TLX allows comparisons to be made between tasks in terms of the mental and physical demands placed on the subjects.

To reduce the possibility of any task ordering effects, the subjects were randomly allocated to four evenly-sized groups comprising two male and two female subjects. Each group was allocated a different task order, thus:

Group 1: Task 1, Task 3, Task 2, Task 4

Group 2: Task 3, Task 1, Task 4, Task 2

Group 3: Task 2, Task 4, Task 1, Task 3

Group 4: Task 4, Task 2, Task 3, Task 1

Subjects completed a short questionnaire about their past experience of using mobile phones and PDAs, and their past experience of buying stocks and shares and cinema tickets. Instructions were given telling the subjects what stocks/movie tickets they were to buy/sell. Immediately following each task they completed a short two-part questionnaire. The first part asked a specific closed question that could only be answered by having used the application (e.g. "what is the price of the stock SYSB?"). The second part asked for responses (rated on a five-point "strongly disagree" to "strongly agree" scale) to nine statements about the task (e.g. "You would like to use this system to

buy and sell stock"). Then, subjects completed a TLX task-load rating sheet for the task.

Correctness scores for subjects were calculated for each aspect of a task that was successfully completed. The time taken to complete tasks was also recorded. As we were hoping to find no differences between application/device usability we specified the following hypotheses:

- H1—There is no significant difference in task duration between the devices.
- H2—There is no significant interaction between application and device on task duration.
- H3—There is no significant difference in correctness scores between the applications.
- H4—There is no significant difference in correctness scores between the devices.
- H5—There is no significant interaction between mobile commerce application and mobile device on correctness scores.
- H6—There is no significant difference in user satisfaction between the applications.
- H7—There is no significant difference in user satisfaction between the devices.
- H7—There is no significant interaction between application and device on user satisfaction.
- H8—There is no significant difference in workload between the applications.
- H9—There is no significant difference in workload between the devices.
- H10—There is no significant interaction between application and device on workload.

2.4. Results

For each task there were four sets of results to be analysed: the time taken to complete the task, the correctness score, the questionnaire responses, and the subjects' TLX workload assessments. The task results, together with the results of statistical analysis of variance (ANOVA) tests are shown in Table 2.

Table 2 Task results

	Stock broking		Ticket purchasing		ANOVA					
					Application		Device		Interaction effect	
	Phone	PDA	Phone	PDA	F	p	F	p	F	p
Task duration (sec.)	20	18	217	226	N/A	N/A	0.052	>0.05	0.114	>0.05
Percentage correct	94.53%	93.75%	92.19%	95.70%	0.006	>0.05	0.294	>0.05	0.726	>0.05
User satisfaction	71.39%	70.69%	78.19%	74.58%	3.053	>0.05	0.495	>0.05	0.227	>0.05
TLX score (workload)	48.44	43.12	25.44	29.44	15.278	**<0.01**	0.02	>0.05	0.984	>0.05

2.4.1. Task duration

Because more steps were involved in ticket purchasing (such as browsing film and cinema lists)

it is not meaningful to compare task durations between the two applications.

Duration for the tasks on the two devices is slightly different for both applications. However, the two-way ANOVA shows that these differences are not significant and so we accept hypotheses H1 and H2 and conclude that there was so significant difference in task duration between the two devices.

2.4.2. Task correctness

Correctness was calculated by awarding a mark for satisfying each component of the transaction. For example, marks were awarded for choosing the correct cinema, film title, showing time, etc. Table 2 shows a non-significant difference in scores between both the application and the device (p>0.05). The device on which the tasks were performed also had no significant effect on the scores (p>0.05). Finally, there was no observed interaction effect between the application and device types (p>0.05) and so we accept hypotheses H3, H4, and H5, and conclude that task correctness was not affected by either the application or the device.

The overall high accuracy rates (above 90%) suggest that the subjects understood the systems and the task requirements. Although the score differences were not statistically significant, the broking task on the phone and the ticket purchasing task on the PDA, whilst having similar means to their counterparts, had greater ranges of scores with some low outliers. It is possible that the higher mean for the PDA ticket purchasing arises from the difference in screen space on the two devices. Purchasing tickets required the user to look at lists of films, showing times, cinemas etc. The smaller screen on the phone meant that users had to do more scrolling which may account for the higher error rate.

2.4.3. User satisfaction

User satisfaction was measured for each task using a questionnaire. Percentage satisfaction scores were calculated for each task by deriving an overall rating for each subject's nine task-related responses and then computing the mean. From Table 2 we can see that for both applications the mobile phone had a higher mean satisfaction score than the PDA but that this difference is not significant (p>0.05). Therefore, we accept hypotheses H6, H7, and H8 and conclude that user satisfaction was not affected by either the application or the device. It is slightly puzzling that the ticketing application on the phone had the highest satisfaction rating yet had a lower accuracy score than on the PDA. This may be because only three of the 16 subjects had prior experience of PDA usage whilst all subjects had used mobile phones before. Thus, the relative unfamiliarity of the PDA

interface may have affected their opinions.

Analysis of the responses to the nine individual usability reveals that only statements 1 ("This system is easy to use") and 9 ("In general the system's response time was fast and you are satisfied with it") yielded results of statistical significance. For statement 1, an ANOVA suggests that ease of use was affected by the application (F=9.174, $p<0.01$) but not by the device (F=0.021, $p>0.05$) or by any interaction between device and application (F=0.187, $p>0.05$). Similarly, responses to statement 9 were significantly affected by the application (F=10.161, $p<0.01$) but not by the device (F=2.788, $p>0.05$) nor by the device/application interaction (F=0.023, $p>0.05$). This suggests that the device on which applications run has less impact on user acceptance than the applications themselves.

2.4.4. Workload

Using the NASA TLX method subjects assessed their levels of workload for the four tasks. TLX requires subjects to rate their experienced level of workload in five areas: mental demand, physical demand, temporal demand, effort, and performance. Subjects rank these five areas in terms of their perceived importance and contribution to the workload for each task. From these ratings and rankings an overall workload figure on a scale of 0-100 is calculated.

From Table 2 we see that the difference in workload between the two applications is highly significant (p<0.01). The devices themselves had no significant impact on the workload reported by the subjects. However, we do observe a slightly higher (though not statistically significant) workload rating for the mobile phone over the PDA in the broking application. This task required more data entry than the ticketing application (which had more list selection operations). It is possible that users might have found data entry harder to do with a phone keypad than with the PDA. (A single key on a phone keypad can be used to enter a digit, one of several upper and lower case characters, and one of a selection of punctuation/white space characters).

Thus we reject hypothesis H8 and conclude that the workload was significantly higher for the stock broking application. However, we accept hypotheses H9 and H10 and conclude that the devices and the device/application interaction had no significant impact on subjective workload.

3. Porting a wireless business roaming application

Overall then we find support for the idea that development environments like J2ME allow us to write applications once and deploy them across different platforms without significant impact on the user. However, we are aware that the types of interface components used were the same between the two devices. What this ignores is that whilst a generic interaction design can be ported across a range

of platforms, the different capabilities of different devices may mean customising the interface design for a specific device gives a better user experience. Chittaro and Dal Cin found [4] that different methods of implementing between-screen navigation and list selections on a WAP phone significantly affected usability. Also, the applications we developed were designed especially for the mobile devices. However, there is a need also to migrate applications from server/desktop environments to mobile platforms (e.g. warehouse picking lists), Therefore, in our second investigation we explored whether a Compaq iPaq PDA (which has a very different interface from the Palm OS machines) could provide a usable and useful interface to a an existing real business application.

Our chosen scenario was a worker needing to extract stock from a warehouse to fulfil an order. They would normally print a picking list and walk round the warehouse picking stock from the shelves. On return they would then update the stock control system with details of the picked stock. With a PDA they could update the system as items are picked, and perhaps update the main database to indicate stock items that could not be found so that a purchase order is raised with the supplier.

3.1. The prototype

For the prototype we wished to implement a single function area of a very large application to allow an heuristic evaluation of the potential of a handheld device to function as an extension of a mainstream product. The prototype should be able to demonstrate areas of strengths and weaknesses in user interaction of an interface used on such a device. The popular Sage Line 100 accounting system was installed on server and a prototype picking list application to interface with the server was installed on a Compaq iPaq 3850 with a Compaq WL110 wireless PC card.

Technologies like Wireless Application Protocol (WAP) allow functional and interface components of existing applications to be mapped across a range of mobile devices. However, although handheld devices can access and manipulate data like a conventional computer, existing server/desktop applications are not well suited to run on these devices. For instance, much of the information provided by the Sage 100 picking lists screen is not needed by a person picking an order. So, Figure 1 shows how the application was redesigned specifically for use on a small touch screen (rather than shrunk-to-fit).

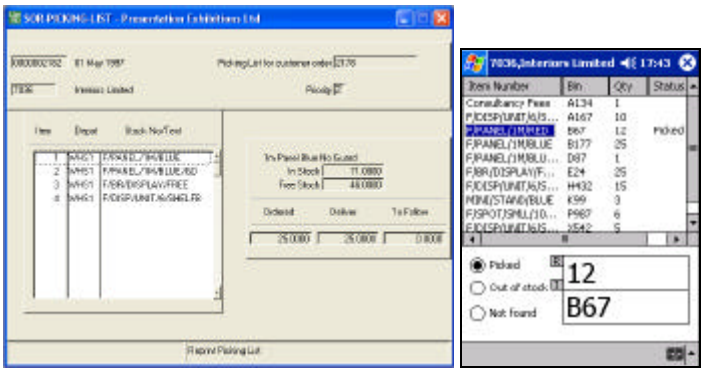

Figure 1 Desktop vs handheld picking list screens

3.2. Usability issues

The prototype system was evaluated by letting five experienced Line 100 users interact with the application. They were asked to perform a variety of stock manipulation tasks. No formal experiment was conducted; instead we asked the users for their impressions of the system with a view to using their feedback to develop a second version of the system that could be formally evaluated.

3.2.1. Drag and drop

The entire iPaq screen area can be accessed with a single stylus movement making its sylus-based drag and drop well suited to the manipulation of objects. Using the drag and drop model stock codes became the objects being manipulated which could then be dropped onto action zones labelled 'Picked', or 'OUT' (out of stock, see Figure 1). This helped to create what Norman and Draper described as "*…the sensation in the user of acting upon the objects of the task domain themselves*" [7]. Of course, this interface was not so suitable when a finger was used instead of a stylus; more of the screen was obstructed during the drag operation with the potential for hiding information required to complete the drop operation.

3.2.2. Colour and sound

The iPaq has a 256 colour capacity which is sufficient to provide a very clear and sharp interface. Good definition of controls is possible through the use of colours, but the screen was overwhelmed by strong sunlight. Also, colour can be used to indicate state, but this can have serious drawbacks for users who are colour blind. Indeed, a red-green colour blind user had difficulty in discriminating between small red and green objects.

Many of the newer PDA devices have audio capability, the utility of which is dependent on the environment's ambient noise. Reminders and warnings with audio attached to them work well when

in a low noise situation, but are more difficult to hear in the open or in noisy places [8]. Headphones would help with this problem but there are serious ergonomic and safety issues concerning their use in industrial environments.

3.2.3. Messages and status controls

There is no physical space for a status bar on the iPaq's forms, and so program status had to be obvious from the current form. Messages were thus provided either in a designated field on the form, or via popup message boxes. Another feature of a touch screen interface is that there are no mouse-related focus events, such as mouseover or mouseleave. So, the method of providing visual feedback to the user about the status of, for example, a clickable picture such as can be found on a web page cannot be used. This limits the feedback a designer can expect to provide a user on the status of any given control. Using sound to provide feedback is a possible way forward and Brewster [8] has described some of the issues associated with this.

3.2.4. Scrolling, lists and tree views

It is difficult to use dragging to control a scroll bar on a touch screen. This is especially difficult when the iPaq's Transcriber tool is turned on, as it sometimes interprets the drag as a gesture or line. It is easier to use the clickable components that move the bar, such as the arrow buttons at either end. An associated problem is handling lists of information. Tree views are difficult to manage on small screens, especially if they expand to many nesting levels. One option for managing large lists is to use the metaphor of a drum or cone of data. The idea is to have the middle of the drum (i.e. the part closest to you) enlarged with the edges of the drum compressed, and therefore providing more lines overall (see Figure 2a).

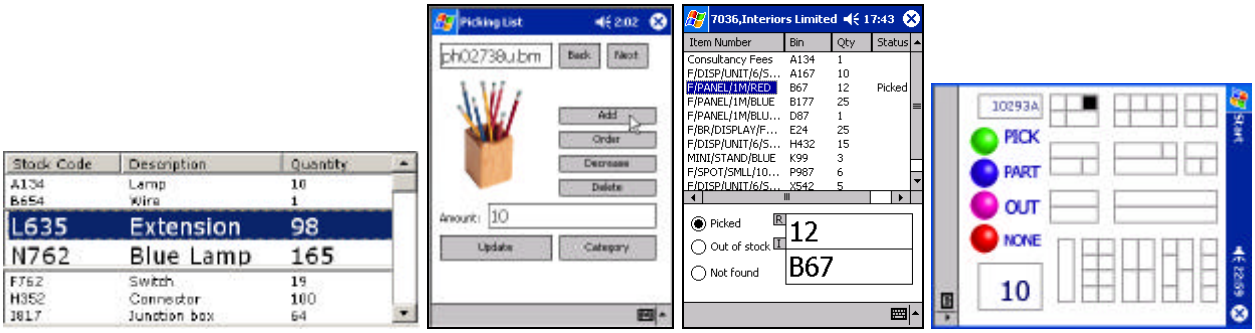

Figure 2 Alternative picking list representations. (a) shows a drum list view, (b) uses a single screen per item allowing good description and pictures, (c) is a standard list view and (d) uses a warehouse map to guide the user

3.2.5. Visibility of system status

Keeping the user informed of what the system is doing at any time was difficult given the iPaq's small screen area. In some respects the system status is not always obvious from the screens in the prototype. For example, the first version of a form showing a list of all stock items on the accounting system caused the system to appear to hang. In fact there was a delay between the extraction of the data from the server and its transmission and subsequent display making it appear that nothing was happening. This was solved by breaking the transmission into chunks and displaying the chunks as they were received.

This had two effects: users saw interface being updated and so knew the software was still functioning despite the screen not responding to input. Secondly, it managed expectations of response time. It also considerably increased the overall time for the data transfer. Users rated the faster option as more satisfactory, but this was only after a few uses. On first use of the form they were unsure what was happening when the refresh was based on the method that shows no status. Only after becoming comfortable with the interface did they then prefer the faster method. This is something to consider when designing an interface for use by novice and expert users alike.

3.2.6. Picking list interaction

The order of the items on a typical printed picking list is normally in the most efficient picking order. So the prototype allowed the user to walk through the warehouse looking for items to pick, presenting each item in the same order as a paper picking list.

There were several ways of assisting the users in this task, either by showing a picture of each item as it was selected or by showing the bin location of the item (see Figure 2b and Figure 2c). This was further enhanced by also locating the item on a representation of the warehouse (Figure 2d).

Users reported that the warehouse location form was logical and easy to use despite the lack of obvious buttons.

3.2.7. Handedness

A design issue that does not usually affect desktop applications is user handedness. On a mobile device the side of the screen on which controls are positioned is very important. When a user touches the screen with a finger or stylus the hand obscures part of the display. If the user is left-handed and the controls are placed on the right, the bulk of the screen will be hidden by the user's hand. On a touch sensitive screen the user's hand may inadvertently trigger an obscured control. Scrolling becomes slower as the hand must be moved out of the way to see the current list status. The prototype was built with the option to switch the usable controls to either side of the screen and left-handed users reported a preference for this facility.

3.2.8. Data entry

Every user had difficulty with the iPaq's Transcriber input software. Before attempting to write in a text box the user must activate the stylus's pen mode. For form manipulation the pen mode must be disabled. Users often forgot to do this and delay and confusion resulted when going from selecting a stock item (pen off) to changing a quantity or location (pen on). Some users commented that it would be better if the pen mode automatically turned itself on and off as the form context changed. This would have prevented the user from making mistakes on the form, but it is not possible to control the pen software from all programming platforms (e.g. Visual Basic).

Some users attempted to write in the spaces provided for the data display. For example, the quantity field is very small (see Figure 2b), but everyone tried to write in this box to change the quantity. When questioned they said they thought the text box was where the information was needed, so they presumed they had to write in that box. They did not expect the system to read the information from anywhere on the screen and put it into the correct place. Unlike the Palm OS devices which have a dedicated writing area, the iPaq's Transcriber software allows writing to be done anywhere on the display. Devices differences like this make multi-platform deployment problematic. The iPaq's data boxes appear to have writing affordances whilst its screen superficially resembles a desktop display which does not normally afford writing (keyboards provide the input), yet the iPaq's entire screen surface possesses an actual input affordance causing confusion amongst users.

A possible solution to this affordance/perception imbalance is to make sure fields are large

enough and in a convenient screen location to allow writing in. Some users tended always to write in white space, so a reasonably large area of blank space could be left for this purpose.

3.2.9. Screen aesthetics

Small screens make it hard to display lots of information at once. Lists had this effect in the prototype, providing a very compact display area with high information density. The small fonts required to display the volume of data made these forms less popular until users realised that only these forms gave feedback on the picking lists status. The tasks of reading and updating slowed down with these forms, whilst the large format forms were faster and reportedly easier to use (see Figure 2).

4. Conclusions

The first study indicated that the complexity and nature of the task itself appeared to be more important factors than the device on which they were run. This in turn suggests that provided a good task and interaction design is carried out, applications can be written once and rolled out across multiple platforms using technologies like J2ME. However, as the second study suggests uniformity of experience is probably achieved at the expense of the greater usability benefits that accrue from tailoring applications to exploit the interaction characteristics of individual devices.

The results of the pilot studies have provided interesting areas for further exploration. The results indicated higher error rates on mobile phone tasks that required scrolling through long lists. This aspect should be explored more formally to see what effect devices have on list scrolling tasks. Alternative representations could also be investigated to see how such data might be better presented on small screens. Brewster has provided some evidence that button size on a PDA can be reduced if auditory feedback is used [8] and Vickers and Alty have demonstrated that sound can be used to communicate quite sophisticated computing information [9-11].

For the picking list application the next stage is to build a more comprehensive version that makes good and consistent use of the features that received the most positive feedback. To evaluate these features, alternative versions will be produced that make sole use of the list view, large format, and landscape format forms, and a formal experiment conducted to allow quantitative analysis of the interfaces. User feedback supports the idea that mobile devices can be used to carry out business functions that have previously been restricted to desktop workstations.

Forms containing large amounts of data that work well in desktop environments were not well received by users of the prototype and simply converting a desktop application to run on a mobile

device by interface markup techniques will not work well. To take full advantage of mobile devices processes need to be re-designed for a mobile environment. It is unlikely that entire desktop applications can be (or should be) successfully migrated to today's handheld mobile devices. For example, due to the restrictions of large volume data entry, it would not seem appropriate to transfer invoice entry to a PDA. Of course, keyboard attachments can be connected to handheld devices to allow volume input, but doing this begins to remove the advantages of a handheld device: if a mobile platform with a keyboard is needed then a laptop/notebook computer would be a better choice.

At the moment, the choice seems to be between implementing an application on multiple platforms but with a generic interface that doesn't exploit device characteristics or tailoring an application for individual devices which requires source code amendments. Approaches based on extensible markup language (XML) such as XUL (XML-based User Interface Language) may provide a useful way forward in separating functionality from the interface. Assuming this challenge can be met, the secondary challenge of migrating server/desktop functionality to mobile devices requires ways of automatically selecting which elements of an existing application's functionality is suitable to be run on different mobile devices.

5. References

[1] A. Walker, S. A. Brewster, D. McGookin, and A. Ng, "Diary in the Sky: A Spatial Audio Display for a Mobile Calendar," in *People and Computers XV-Interaction without Frontiers: Proceedings of IHM/HCI 2001*, Alan Blandford, J. Vanderdonckt, and P. Gray, Eds. London: Springer-Verlag, 2001, pp. 531-539.

[2] J. Roth, "Patterns of Mobile Interaction," *Personal and Ubiquitous Computing*, vol. 6, 2002, pp. 282-289.

[3] M. D. Dunlop and S. A. Brewster, "The Challenge of Mobile Devices for Human Computer Interaction," *Personal and Ubiquitous Computing*, vol. 6, 2002, pp. 235-236.

[4] L. Chittaro and P. Dal Cin, "Evaluating Interface Design Choices in WAP Phones: Navigation and Selection," *Personal and Ubiquitous Computing*, vol. 6, 2002, pp. 237-244.

[5] NASA Human Performance Research Group, "Task Load Index (NASA TLX) v1.0 computerised version," NASA Ames Research Centre, 1987.

[6] S. Hart and L. Staveland, "Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research," in *Human Mental Workload*, P. Hancock and N. Meshkati, Eds. Amsterdam: North Holland B.V, 1988, pp. 139-183.

[7] D. A. Norman and S. W. Draper, *User-Centred System Design: New Perspectives on Human-Computer Interaction*. Hilsdale: Lawrence Erlbaum Associates, 1986.

[8] S. A. Brewster, "Overcoming the Lack of Screen Space on Mobile Computers," Glasgow University, Department of Computing Science Technical Report TR-2001-87 April 2001.

[9] P. Vickers and J. L. Alty, "Using Music to Communicate Computing Information," *Interacting with Computers*, vol. 14, 2002, pp. 435-456.

[10] P. Vickers and J. L. Alty, "Musical Program Auralisation: A Structured Approach to Motif Design," *Interacting with Computers*, vol. 14, 2002, pp. 457-485.

[11] P. Vickers and J. L. Alty, "When Bugs Sing," *Interacting with Computers*, vol. 14, 2002, pp. 793-819.