A USER-ADAPTIVE SMALL-SCREEN OPTIMIZATION APPROACH FOR SEARCHING BOOKS

Enrico Bertini[1], Andrea Calì[2,1], Silvia Gabrielli[1], Tiziana Catarci[1]
{bertini, cali, gabrielli, catarci}@dis.uniroma1.it

[1]Dipartimento di Informatica e Sistemistica
Università di Roma "La Sapienza"
Via Salaria 113 - 00198 Roma, Italy

[2]Free University of Bozen/Bolzano
Faculty of Computer Science
Piazza Domenicani 3 - 39100 Bolzano, Italy

ABSTRACT

In this paper, we describe a user-adaptive system designed to access a repository of books with small screen devices. The purpose is to investigate the idea of employing usage data as a source for user interface adaptations, to overcome the lack of screen space and limit the need for explicit user input. We present an overview of potential adaptations, describe the user model and the design of an application for an handheheld device, and report on a preliminary expert-based usability study.

1. INTRODUCTION

Searching is one of the typical means of finding specific information from digital libraries and from the Web in general. It is also interesting to observe that searching is one of the tasks that are suitable for mobile devices. While the foregoing is an indicator of an opening for using mobile devices for searching in digital libraries, the limited input and output capabilities of such devices can make the task very time-consuming. While it is true that a lot has been done towards this direction, there still exist great opportunities for minimizing input and interaction activities, and also for optimizing the small screen space by primarily relying on usage data/patterns. In this paper we propose and describe an approach for adapting the content of Web pages to mobile device users while searching for books. In the adaptation, the users are first presented the information that they are likely to need. The adaptation is primarily based on usage data, i.e., data about how each user fills in forms and interacts with the data.

2. ADAPTATION FRAMEWORK

In order to characterize input sources for collecting usage data and potential adaptations, we refer to a common pattern. It has four stages describing the main actions carried out while searching and the corresponding information provided: (1) *input*, where users provide search parameters; (2) *results*, where the system returns a list of items that most likely corresponds to the parameters selected; (3) *details on demand*, where users browse/scroll the list and either, select one item in the list to obtain more details on it, or step back to the search form to refine the query; (4) *further actions*, where users, from the detail page, select new links, activate procedures, make selections. Taking into account this pattern, we can identify some sources for usage data and envision potential adaptations.

Sources of usage data. We consider four types of user input, namely: requests for sorting, form input, further actions, explicit requests for adaptation.

- Requests for sorting: the user sorts the result according to some criteria (e.g., order books by price).
- Form input: the user inputs parameters in the form to guide the search.
- Further actions: the user is in the details page and select further actions to request services (e.g., reserve the hotel), ask for related objects (e.g., books by the same author), activate procedures (e.g., store this in my wish list).
- Explicit requests for adaptation: the user selects some parameters to adapt the user interface explicitly. For instance, s/he requests the system to display results without including images so that screen space is saved and more items can be seen in one screen.

Adaptations. By observing this data we can enable some adaptations aimed at either: reducing the amount of screen space occupied, reducing the amount of explicit input, reducing navigation time. The envisioned adaptations are the following.

- Push-forward/pull-back: User interface objects can be moved up or down in the navigation hierarchy according to usage frequencies. Those that are rarely used can be pushed to a one step forward page, so that screen space can be saved, while those frequently used can be pulled back to allow early access. For instance, rarely used form elements in the input page can be pushed forward to an advanced search page and substituted by a reference or, in the details page, the link to a service which is frequently used, can be moved one step back to the result page.

Figure 1 – The steps required to issue an adaptive search and adaptation refinements.

- Defaulting: In the input page the system can provide defaults for frequently used form elements, automatically selecting parameters and filling text fields, the same way some standard browsers do. When users often access the same search functionality, it is very likely that some, if not all, the specified values are the same. Since typing does not suit very well with small keyboards and styluses, this can save time and effort.
- Filtering: Information considered not relevant for the specific user or for the current task can be filtered. For instance, for each item of the result list, the system can decide to display only a subset of the available attributes (e.g. picture, title, description) thus saving some screen space.
- Sorting: In the result list the results can be ordered according to some criteria, e.g., according to one attribute (e.g. price of a book) if frequently selected. Similarly, in the detail page, an image can be moved in the upper part of the page, if there is evidence that it is one of the main features the user wants to look first.
- Level of detail: Objects can be represented at different levels of detail. In the detail page, an image can be presented at different resolutions and size, a textual description can be presented at different levels of detail (e.g. the entire text or a summary), a link can be presented entirely or with a small icon.

The product of usage data and adaptations can provide potentially useful mechanisms. The point is in finding, for a given adaptation, the input source that best fits its working. For instance, defaulting can be run according to form input. As the system collects the history of selections and text input in forms, it can suggest automatic filling of fields in future sessions, so that explicit input is saved. Similarly, request for sorting the result can be used in future sessions to automatically sort further request. Here we focus on another combination on which our prototype is based, that is, *explicit requests for adaptation* and *filtering*.

The prototype is a web based mobile access to the Amazon.com book search facility, and it is organized around the pattern described above. When the user accesses the web site, s/he is presented with an input form to search books by keyword and an option to ask for an adapted result list (Figure 1a). If the adapted result option is selected, the system chooses a set of item attributes to display and uses it to display the results (Figure 1c). Then, the user can either move on a detail page, request a new search, or refine the current one and, in the latter case, s/he can ask a different set of features to be displayed (Figure 1b). The user model originates from the input collected during these operations and in particular on their frequencies. Each time the user issues a search the system keeps track of the selected combinations and updates a data structure that maintains this information.

The user model is based on the frequency of selection of item attributes. As the user selects them, the system updates a usage model taking into account various aspects of the user behavior. For each attribute, we compute the following data:

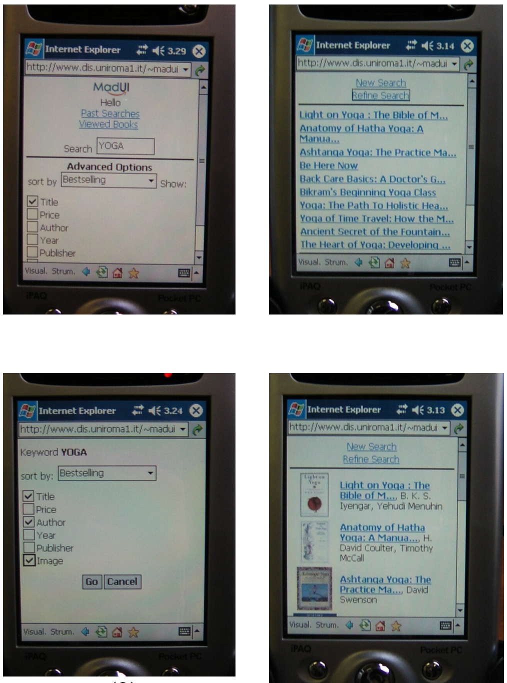
- **Total Frequency (TF)** - the total count of selections from the first time the system has been used.
- **Recent Frequency (RF)** - the count of selections made in the lapse of time going from the last time the service has been used to the last K sessions, where K is a system parameter.
- **Session Frequency (SF)** - the count of selections made in the running session.

These values allow for different adaptations in different moments. When the user accesses the system for a new session, it returns an adaptation that is based on the entire history of selections. When requesting adaptations within the current session, it gives more importance to selections made within it. In the former case, a linear function of total frequency and recent frequency is used, so that attributes steadily recurring over time are suggested. In the latter case, a linear function involving also the session frequency is used. This allows to take into account the case when a user departs from its standard behavior just for the period of a single session.

In particular, at each form and at each result presented to the user, the system chooses the attributes having the highest value calculated on the linear functions of the frequencies described above. The functions are aimed to predict the proper relevance of different attributes within a session or at the beginning of a new one, so as to present to the user the attributes he is likely to want. Of course, the user can still explicitly specify attributes s/he desires.

3. EXAMPLE

In Figure 2, we exemplify a typical interaction with our prototype, where the user goes from a basic search to an explicit request for adaptation, that the system fulfills by providing a filtered set of data in the results page.



(3)

Figure 2 - Typical search session with attribute selection and interface adaptation.

In the first screenshot, the user has inserted a keyword corresponding to her/his information need into the search field. The lower part of the screen presents a combination of data features to be displayed in the result list. These are automatically suggested by the system according to past interactions, as described above. When the user issues the query, a results page is returned where each book is described only by its title, according to the previous request (screenshot 2). If the details provided are not enough, the user can decide to refine the previous search explicitly, by asking to display more details for each item. Screenshot 3 shows how the prototype allows an easy adaptation by enabling the user to select book attributes to be displayed. The user selects: Title, Author, and Image and Screenshot 4 shows the result obtained when issuing the refined request. More details are displayed for each item and, consequently, a reduced number of books can be seen in one single screen. The purpose of the system is, indeed, to find the right balance between level of detail and screen optimization. As the system learns what is the preferred combination of attributes for the specific user, it is likely that the right one is selected according to her/his specific needs.

4. EVALUATION

Past evaluations of adaptive systems pinpoint the risk for the users to get disoriented and confused by a system's responses, leading to usability problems such as: experience of a lack of control during the interaction, unpredictability of system's behaviour, misunderstanding of the rules governing the adaptation mechanism. Studies reported in [1] [2] have shown how the usability criteria of *controllability*, *predictability* and *transparency* should be applied throughout the whole design cycle of adaptive systems. Although it is common to evaluate the benefits of adaptive interfaces by comparing their performance to those of similar non adaptive versions of a same system, this approach might not be particularly useful or appropriate during early stages of design, when developers need to explore and capitalize more on usage patterns to understand how to progressively refine valuable adaptive features. Expert-based evaluations are good candidates and cheap techniques to apply in these cases. They have also proved to be adequate methods for assessing the correctness of input data acquisition for user modelling [4]. During our prototyping cycle we have started by applying Heuristic Evaluation [3] to collect and feed usability data into the experimentation and refinement of the adaptation features derived from our framework. Three usability experts were involved in using the prototype to carry out a number of information seeking tasks, going from simple book searches to complex ones. Evaluators' reports contained an in-depth analysis of usability flaws and a detailed description of how much the adaptation features already implemented were compliant with the usability criteria mentioned above. Generally, experts agreed in finding the enabled adaptation features of parameters selection-filtering and results' sorting particularly effective in allowing user control of interaction and rapid access to relevant information during both simple and more complex searches. Minor refinements of user-system interaction were suggested by evaluators on these features. By contrast, experts found that the design of an optional input mode made available to the user, i.e. the reuse of past searches as a way of easing and speeding up the retrieval and display of relevant information results, was not fully compliant with the criteria of predictability and transparency. Suggestions were given to designers on how to improve this feature to make it more usable and easily understandable by users. Current work is leading to the release of a more complete and adaptive version of our prototype, on which we are planning to conduct extensive user studies, both within and between sessions, to evaluate the benefits entailed by the adaptation framework we have described.

5. RELATED WORK

The problem of optimizing screen space has gained much interest in recent years. Brewster in [5] investigates the idea of overcoming the lack of screen space by using non-speech sound, so that the limited visual feedback is compensated by audio feedback. The RSVP Browser [6] deals with the lack of screen space using time as an additional resource. The idea is that what cannot be seen in a single screen can, at least, be seen in a short time by sequentially displaying in a rapid visualization (less than 1 second) the various segments that would not fit in a single screen. Exploiting the idea of page summarization, Power Browser [7] presents the content of standard web pages by displaying links descriptors instead of the entire content and providing a hierarchical list of links extracted from current and subsequent pages. As to adaptive systems specifically designed for small screen devices, Toogle [8] is a interesting system. It tries to increase the accuracy of items retrieved in Google searches by monitoring the selections of result items performed in the current session. Equally interesting, Smith presents in [9] an adaptive web portal which monitors the user's selection of menu items of a WAP service to promote frequently used menu items so that the user can rapidly access them in early stages of the navigation space. The solution provided in this paper is similar in that the user model is constructed monitoring usage data. The difference, however, is that the main purpose of our prototype is to efficiently use the available screen space, more than increasing result accuracy or reducing navigation.

Acknowledgments. This research was supported by DELOS network of excellence on Digital Libraries (http://www.delos.info), and by project MIUR-FIRB-MAIS: Multichannel Adaptive Information Systems (http://www.mais-project.it).

6. REFERENCES

[1] K. Höök, "Evaluating the Utility and Usability of an Adaptive Hypermedia System", *Proceedings of 1997 International Conference on Intelligent User Interfaces*, ACM, Orlando, Florida, pp. 179-186, 1997.

[2] K. Höök, "Steps to take before IUIs become real", *Journal of Interacting with Computers,* vol. 12, no. 4, pp. 409 426, February 2000.

[3] J. Nielsen, and R.L. Mack, "*Usability Inspection Methods*", New York: John Wiley & Sons, 1994.

[4] S. Weibelzahl, and C.U. Lauer, "*Framework for the evaluation of adaptive CBR-systems*", in I. Vollrath, S. Schmitt and U. Reimer (Eds.), Experience Management as Reuse of Knowledge. Proceedings of the 9th German Workshop on Case Based Reasoning, Baden-Baden,Germany, pp. 254-263, 2001.

[5] S. Brewster. *Overcoming the lack of screen space on mobile computers.* Personal Ubiquitous Comput., 6(3):188–205, 2002.

[6] O. de Bruijn, R. Spence, and M. Y. Chong. Rsvp browser: *Web browsing on small screen devices.* Personal Ubiquitous Comput., 6(4):245–252, 2002.

[7] O. Buyukkokten, H. Garcia-Molina, A. Paepcke, and Terry Winograd. *Power browser: efficient web browsing for pdas.* In Proc. of the SIGCHI conference 2000, pages 430–437. ACM Press, 2000.

[8] J. D. Ruvini. *Adapting to the user's internet search strategy on small devices.* In Proceedings of the 8th international conference on Intelligent user interfaces, pages 284–286. ACM Press, 2003.

[9] B. Smyth and P. Cotter. *The plight of the navigator: Solving the navigation problem for wireless portals.* In Proc. of the 2nd International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, pages 328–337. Springer-Verlag, 2002.