Evaluation of Wireless Usability with a Java Software Agent

Tobias Ceska and Stefan Kuhlins
Department of Management Information Systems III
University of Mannheim
D–68131 Mannheim
Germany
email: {ceska|kuhlins}@uni-mannheim.de

Holger Nösekabel
Department of Management Information Systems III
University of Regensburg
D–93053 Regensburg
Germany
email: holger.noesekabel@wiwi.uni-regensburg.de

ABSTRACT
This paper describes the value of usability, especially in the context of mobile devices. WAP sites offer companies the chance to provide mobile services to their customers or to improve the mutual relationship. In order to maximize the user experience, several usability criteria need to be followed. The development of such a catalog of criteria forms the basis for the implementation of our "Wireless Usability Software Agent" (WUSA). This agent applies these criteria to WAP sites and generates metrics that allow evaluation of the state-of-the-art regarding WAP sites. As far as we know, current solutions can test single sites, but neither the implementation of a comprehensive usability catalog nor an extensive data collection possible with an agent has been done and publicized so far. With these results, common problems and errors can be pointed out giving developers the chance to avoid and correct them.

KEY WORDS
M-Commerce, Mobile Usability and Evaluation, Software Agents, WAP Page Design Tools, Java Technology and Applications

1 Motivation

Over the last few years, two technologies have created new paradigms that influenced everyone's life. The first one is the World Wide Web, which enabled the exchange of multimedia information. Corporate sites allowed companies to disseminate detailed product information and establish a closer connection to their customers. The second technology is mobile communication and computing. EMC [1] estimates that 1,148 million people worldwide already own and use a cell phone. Another factor is that mobile phones are considered to be "trusted devices"—ready at hand, small, lightweight and often containing personalized information like an address book.

Soon after the first implementation of a mobile network, there was the users' desire to combine the two technologies: the mobile web was the idea that stood behind the wireless application protocol (WAP), which was initiated by Ericsson, Motorola, Nokia and Phone.com as they founded the WAP Forum Ltd. in 1997. The first draft of WAP was released in 1998 [2, pp. 2–3]. WAP and its XML-based markup language WML were designed to heed the limitation of the mobile devices, most notably the small screen size (4–6 lines of text with about 12 characters per line), the limited bandwidth (up to 14.4 kbps with GSM at first) and the monochrome display.

Users in a mobile context have unique requirements. For one, when using a mobile service they are in search of specific information and want to retrieve it very quickly—they are "hunting" instead of "surfing" [3, p. 16]. Sites with good usability allow users to achieve this. Finding the right information quickly also helps to keep the costs down for the user, another factor that builds user acceptance and contentment. Still the question remains what makes a site usable, especially with regard to a mobile context and how usable are WAP sites. While a single content provider can conduct usability studies for his service, an analysis of current WAP sites can point out common mistakes. Knowing these problems is a first step towards avoiding them, enabling the usability study to focus on other issues that cannot be measured in an automated way.

In this paper, we describe our software agent for automated evaluation of WAP sites. In Section 2 related evaluation methods of mobile services are discussed. Section 3 suggests a catalog of possible usability criteria, which also represents the criteria analyzed by the agent. The agent itself is presented in the fourth Section. Section 5 lists the generated data, selected results, and most common mistakes the agent discovered in current WAP sites. Section 6 draws a conclusion and points out further development steps for the software agent.

2 Related Work

Besides evaluating a complete site with regard to usability, the service provided can also be subject to evaluation. One approach to review the feasibility of a mobile service is the COMPASS approach as suggested by Amberg, Hirschmeier, and Wehrmann [4]. COMPASS collects and displays data over five areas: the situation concept, interaction model, usage cycle, system architecture, and user acceptance. User acceptance is queried by merging the basic idea behind the balanced scorecard [5] with the technology-acceptance model [6], which results in a matrix with two dimensions on the x axis (usefulness and effort) and two dimensions on the y axis (services and determining factors of the services). The elements of the matrix are perceived usefulness (usefulness and services), operability (effort and services), mobility (usefulness and determining factors), and cost (effort and determining factors). For each of those four elements a number of questions regarding a) the first and b) the continuous usage are generated and queried (for example social acceptance, configurability, monetary cost, infrastructure, and availability). Afterwards, the results can be displayed in a radar chart.

The COMPASS approach allows a service to be evaluated from a users point of view. It can be applied to any mobile service, not just to WAP sites, but detailed information about the usability cannot be gained other than the replies from the interviewees. Another approach is WebQual, which has been extended to WAP sites [7] or conducting interviews with users who have done specific use cases [8].

Some companies offer services for testing WAP sites very similar to our agent. Most of them categorize the features of their tools in functionality-, usability-, and load testing.

The NCC Group's mCheck covers all three categories [9]. They promise: "Don't waste your time pushing buttons ..." because manual testing by the developers is too time consuming and therefore too expensive. Instead you can use mCheck to test your WML code for specific handsets and for syntactical correctness. Afterwards, your WAP application should operate efficiently on different mobile devices and gateways. They ensure usability and a consistent "look and feel" after the test.

The NCC Group cooperates with Motorola. Together they published the *Motorola Certification Program – WAP Certification Developers Guide* [10]. The complete test of a WAP site for the certification costs $2,000. The criteria of this guide are checked by mCheck and mLoad. The latter is a performance tester. mCheck works very similar to our agent. At first, it sends a HTTP request to the application's server and downloads the start page. Then all referenced pages are validated against the WML DTD and links to missing pages, cards and images are reported as well as other errors. All WML code is recorded and can be viewed later.

The *e-Test Suite for WAP* is the analyzing tool of Scientific Computers [11]. It provides a way to test the functionality and scalability of wireless applications and traditional web applications extended for use with wireless devices. These Web testing software tools perform load testing, functional and regression testing as well as application monitoring throughout application development and post deployment. You can record interactions with a wireless application using a PC-based phone simulator. The WML Document Object Model of each page is saved and test cases to validate the objects are automatically inserted. Without any programming knowledge, it is possible to develop *WAP Visual Scripts*, which are based on the Document Object Model of the WML decks and can be used to configure your tests. *e-Load for WAP* reuses recorded tests to emulate thousands of virtual users accessing the application simultaneously.

The *Testing Service* of the *Open Group* [12] is strongly oriented at the *Certification and interoperability testing program* of the WAP Forum [13]. They provide a possibility to test the products before the companies apply for the certificate. Their advantage is that they have identical tests available in the WAP certification programs because they are the certification authority. They offer an online service as well as licensed products. The prices differ strongly: a 30 day access period to their online testing services costs $950 for an individual, whereas the license fees for the downloadable testing services are up to $48,000.

Argogroup's *Monitor Master* is another product for testing WAP sites [14]. With *Monitor Master* you can automate the measuring and monitoring of the quality and usability of mobile content. The *Monitor Master* comes in three configurations, tailored to meet functional, monitoring, and load testing needs. It simulates real users accessing mobile data services. Scripting and auto browsing perform this. The content is analyzed against compliancy and conformance to style guides, interoperability and usability on a wide range of devices. The *Monitor Master* tests and monitors from multiple start URLs against multiple devices.

TestPros' WAP Test Lab Services [15] help to identify, isolate and fix problems with micro-browser compatibility, wireless device compatibility, network interoperability and application performance. The functional tests are done with a combination of handheld devices and wireless device simulators. Their load tests include stress tests to determine system performance. The usability tests evaluate not only the readability and accessibility of content (navigation, language, etc.) but also the consistency of your m-commerce with your e-commerce site.

Finally, you can also use a simple XML parser to check at least the correctness of your site if your WML development environment does not do this. Moreover, there are some free tools to test the availability of linked pages.

3 WAP/Handheld Usability

Within the domain of the World Wide Web, companies soon realized that user experience could be improved by observing usability rules. As a consequence, several catalogs of test criteria were developed that can be applied to most sites. Due to the differences between the wired and the wireless web, those catalogs cannot simply be transferred but need to be adapted to the new situation [8, p. 4]. Some of these include the WAP browser manufacturer development guidelines. For example Openwave, Sprint, Motorola and Nokia all provide documentation of how to use their browsers' capabilities to full extent, though their documents mostly cover technical aspects.

In this Section, we create a more comprehensive catalog of usability criteria categorized in five different sections: technical criteria, layout and design, services, user-friendliness, and finally, information presentation. This modular structure can easily be adapted and extended depending on the individual requirements.

Criteria in the technical section can be measured and contain the *time needed to load a WML deck*, the *deck size*, the *availability of the deck* and the *number of errors in the WML code*. The loading time of the deck is a direct result of deck size and achieved data rate with the end user device. Therefore, it is advisable to run tests with several different types of connections (GSM, GPRS, HSCSD), at different times (in the morning, at noon, in the evening) and with different providers—or at least tests within the expected user scenario if it is known with which type of connection and at what time the service will be used most likely. Deck size is also important since the default maximum deck size, as specified by the wireless session protocol, is limited to 1,400 octets [16, p. 81]. Although the client can override this default value, many older mobile phones and some current browsers adhere to this limit. This can create user irritation, since the service might work with one device but not with another and the reason for failure is not apparent.

The second section, layout and design, is problematic for two reasons. First, the display of the mobile device is too small to allow neither the presentation of extensive information nor the exact placement of items on the screen. (WML does not provide the syntax for this, most mobile phones are only able to show characters and the number of pixel on a display varies widely.) Second, the WBMP graphics format only allows two colors, and many mobile phone displays are only black and white. Designers are therefore unable to utilize colors as additional means to convey information or adhere to colors possibly defined by the corporate identity. What can be achieved, though, is that a user retrieves the desired information as quickly as possible. For this, *each element on a displayed page is classified* either as informative (information or content relevant to the user), functional (navigation) or decorative (logo or bullet with no function or information). A page can also be put into one of these categories, depending on the dominant element type. For example, a card with several functional elements (hyperlinks) and very few and small decorations (graphics) is counted as a menu page. The overall design of the site should also reflect the *corporate identity* as close as possible, although this basically means the usage of company logos and therefore is hard to measure automatically. Also, the design should be homogeneous—links should always be formatted the same way and placed at the same location of each card (at the top and bottom of each card).

Services offered by the WAP site are examined in the third section. Whereas the *quality of the content* is relative to the needs of the user (and thus not subject to measurement), the *availability of the content* can be measured by checking for working links within the site. Dead links are cumbersome for the mobile user, especially if they occur deep within the system, after he already has spent some time on navigation. Another service that the WAP site should offer is the possibility to send *feedback* to the content provider. This could be a phone number (preferably with WTAI, allowing the phone to dial this number directly [17]) or a server script for e-mail messages.

One point with user-friendliness is the *mobile usability*. A user should be able to use a mobile service without prior customization at a networked PC and all preferences should be changeable with a mobile phone if necessary. *Navigational items* should be present on every card since they assist the user. The site should not rely on the browser for providing these items—even though many browsers do provide “back” and “home” buttons, not all of them do and the manner in which these functions can be activated differs from browser to browser.

The section about information presentation regards the content that the user receives based on his actions—which means in most cases navigating through the site, although the input from external data sources like positioning services for location based services can also be factored in. One metric that can be computed is the *information density* by calculating the ratio between informative elements and the total number of characters displayed. A high ratio means that much information is presented to the user, but the page is less concise due to the small screen. Therefore on pages that qualify as informative pages, a high information density is sensible whereas on navigational pages (menus) a low density is preferable. The same holds true for the *length of a page*. Forcing the user to scroll is acceptable when the information presented is read in a serialized way. On menu pages scrolling should be avoided or at least frequently used items should be placed at the top. This does not circumvent the problem that the items at the bottom are not visible to the user and he will not know about those features unless he decides to scroll down posing the risk that they are likely to be overlooked.

A hypertext flowchart or a distance matrix can graphically represent the navigational structure of a WAP site. Botafogo, Rivlin and Shneiderman [18] proposed several hypertext measurements, two of them being *compactness* and *stratum*. A high compactness suggests many references within the site—this gives the reader freedom to choose which page to view next, but also increases the change for disorientation since the hypertext path of the previous pages is less obvious. The stratum on the other hand reflects the order for navigating through a site. A high stratum hints at a linear structure, whereas a low stratum may again cause “lost in cyberspace” and overtaxes the user with too many links. Additional information can be gained by analyzing the server log files of a site, this makes it possible to generate a path for each user: where did he enter the site, where did he go, which pages did he view multiple times.

In total, this catalog of criteria is less device and browser specific, although Openwave for example lists some of the criteria in their WAP usability guidelines [19]. They note that on small screens important content may lost if it is too low on a given page and that the WTAI protocol should be used whenever possible. Furthermore a maximum deck size of 500 bytes (encoded WML) is recommended to keep latency down.

4 WUSA: The Wireless Usability Software Agent

4.1 Program Description

Our agent is a Java program that evaluates WAP pages and presents statistics of the collected data. We chose Java because it offers sophisticated class libraries for networking, regular expressions and multithreading. First, the agent retrieves the WAP start page of a web site and evaluates it. Next, the agent finds and evaluates all linked pages. In the following, the pages of a web presence altogether are called *web*. After an analysis of various criteria (see Section 3), some statistics and an overview of the detected errors are created and saved in a database. The statistics of all pages of a web are summarized in *web statistics*, which then can be summarized to *branch* and *overall statistics*. Java Servlets show the results as dynamically created HTML pages.

The WAP start pages of all webs that should be evaluated are listed in a file the program reads in. Besides the start page of each web and additional URLs that should be counted as internal, there is also some common information like the name of the company, the URL of their homepage and the company's branch.

The domain of the start page determines which linked pages belong to that web, namely all pages whose URLs share the same prefix. However, sometimes the start page has an abbreviated URL that is easier for the user to type in (e.g., http://wap.companyname.com/), whereas all other pages of the web start with a different and longer prefix (e.g., http://www.companyname.com/wap/products/a-m.wml). Since those URLs should not be handled as external URLs you can specify them in the list. Moreover, the agent compares IP addresses for different domains to decide whether a domain belongs to the start page. The agent checks the reachability of external links in order to determine and count broken links.

The program stores every evaluated WML and WBMP file in the database. With the help of saved files it is possible to view the source code of every fetched page and compare it to a version retrieved earlier.

At first, the start page of the web is evaluated. Every page is downloaded via an *HttpURLConnection* whose accept header is set to *text/vnd.wap.wml*. By doing that, the *HttpURLConnection* acts like a real mobile device. This is necessary because some servers would respond with an HTML answer otherwise.

The agent uses the *DOMParser* of *Xerces* [20] to process and validate the code. If an error occurs, it is stored in the database for later error statistics calculation. Next, the agent analyzes the tags of the actual page. Then it classifies the page as functional, informative or decorative. Last, it traverses all references to other files (WML and WBMP). Links to other sites are recorded and the pages are processed in the same way, whereby for each page, a new thread is started.

The area taken up by characters and images and the number of corresponding tags determines the classification of a page as functional or informative. If the functional content of a page outweighs the informative content, and the functional area outweighs the informative area, the page is classified as a functional page. Otherwise, it is an informative page. Functional text and images are surrounded by the tags `<a>`, `<anchor>`, etc. Informative text is found in tags like `<p>`, `<i>`, `<b>`, and so on. Informative images exceed 20 pixel in both dimensions. Otherwise they count as decorative images like lines, bullets, etc. The size of a WBMP can be determined from its header information, whereas the size of each character defaults to a given value as the display of characters varies on different mobile devices.

The links of a web are not only collected to acquire the complete web, but also for recording on which page a link was found. In this way, it is possible to calculate a distance matrix that aids in analyzing the navigation structure of the web.

All data resulting from the evaluation is saved in a database. Besides the items already mentioned, this also includes more easily obtainable facts, e.g., file sizes and status codes.

The evaluation of a web continues until all URLs of the collected list are processed or certain limits are reached (500 pages per web or maximal link depth 10).

When all pages of a web have been visited, the response times of some random pages are checked for every link depth. This cannot be done during evaluation since at that time more threads are running and therefore an exact measurement is not possible.

In a next step, web statistics for the evaluated web are created. The page statistics are averaged and minimum as well as maximum values are saved.

After all webs of the list file have been processed, the actual overall statistics are calculated and saved in the database. The overall statistics consist only of the latest evaluation of each web.

4.2 User Interface

Start Page: The results of the evaluations can be viewed via Java Servlets. The start page allows access to the overall statistics and all web statistics, which are kept in the database. Additionally, branch statistics can be calculated on the fly. Within the web statistics, it is possible to search for company names, also limited on a certain branch. The shown web statistics can be sorted alphabetically, by branch, number of pages or evaluation date.

Overall Statistics: Besides presenting the data of the actual overall statistics, this page also allows comparing the actual overall statistics with earlier overall statistics. Furthermore, the chronological development of the number of evaluated pages and the percentage of erroneous pages are illustrated by graphs.

Web Statistics: In addition to the data of an evaluation, the web statistics page also provides the ability to compare the current to earlier evaluations and any other web statistics. Hereby, it is possible to compare two business competitors' web statistics directly. Analogously to the overall statistics, the chronological development of the number of evaluated pages of this web and the percentage of erroneous pages are presented in graphs.

Details View of a Web: The details view of a web contains links to all pages found in the web. They are arranged in a tree that shows how many clicks a mobile device user needs to get from the start page to every other page on the shortest way. The calculation of the shortest way path and the distance matrix, which can also be found on this page, is performed using the shortest path algorithm of *Dijkstra* [21]. Furthermore, there are links to an image and an applet, which illustrate the navigation structure through a graph.

Page Details: As previously mentioned, not only the evaluated data is shown, but there is also the possibility to compare with the data of earlier evaluations. By clicking one of the links to WAP simulators, the page can be viewed. The source code of the page is displayed with syntax highlighting and links to XML errors. When comparing to another version of that page, the differences in the source code are emphasized. The comparison can help developers to repair errors, which appear only in newer versions. A link to a small overview with details of the images of the page and the WBMPs shown (converted into PNG) completes the page details servlet.

Search: As aforementioned all WML pages are saved in our database. That is why we can also offer a fast WML search engine. You can search for keywords and even full text.

5 Evaluation

At first, our evaluation covers the WAP sites of Forbes' Global 500 companies [22]. For that purpose we implemented another agent that retrieves the URLs of these 500 companies' homepages and searches for WAP sites. Several constructions of possible start URLs for WAP sites were tried, e.g. http://www.companyname.com was transformed into http://wap.companyname.com, http://www.companyname.com/index.wml and some others. The surprising result was that less than 10% of those companies run a WAP site. After that, we were looking at WAP directories like *wapjag.com*, *wapjag.de*, and *mobile.yahoo.com*. Altogether we analyzed nearly 1,000 WAP sites (see Table 1). The latest results can be found at WUSA's homepage at http://sconner.wifo.uni-mannheim.de/wusa/.

Number of analyzed WAP sites	946
Number of analyzed WML pages	97,568
Imperfect pages	32 %
Functional pages	43 %
Informative pages	40 %
WAP sites with feedback mechanism	49 %
Average size of deck	1,230 bytes
Average time to load deck	1,033 ms

Table 1. Summary of key criteria

The majority of the evaluated sites are error-free, but about 32% are imperfect. Nevertheless, mobile devices can display most of them. For example, if there are broken links or missing pictures mobile devices will still present the page content. Conspicuous is the fact that more than every second link to external pages leads to an erroneous page and even the half of those links reference not available pages.

The most frequent error is that the content length is too big. The sizes of about 78% of faulty pages exceed the limit of 1,400 bytes. Other typical errors are violations of the WML DTD so that the respective page is not in valid XML format. Very often attributes or elements that are not declared in the DTD are used. Also required attributes are sometimes missing or the attributes are not embedded in quotes. Some developers also forget end tags or write illegal content in the prolog or use invalid characters or ids. In some cases links lead to HTML documents what makes no sense for mobile devices.

The response times of the evaluated sites vary widely. Some pages send their responses in less than 100 ms whereas others need umpteen seconds. The averaged response time with our Internet connection, which is definitely under one second, is fortunately low. But you have to consider that the response time of the agent is a bit faster than the time a page needs to be displayed on a mobile device, because in contrast to mobile devices the agent bypasses the WAP gateway.

The average web consists of about 100 pages and has a maximal link depth of five. Nearly every second web offers at least one feedback possibility (that may be an e-mail address, a phone number or something like that). In comparison, only 7% of the pages offer one. The average page has about 210 readable characters and consists of nine rows, which makes a size of 1230 bytes. Additionally every fifth page has a picture what lets the average size of bytes to be loaded increase by 73 bytes.

A usability study conducted by Ramsay and Nielsen [8] in 2000 supports the findings, even if many of negative aspects result due to a slow and unstable connection – this emphasizes the point that the deck size has to be small enough to allow for low latency. Also, the user of the study reported several problems with large decks that were not displayed correctly on their device. Another problem was that pages with many lines in a menu hid the lower links, which is consistent with the observed average of 9 rows per page.

6 Conclusion and future enhancements

In contrast to the commercial tools WUSA can create statistics to deduce common statements/results. This is caused by the fact that WUSA analyzes loads of webs and is not straightened to test the WAP site of only one company.

Future enhancements might include the analysis of location-based services. Currently this is not feasible, since the required bandwidth and computing power is only available in a PC based system connected to the Internet. This PC is not mobile in itself, so the data gained by locating the device does not reflect real usage where the user is changing his location constantly. As mentioned, the quality of the replies depends on the underlying database of the service and can only be judged by human users unless the agent itself has an extensive database that enables a comparison of the results delivered. One way to collect location dependent information would be to run the collecting software on a mobile device and relay the data to a central server. Alternatively, the agent could be run entirely on a mobile device like a PDA or laptop, which provide sufficient computing power and connectivity. Or the provider generates different location information for the agent; this is technically possible although it requires the provider to modify the location server.

WUSA should also be able to evaluate future formats of mobile web sites. Especially if a format of the next generation will be XHTML, only a few modifications of WUSA's source code will extend its functionality to do the same work for XHTML pages. Other updates like additional image types are even easier to manage, so that the first steps to cover the renewals of WAP 2.0 can be taken in the near future.

As mentioned above, some companies of the presented testing tools work together with certification organizations and thereby can prepare your WAP site for the certification test or obey special style guides. Our testing tool, on the other hand, works independently and is strongly oriented towards the official WAP specifications [23].

References

[1] EMC Wireless World Map, http://www.emc-database.com/

[2] Marcel van der Heijden and Marcus Taylor, *Understanding WAP: Wireless Applications, Devices, and Services* (Boston: Artech House, 2000).

[3] Scott Weiss, *Handheld Usability* (West Sussex: John Wiley and Sons Ltd., 2001).

[4] Michael Amberg, Markus Hirschmeier, Jens Wehrmann, Ein Modell zur Akzeptanzanalyse für die Entwicklung situationsabhängiger mobiler Dienste im Compass Ansatz, *Proc. 3rd Workshop Mobile Commerce*, Augsburg, Germany, 2003, 73–87.

[5] Robert S. Kaplan and David P. Norton, *The Balanced Scorecard – Translating Strategy into Action* (Boston: Harvard Business School Press, 1996).

[6] Fred D. Davis, Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology, *MIS Quarterly*, 13 (3), 1989, 319–339.

[7] Stuart J. Barnes, Kenny Liu, and Richard T. Vidgen, Evaluating WAP News Sites: the WebQual/M Approach, *Proc. of the 9th European Conference on Information Systems*, Bled, Slovenia, 2001.

[8] Marc Ramsay and Jakob Nielsen, *WAP Usability – Deja Vu: 1994 All Over Again* (Fremont, Nielsen Norman Group, 2000).

[9] The NCC Group, Tools for Wireless Internet Testing, MCheck, http://www.nccglobal.com/testing/mi/mcheck_downloads/mcheck_tools.htm

[10] The NCC Group, WAP Application Certification, http://www.nccglobal.com/testing/mi/images/wapapplication0103.zip

[11] Scientific Computers, e-Test Suite for WAP, http://www.scl.com/products/empirix/datasheets/wap.html

[12] The Open Group, The Open Group's Testing Services, http://www.opengroup.org/testingservices/wapts.html

[13] The WAP Forum, Certification and interoperability testing program, http://www.wapforum.org/cert/go-cert.html

[14] The Argogroup, Monitor Master, http://www.argogroup.com/products/monitormaster/architecture.shtml

[15] TestPros, WAP Test Lab Services, http://www.testpros.com/testingServices/e_commerce/waptest.asp

[16] Wireless Session Protocol Specification, http://www1.wapforum.org/tech/documents/WAP-230-WSP-20010705-a.pdf

[17] Wireless Telephony Application Specification, http://www1.wapforum.org/tech/documents/WAP-266-WTA-20010908-a.pdf

[18] Rodrigo A. Botafogo, Ehud Rivlin, and Ben Shneiderman, Structural analysis of hypertexts: identifying hierarchies and useful metrics, *ACM Transactions on Information Systems*, 10 (4), 1992, 142–180.

[19] Openwave Systems Inc., Top 10 Usability Guidelines for WAP Applications, http://developer.openwave.com/resources/uiguide.html

[20] Apache Software Foundation, Xerces2 Java Parser, http://xml.apache.org/xerces2-j/

[21] Edsger W. Dijkstra, *A note on two problems in connexion with graphs*, Numerische Mathematik 1, (Berlin: Springer, 1959)

[22] Forbes, The Global 500, http://www.forbes.com/global/2002/0722/global.html

[23] WAP Forum, WAP Specifications, http://www.wapforum.org/what/technical.htm