WapSearch: A system for searching the Web using WAP enabled devices

Duncan Mason, Joemon M Jose, and Anu Joseph

University of Glasgow
Glasgow G12 8QQ
{masondx,jj,anu}@dcs.gla.ac.uk

Abstract. The World Wide Web is a rich source of digital information. Web search engines allow users to locate resources on the Internet thus helping them to satisfy their information needs. The advent of the Wireless Application Protocol (WAP) has made wireless Internet access possible. However, WAP users do not have access to the same quality of online Information Retrieval systems as 'wired' users. We present a system, named WapSearch, that enables WAP enabled mobile phones and PDAs to search for and browse Internet content written in HTML. The WapSearch system is based around an HTML/WML conversion tool and a popular Web search engine and employs a novel document navigation functionality designed to aid the Information Retrieval process on these devices. In short, we support sophisticated document browsing by bypassing the (typically small) cache memory of the client device, instead exploiting the server's much greater memory and processing capabilities.

1 Introduction

Recent advances in mobile computing technology have made it possible to deliver Web content to mobile devices using the Wireless Application Protocol (WAP). Owners of WAP enabled mobile phones and PDAs now have access to a variety of on-line services previously available only to desktop computer users. However, WAP devices are unable to directly browse resources authored using HTML. Instead, WAP applications are written using the Wireless Markup Language (WML)[Arehart et al., 2000].

In order to serve the emerging mobile Internet user community, many content providers will maintain separate Internet and WAP services. This involves information authoring in both the afore mentioned markup formats. Duplicating content is undesirable because it can lead to problems with data consistency. If consistency problems do occur, the quality of information and, therefore, services available can be degraded. Essentially, information duplication has the effect of isolating wired and wireless web content, leaving WAP users with access to only a small fraction of the information resources available on the Internet.

Wireless Internet technology has great potential because of the immediacy of access to information it allows. Yet this potential will remain unfulfilled until all Internet users have access to the same information services regardless of the devices they are using. There are two ways to approach this problem of content fragmentation. One is for content providers to implement a Web publishing framework in which information is separated from its presentation format. This can be achieved by applying format to information dynamically using a mixture of technologies such as Java, XML and XSLT[Arehart et al., 2000]. The Apache Software foundation's 'Cocoon' project is an example of a package that supports this kind of functionality (*http://www.apache.org/cocoon*). The main disadvantage of this approach is that it places a short term setup overhead on the content provider. Although it may produce long term benefits, many organisations are unlikely to convert to this model of web authoring any time soon so fragmentation will persist. Also, this strategy is probably unsuitable for organisations or individuals with a minimal presence on the Web.

Another approach is to empower the client device and make it possible for the user to browse content written in any format and this is the approach we have taken. A server side system we have created aims to counter content fragmentation problems by giving WAP users access to HTML resources. In the following sections, we provide a brief introduction to the WAP paradigm, describe the key design features & functionality of our system and discuss how the WapSearch system can be extended to take advantage of new advances in mobile Internet technology.

2 WAP & WML Background

WAP defines an application environment and a set of communication protocols designed to bridge the gap between the Internet and the increasingly large number of mobile computing devices in circulation. Developed by mobile communications manufacturers such as Nokia, Ericsson and Motorola, it is the de facto world standard for advanced Internet communication on mobile devices. WAP design is intended to support the communication of data over typically unreliable, low bandwidth mobile networks such as GSM. The basic structure of the wireless Internet is shown in Figure 1 (left).

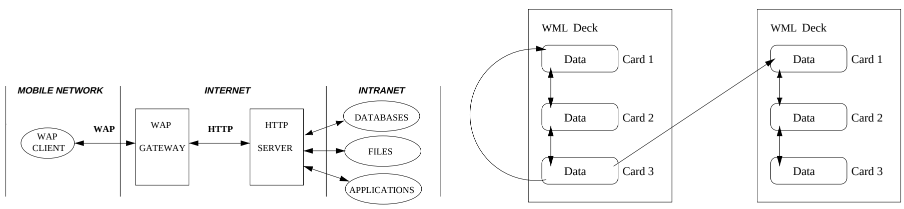

Fig. 1. A high level illustrations of WAP topography & Deck Structure

WAP content is stored on file servers in the same way as regular HTML files. Wireless devices communicate with file servers known as WAP gateways.

These machines translate WAP requests in to HTTP requests and fulfil them by communicating with other file servers on the Internet. HTTP responses received from the Internet are then converted in to WAP format by the gateway and forwarded to the mobile client.

WML: Micro-browsers residing on WAP devices read content written in WML. This markup language, like the WAP protocol is designed for devices with low bandwidth network connections, limited memory and restricted I/O capabilities so the tag set defined by the language is small, giving the developer a small number of text and image formatting options.

WML files are structured as one or more WML 'Card' elements embedded within a single WML 'Deck'. Card elements contain the data to be presented to the WAP user and are capable of linking to other Cards in the same Deck as well as other WML Decks. The Deck/Card navigation metaphor is illustrated in the Figure 1 (right), the diagram shows how data held in cards can be linked together. The WAP 1.1 specification recommends that the size of a WML Deck should not exceed 1.4 Kilo Bytes. This reflects the hardware limitations of many of the WAP enabled devices currently in circulation.

3 WapSearch Architecture

The WapSearch system is based around an HTML/WML conversion tool and the AltaVista search engine[1]. It allows a WAP user to search for HTML documents as they would using a desktop PC. The results of a query are returned as a result set of hyperlinks and document summaries. The user can then follow these links to HTML documents which are converted 'on the fly' by our HTML/WML conversion tool in to a format that the WAP micro-browser can read. This method makes virtually all HTML content available to WAP users and does not rely on the content provider having implemented a full Web publishing framework.

Figure 2 illustrates the architecture of the system. The WapSearch system was developed in Java. The most significant system components are introduced below.

- **The WAPServlet**
  A component written using the Java Servlet API that handles and services requests from mobile clients. It will analyse a request and possibly invoke the services of another component to perform necessary low level work in order to formulate a response. During this process, the Servlet interrogates HTTP request headers to identify the client device. This information is used to format the size of WML files created as a result of the conversion.
- **The 'Loader' Module**
  This module wraps around the AltaVista search engine, passing user-supplied query terms to the facility and converting the search results in to WML format.

[1] Although we have initially used AltaVista, a wrapper could easily be created for most Web search engines

- **The 'Converter' Module**
  Another module which makes it possible to convert arbitrary HTML files in to WML format.
- **The 'WML' Module**
  This module provides services which support the dynamic creation of complex WML Decks and Cards within the context of a single converted HTML document.

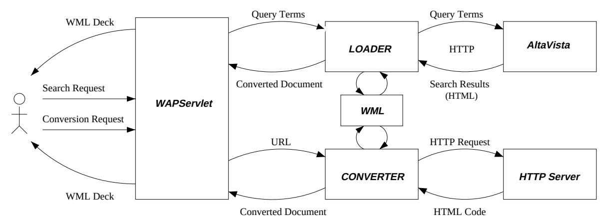

Fig. 2. WapSearch Architecture.

3.1 Search Phase

The primary function of the WapSearch system is to provide an information retrieval service to WAP users. On receiving a request for the WapSearch search facility, the WAPServlet module generates the interface shown in Figure 3. The user can enter multiple query terms in the input element and initiate a search on these criteria by selecting the hyperlink labelled 'Do Search'. This passes the query terms from the client device to the WAPServlet module which in turn employs the Loader to generate a WML representation of the AltaVista results page. The first 10 results are then returned to the user.

3.2 Conversion & Navigation Interface

Each search result produced by the Loader module is embedded in a separate WML Card and is displayed as a hyperlink accompanied by a brief summary of the target document (see figure 3). By selecting one of the result links, the user generates another request to the Servlet with the target URL as parameter. The Servlet then invokes the Converter module on this link, passing the embedded URL as parameter. The target HTML document is converted in to WML format by the Converter and presented to the user. Hyperlinks in the source HTML document are inserted in the WML code, if one of these links is selected by the user, a further execution of the conversion phase is triggered. This pattern may be repeated as many times as the user requires, thus allowing them to mine for information as they would on a desktop computer.

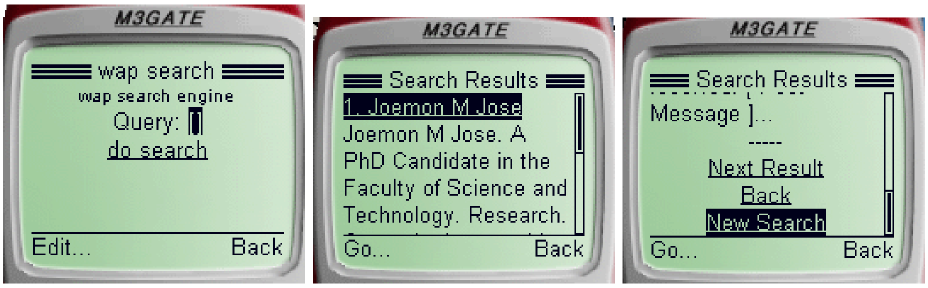

Fig. 3. Search Interface & Results Page Design.

WAP applications typically suffer from extremely poor usability[Ramsy and Nielsen, 2000]. This is due, in part, to the I/O limitations of current WAP devices (small screen, limited keypad etc.), however, poor usability can also be attributed in many instances to poor interface design[Ramsy and Nielsen, 2000]. WAP and WML are in their infancy and as a result, few effective interface design conventions have been established. A large part of our effort in creating the WapSearch system has therefore been invested in developing an interface that aids the information retrieval process and helps the user satisfy their information needs quickly.

As mentioned earlier, technological limitations mean that most WAP devices can only handle WML Decks up to a maximum size of 1.4 Kb. Obviously, most HTML documents are much larger than this so we must fracture HTML documents during conversion and create a number of smaller WML Decks. The WML Decks are packaged together in a single data structure encapsulating the concept of a converted document and stored on the server. The user may perform many iterations of the conversion phase, by using HTTP session management we can build up a session stack of converted documents viewed by the user. This allows us to provide support for advanced navigation between the documents the user has chosen to convert.

A single converted HTML document may result in a large number of fragments so we created an interface that allowed the user to move easily between the fragments of a given converted document and sections of the other converted documents on the session stack. We therefore provide the user with a number of navigation choices at the end of each converted WML Card (as shown in Figure 4). These links allow the user to:

- Navigate to the next fragment of the current converted document.
- Navigate back to Deck preceding this in the current document.
- Navigate back to the first Deck in the current converted document.
- Navigate back to the first Deck of the previous converted document on the session stack.
- Navigate back to the AltaVista search results page (always at the bottom of the stack.)
- Initiate a new search.

Using the links we provide, the user is able to move around the information they have searched for much quicker than they would by moving back and forth through the history stack of the WAP device itself.

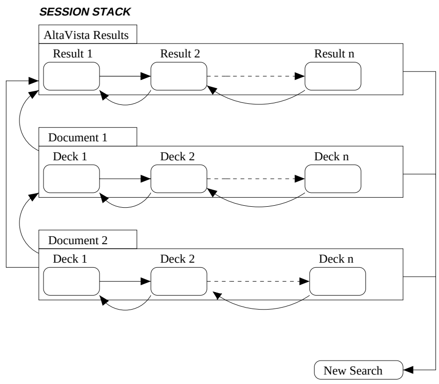

Fig. 4. WapSearch Browsing Functionality.

This kind of browsing is not supported by WML micro-browsers so in order to achieve achieve this functionality, the WAP device must rely on the WAPServlet module to resolve navigation requests. We believe that this technique will lead to significant improvements in system usability.

4 Evaluation

We performed a task-oriented evaluation to find the usefulness of the system as an information retrieval tool. Our aim in this evaluation was to expose some of the strengths and weaknesses of the search interfaces we have created. We followed a task-oriented evaluation so that subjects can be placed in a simulated real-life scenario [Borlund and Ingwersen, 1997,Jose et al., 1998]. We chose this model because traditional IR effectiveness measures such as precision and recall are unsuitable for our system as we currently use the AltaVista search engine to perform our retrieval from the web. Moreover, we are interested in evaluating the usefulness and the user satisfaction with the WapSearch system and hence resorted to a task-oriented evaluation approach [Jose, 1998, chapter 7: pages 129-169]. Our objectives of the evaluation were:

- The effectiveness of the WapSearch interface and browsing functionality
- The role of document summaries in WAP based information retrieval
- The quality of HTML/WML conversion tool

We used WAP emulators in a laboratory setting for the experiment. Our user group was made up exclusively of 18-29 year old individuals. All had experience of Internet search engines and all but one were mobile phone users. These characteristics made them entirely suitable for our evaluation.

4.1 Experimental Design

We employed a related samples, within groups experimental design in which we used nine evaluation subjects, three interface types and three evaluation tasks. The independent variable in the experiment was system type so we asked participants in the evaluation to complete the three task using a different interface. As in [Jose et al., 1998],subjects were assigned with each task system based on a Greco-Latin square design [Tague-Sutcliffe, 1992]. From now on we shall refer to the three system types used in the experiment only as system 'A', system 'B', system 'C'.

Of these interfaces, system 'A' was the most basic, as it did not include document summaries in the search results pages and implemented linear browsing between WML elements. System 'B' also only supported linear browsing but included document summaries in the search result pages. System 'C' included document summaries and supported non-linear browsing.

4.2 Evaluation Procedure

For each subject, the procedure was as follows.

- An introductory orientation session
- Pre Evaluation Questionnaire: This questionnaire was completed by participants before beginning the experiment. It was designed to gather some simple personal details so we would be able to build a profile of our user group if necessary and also their level of familiarity with the kind of technologies they would be using during the experiment
- Performance of the tasks on selected (based on the Greco-Latin square design) system.
- A Post Search questionnaire: On completion of each task, subjects were required to complete questionnaires that allowed them to describe their experiences of using that particular system. This included a set of nine 7 point semantic differentials [Preece, 1994]; three probed user's perception of the task they had been asked to perform, the other six referred to their experiences during the actual search process. Each differential had a label such as 'interesting' or 'easy' and the respondent assigned a score to each term on a scale of 1 to 7. A score of 1 indicated the absolute agreement and a score of 7 indicated absolute disagreement. The task questionnaire also asked the user, using Likert scales, to indicate how confident they were that they had been able to complete their task. A lowest score indicated complete confidence and a maximum score of 5 indicated no confidence at all. Finally the users were asked to indicate, using another scale with range of 1-5, how relevant they thought the results of the search had been.

– A Post-evaluation Questionnaire: Once the participant had completed all three tasks and completed all of the task questionnaires, they were required to complete a final questionnaire that asked them to rank the interfaces they had used in order of preference and write down some general notes. Users were also asked to indicate with a yes or no answer whether or not they thought the application they had been using would be a useful service for mobile devices.

In addition to this, we have used automatic logging to capture additional data, like the number of links followed, number of query iterations and the time taken.

4.3 Results & Analysis

We compared systems A and B to test the hypothesis that the inclusion of search system provided summaries in the results page improve the user performance. We compared systems B and C to test the hypothesis that inclusion of our non-linear browsing approach improves the usability of the system.

We compared the sum of scores for the all the differentials dealing with the search process for all systems. On average, respondents experienced marginally higher levels of satisfaction when using system 'A' than they did using system 'B'. Similarly, system 'C' seems to give a more acceptable user experience than system 'B'. However, a Mann-Whitney test statistic give no significance at the 99.95% confidence level.

We also measured the level of confidence participants felt in their ability to satisfy their information needs using different interfaces. This is done using Likert scale techniques. On the average, respondents displayed highest level of confidence when using system 'C' and lowest levels when using system 'A'. This indicates that both of our hypothesis holds true.

At the end of the experiment subjects were asked to specify the system they liked most. Except for one, all the users preferred system C, one with our non-linear browsing. This supported our hypothesis 2. From the logged data, it was noted that subjects using System C tended to reformulate the queries more often suggesting a higher level of interaction.

The written comments provided by the participants in the experiment gave us some extremely useful feedback about features that were common to all three WapSearch interfaces. By far the most common complaint was the length of time it took to find the information that was deemed to be relevant. Most users became impatient while they waited for the HTML/WML conversion process to complete. Future work on the system should concentrate initially on speeding this process up.

With regards to the inclusion of document summaries, in the search results pages, the user group was divided in two camps. Some seemed to use and appreciate information while others find them to be positively misleading and annoying. We were using the AltaVista provided document summaries, which were later proved to be ineffective [White et al., 2001].

The other most commonly written comment was that the system often appeared to do nothing when the user selected a link. This is not an interface issue, rather, it is a problem with the HTML/WML conversion tool. Nonetheless, undoubtedly, improvements in this aspect would lead to an improvement in the system performance.

A handful of users commented that, the interfaces would have a more intuitive feel if the 'back' link was provided at the top of the page, rather than at the foot alongside the 'more' link. This is a relatively trivial improvement to make.

Also it is noticed that, many users became confused when they returned to the results page of a search they had run from a converted HTML document. This was because when they selected the 'search results' link, they were presented with the first search result for the current query. If the user had selected, say, the fourth or fifth search result, they expected to be returned to this position in the search results from any subsequently viewed documents. This improvement can be made with minimal alteration to the code in the WAPServlet that handles intra-document links.

Finally 6 of 9 participants indicated in the last question in the evaluation task booklet that they thought the WapSearch system would be a useful service for users of mobile devices.

4.4 Future Works

We have also given a great deal of thought to how WapSearch could be adapted in the future to incorporate some of the more exciting developments in mobile Internet technology. A number of extremely interesting specifications for wireless technology have been created in recent years. We are particularly interested in those which will enhance support for context sensitive application development. For instance, Global Positioning System (GPS) when implemented for WAP phones and PDAs will create many opportunities for location Dependant activities. Although the first version of our system employs the AltaVista search engine, it could be adapted to wrap around virtually any HTML search facility. Organisations such as Google already allow users to search uniquely within their own country. Location data retrieved from the client in HTTP headers could be resolved by a variation of the WAPServlet and used to select a geographically appropriate information service with potential improvements in effectiveness.

Another interesting specification is the UAProf syntax developed by the WAP Forum[2]. This technology will allow a client device to inform a server what its hardware and software characteristics are. This would allow developers to optimise application output for specific devices with potentially massive improvements in usability. We have structured the WapSearch code in such a way that it will be possible to integrate these features in to our application when they are implemented by device manufacturers. For instance, the converter module already takes parameters describing the client device's memory and screen sizes and uses them to alter the size and format of the WML code it produces. When

[2] http://www.wapforum.org

dynamic user profiling technology becomes available, minimal alterations to the WAPServlet code could result in massive improvements in system usability.

5 Conclusion

We have introduced a novel approach to creating a valuable information retrieval mechanism for mobile Internet users. The system has been evaluated in a task-oriented user-centred setting and the strengths and weaknesses of the system have been brought out. Experiments have shown the usefulness of our non-linear browsing approach. We have identified areas that need to be improved. In addition, we have discussed a number of future technological developments that can be incorporated into the WapSearch system. The design we have employed means that the WapSearch system is flexible and provides a framework in which some really interesting and valuable mobile information retrieval services can be developed.

Acknowledgements

We would like to give special thanks to Andrew Ritchie, who developed the original version of our HTML/WML conversion tool, and Ryen White for providing a parser for the AltaVista search engine.

References

[Arehart et al., 2000] Arehart, C., Chidambaram, N., Guruprasad, S., Homer, A., Howell, R., Kasippillai, S., Machin, R., Myers, T., Nakhimovsky, A., Passani, L., Pedley, C., Taylor, R., and Toschi, M. (2000). *Professional WAP*. Wrox Press Ltd.

[Borlund and Ingwersen, 1997] Borlund, P. and Ingwersen, P. (1997). The development of a method for the evaluation of interactive information retrieval systems. *Journal of Documentation*, 53(3):225–250.

[Jose, 1998] Jose, J. M. (1998). *An Integrated Approach For Multimedia Information Retrieval*. PhD thesis, The Robert Gordon University, Aberdeen.

[Jose et al., 1998] Jose, J. M., Furner, J., and Harper, D. J. (1998). Spatial querying for image retrieval: a user-oriented evaluation. In Croft, B., Moffat, A., Van Rijsbergen, C. J., Wilkinson, R., and Zobel, J., editors, *Proceedings of the Twenty First Annual International SIGIR Conference on Research and Development in Information Retrieval*, pages 232–240. ACM Press.

[Preece, 1994] Preece, J., editor (1994). *Human-Computer Interaction*. Addison-Wesley.

[Ramsy and Nielsen, 2000] Ramsy, M. and Nielsen, J. (2000). *WAP Usability Deja Vu: 1994 All Over Again*. Nielsen Norman Group.

[Tague-Sutcliffe, 1992] Tague-Sutcliffe, J. (1992). The pragmatics of information retrieval experimentation, revisited. *Information Processing & Management*, 28(4):467–490.

[White et al., 2001] White, R., Jose, J. M., and Ruthven, I. (2001). Query-biased web page summarisation: A task-oriented evaluation. In *Proceedings of the Twenty Fourth Annual International SIGIR Conference on Research and Development in Information Retrieval*, pages 412–413. ACM Press.