



Provided by the author(s) and University of Galway in accordance with publisher policies. Please cite the published version when available.

Title	DigiMe - Ubiquitous Search and Browsing for Digital Libraries
Author(s)	Kruk, Sebastian Ryszard; Grzonkowski, Slawomir; Gzella, Adam; Cygan, Mariusz
Publication Date	2006
Publication Information	Sebastian Ryszard Kruk, Slawomir Grzonkowski, Adam Gzella, Mariusz Cygan "DigiMe - Ubiquitous Search and Browsing for Digital Libraries", Proceedings of the MoSO2006 Workshop, in conjunction with MDM 2006, 2006.
Publisher	IEEE
Item record	http://hdl.handle.net/10379/489

Downloaded 2024-03-20T08:38:58Z

Some rights reserved. For more information, please see the item record link above.



DigiMe - Ubiquitous Search and Browsing for Digital Libraries

Sebastian Ryszard Kruk*
sebastian.kruk@deri.org

Sławomir Grzonkowski^{†,*}
slawomir.grzonkowski@deri.org

Adam Gzella^{†,*}
adam.gzella@deri.org

Mariusz Cygan^{†,*}
mariusz.cygan@deri.org

* Digital Enterprise Research Institute, NUI Galway
IDA Business Park, Galway, Ireland

[†] WETI, Gdansk University of Technology
Narutowicza 11/12, Gdansk, Poland

Abstract

Vast sources of information, such as the Internet are difficult to browse and/or search through. Existing tools can be sometimes frustrating for many people. So far a number of techniques have been proposed to deliver user-oriented solutions. The problem re-surfaces however within a ubiquitous computing paradigm. Aside from a possible delay in response time, there are additional drawbacks with respect to mobile devices such as: Bandwidth size, storage(what type of costs do you mean) and performance costs and client UI size.

In this paper we introduce three search and browsing features: fulltext search, collaborative filtering and multifaceted browsing, all of which can be enriched with semantic and community information. We present two of the aforementioned techniques implemented in our interface for digital libraries - DigiMe.

1 Introduction

Existing digital libraries readers carry their personal computers, even when they only want to perform a quick search for resources or manage their bookmarks. Situations like this occur very often, especially during conferences, when we meet new interesting people or when we come across new resources that might be helpful in our work. Unfortunately, due to its size, even a laptop is too large to carry everywhere. Moreover, the battery can only work for a few hours and often there is limited access to the electrical mains. Finally, very often the user wishes only to perform

a simple task such as browsing through their favorite digital library, sending an email or text message.

The mobile system described in this article is called DigiMe, utilizes JeromeDL - a digital library with semantics which have been combined with FOAFRealm. FOAF-Realm is a distributed users profile management. Both systems take advantage of Service Oriented Architecture (SOA). Our primary aim was to develop a mobile application that could work with both PDA's and mobile phones. The goal of our application was to provide users with essential library functionality such as searching for resources, managing friends or bookmarks. This paper describes a mobile system that solves some of mobility problems stated above. DigiMe combines several technologies from the Semantic Web and Service Oriented Architecture. This mobile application allows users to acquire ubiquitous access to the search and browsing features of the JeromeDL digital library.

Furthermore, our paper makes the following contribution to mobile computing:

- We show how to cope with semantic information when we have limited computing power.
- We describe an open-source DigiMe implementation.
- We present how to apply the ubiquitous paradigm to a real system.
- We present an architecture for a mobile application that takes advantage of both Service Oriented Architecture and the Semantic Web.

In the next section we present a model for the information management lifecycle implemented in the DigiMe

Project. In section 3 we define challenges and limitations of the ubiquitous search and browsing paradigm. In section 4 we elaborate on the DigiMe prototype implementation. And finally in section 5 we present an evaluation of our approach, comparing it to the state of the art (see section 6) and concluding in section 7 with information on future work within the DigiMe project.

2 Information Management Lifecycle

Information Retrieval studies has occupied many researchers over the past decades. Google is an revolutionized the way we use Internet. This search engine applies state of the art information retrieval techniques. Google makes one entry point for many people who are trying to find different kinds of information. However, we are still aware that to answer user queries correctly, we need something more. The Semantic Web aims to enable users to understand the meaning of the information being processed. On-line communities bring experts and novices close enough together to ease the flow of knowledge.

One of the crucial questions that has to be answered when building an information acquisition and management system is not *how* but *why* [8]. Rose and Levinson [22] classified user goals into three categories:

- resource finding,
- navigational,
- informational.

These three categories adhere to various stages of information acquisition and management. A user will look for a specific resource, and navigates through a large set of results, to acquire precise information about his/her acquaintances and friends. The Information Management lifecycle can be represented by three actions: browse information, result set manipulation and collaboration on cached information. These actions represent different categories within user goals in the search and browsing process.

2.1 Browse Information - Semantically Enhanced Search

The query refinement is one of the main features in modern IR systems. Based on a keyword disambiguation, query refinement narrows/expands the set of results. It also introduces some more thorough ranking of the set of results. The categorisation taxonomies maintained by communities (see Sec. 4) provide semantically rich information that can be utilised during query expansion and disambiguation process [19].

2.2 Manipulate Result Set - Multifaceted Browsing

The faceted navigation [23] is based on the idea of narrowing the set of resources by choosing more specific features from given taxonomies. Multifaceted browsing (MFB) allows to construct filter queries based on values selected from taxonomies. These taxonomies define orthogonal characteristics of resources. Each characteristic is represented by a facet that describes elements from the set of resource which the user manipulates. MFB delivers a list of facets that represents orthogonal taxonomies of attributes assigned to resources.

2.3 Collaborative Information Sharing - Social Semantic Collaborative Filtering

Social semantic collaborative filtering (SSCF) [18] is a type of collaborative filtering [14] that makes use of existing, user maintained social networks and semantic annotations.

SSCF is now in a mature state of development and performs well on rich client machines. A user can fetch all bookmarks and directories and browse these resources offline. There were a lot of obstacles of mobile devices which we faced when constructing DigiMe. The amount of RAM and non-volatile memory is limited, processor speed is slow, and the content appears on a small display. Plus, the network connection is quite slow and usually expensive (e.g. price for GPRS connection). The limitations stated above caused the development of SSCF for mobile devices to become a non-trivial task.

3 Towards Ubiquitous Search and Browsing

According to the research on Information Management lifecycle presented in the previous section, the process of search and browsing can be represented by the following three user's actions:

Step 1 The user is searching for information using a keyword query. Either he/she has a general idea on a possible content or he/she is looking for a precise information that he/she has already read about but has forgotten the location of the resource.

Step 2 Once the preliminary set of results is delivered, the user refines the query constraints and browses through the set of results.

Step 3 The user bookmarks an interesting resource for further reference.

These three steps must be delivered for the mobile platform to provide ubiquitous paradigm in information management, in digital libraries or other information systems.

Next section presents possible challenges and limitations that can occur while developing ubiquitous information management paradigm.

3.1 Challenges and Limitations

The most important questions are: How much data should the application fetch from server at a time? and how much data a mobile device application can store on the mobile device? If we retrieve only the current level of tree nodes the user will suffer huge delays (after making a new request). On the other hand application, the cannot retrieve too much data because memory in mobile devices is relatively small. A User incurs a performance loss, for the data sent to and from his/her mobile device. Therefore, downloading all resources at only once, to get one particular bookmark, would cost too much. It is difficult to find a solution which would be cheap and fast. DigiMe buffers resources selected by the user, and fetches new resources on demand. In this solution the user has to wait to get the next level of bookmarks within the taxonomy, but he/she only gets what he/she asked for. Plus, user does not need to wait for the data when he/she wants to see the information which has been already downloaded. However, the size of the buffer is not unlimited. It depends on the specific characteristics of the mobile device. Once the buffer is full, the least frequently accessed information is removed.

Another issue is how long to keep the buffered information. We decided to store it only for one session. In the current state of development, keeping this data for more than one session would cause almost the same network load as a user who is frequently re-using the same resources. Such a solution can be irritating. Therefore, DigiMe provides a new category of bookmarks and directories, which are stored in non-volatile memory. The user can select resources which should always be available. He/she can also instruct the application to update information stored in the mobile device. Pool of resources is always available for user, independent of network connection.

Screen dimensions and user input devices are another set of problems. Due to this restriction user interface has to be precise and accurate, while also providing a set of dedicated commands. DigiMe displays only one logical level of information at a time. The user can see more detailed information or the next level of within the bookmarks tree upon selecting the appropriate resource.

4 DigiMe - Prototype Implementation

There are three modules in the DigiMe implementation (see Fig. 1):

- The main module communicates with the user and connects to the network. It contains the most important and indispensable application functions. Dedicated searching interface and mobile bookmarking modules are built on top of the main module.
- Searching interface interacts with instances of JeromeDL.
- Mobile bookmarking facilitates profile information delivered by FOAFRealm.

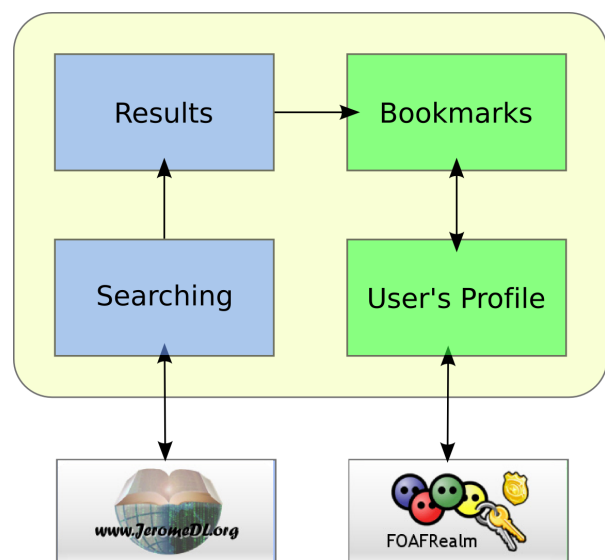


Figure 1. Conceptual diagram of DigiMe

4.1 The Main Module - A User Profile

The main module forms the basis of the application. It is needed for other modules to work properly. It allows a user to authenticate himself/herself using FOAFRealm credentials. After successful authentication (see Fig 2(a)) user can change his/her profile information (name, surname, nickname, phone, etc; see Fig 2(b)) and browse through his/her list of friends (see Fig 2(c)). A user can:

- add new friends,
- remove existing ones,
- change friendship levels (see Fig 2(d)),
- browse friends' bookmarks,

- and send a message (e-mail or SMS) to a selected friend.

Some of DigiMe's functions include: profile management and mobile bookmarking (see sec 4.3). They are based on services available in FOAFRealm (SOA layer). FOAF-Realm features are exposed through the Web Services Standard using Apache Axis platform. Axis builds SOAP messages in RPC style encoded Native J2ME SOAP implementation (JSR172 [4]), which can work with document/literal style only. That is why we were forced to employ kSOAP - an alternative SOAP library for J2ME, which co-operates well with RPC/encoded messages. The searching interface (next application module) uses simple servlet invoking and does not need kSOAP or any other Web Services libraries (for more details see Sec 4.2).

The SOA layer does not only expose FOAFRealm methods to the external application, but also provides secure access to these operations. A special session identifier is requested to invoke any method. External application receives the information after successful login. In the next version of SOA for FOAFRealm, the session identifier will be more secure based on time-out mechanisms. It will be possible to invoke Web Services using a secure (SSL) connection.

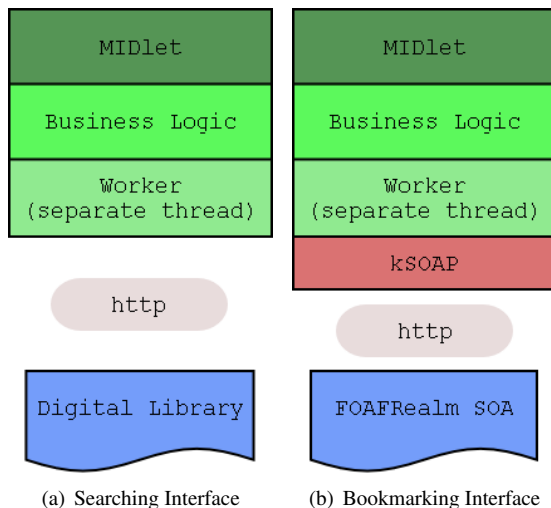


Figure 3. Architecture of the DigiMe as a mobile application

DigiMe consists of a standard MIDlet class, worker class and a collection of business classes (see Fig 3). The MIDlet class is responsible for the communication with user. It builds screens and shows appropriate information. All the network communication is done by the Worker class. It utilizes kSOAP functions to invoke FOAFRealm Web Services. Operations are performed in separate, independent threads and results received from services are passed to the

MIDlet class. Business classes handle operations and store information about friends, user identity and messages (e-mail and SMS).

4.2 Dedicated Search Interface

The search interface module enables searching on JeromeDL digital libraries. It allows users to look for specific resources anywhere, anytime and anyplace. A user can choose the library from a list of available JeromeDL instances or add a new one (see Fig 4(a)). The application checks automatically if the server is on-line before performing any operations on it. Finally the user can also initiate a distributed search over an L2L (HyperCuP[1]) network of JeromeDL instances.

The searching interface module attempts to offer the same functionality as the classic web based interface for JeromeDL. A user can perform a simple search, advanced search (see Fig 4(b)) or even semantically enhanced search. After obtaining the search results (see Fig 4(c)), the user can request additional information about the resources e.g. its abstract (see Fig 4(d)). Using mobile bookmarking module described later (see 4.3), user can add resources found during the search process to his/her bookmarks.

One of the most important differences between mobile and desktop applications, is the size of the information which can be shown to the user. Consequently, the first screen after initiating a mobile search module contains only one search field for entering a search query. Semantically enhanced search in JeromeDL attempts to find results accurately. Therefore, it decreases the cost of communication with the JeromeDL server.

To keep communication time as short as possible, additional data is sent only when requested. Currently, JeromeDL can handle five types of resources: PDF, RTF, document scans, HTML and URL links. Some of these types are almost useless with respect to mobile devices, e.g. there is rather no need to send a big image file to old mobile devices or HTML to mobile device without a web browser. However, user may need to get a context of resources for reading. A searching interface module allows a user to choose in which format the content should be sent, depending on user's requirements and the abilities of his/her device.

As searching does not require invoking any special methods on server side or sending large amounts of data, there is no need to use Web Services. All operations are done simply by calling one of JeromeDL servlets (incl. Open Search Servlet). It is done directly by the Worker class using a HTTP request, without the kSOAP tier (see Fig 3(b)). After the request has been sent, the Worker class waits for the response in separate thread. Finally, upon arrival the response is parsed and presented to the user.

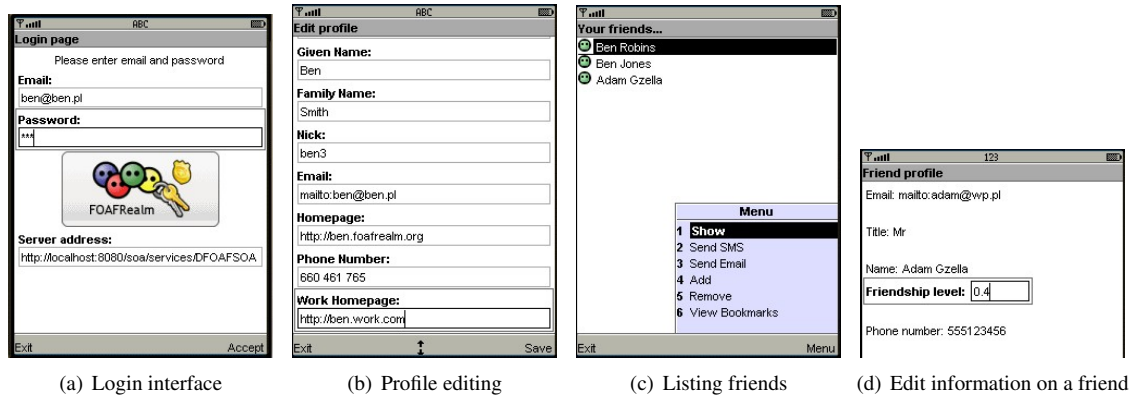


Figure 2. DigiMe Profile Interfaces

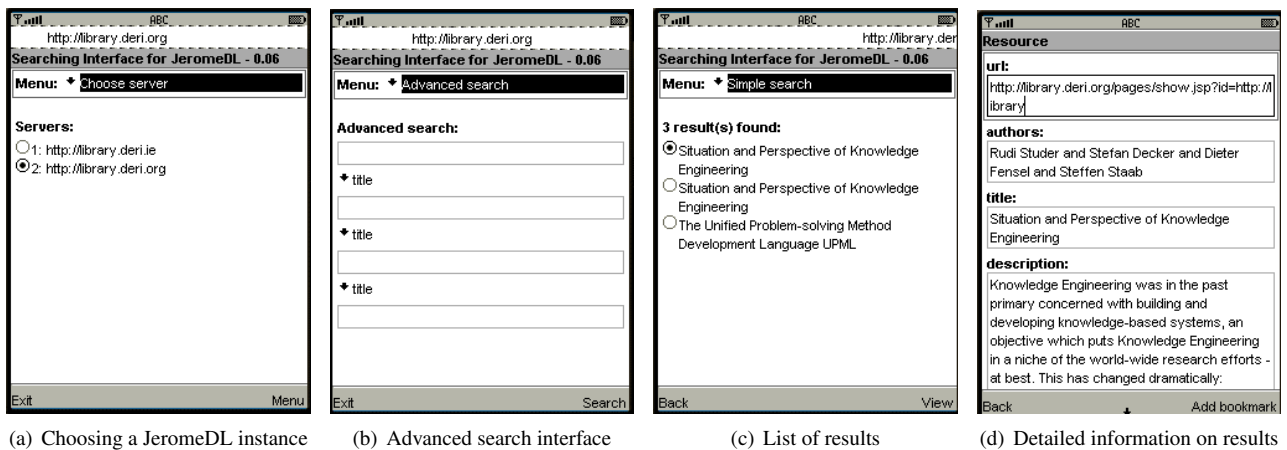


Figure 4. DigiMe Search Interface

4.3 Mobile Bookmarking

The mobile bookmarking module enables a user to browse his/her bookmarks (see Fig 5(a)). He/she can also add new information, remove and/or edit existing bookmarks (see Fig 5(b)). This module allows the user to browse his/her friend's bookmarks.

The DigiMe bookmarking module utilizes the Worker class to invoke special services created in SOA layer. Directories and bookmarks in DigiMe create private taxonomy structures. Each directory might contain both bookmarks and other directories. In DigiMe, the user can browse his/her bookmarks tree. However, a different situation occurs when he/she wants to view resources owned by his/her friends. The application displays bookmarks or directories only when the user accessing the information has sufficient access rights to the resource. Access rights are composed of the maximum distance between user or/and the minimum level of trust/friendship relation [17]. A DigiMe user can apply access policies to each directory in the bookmarks

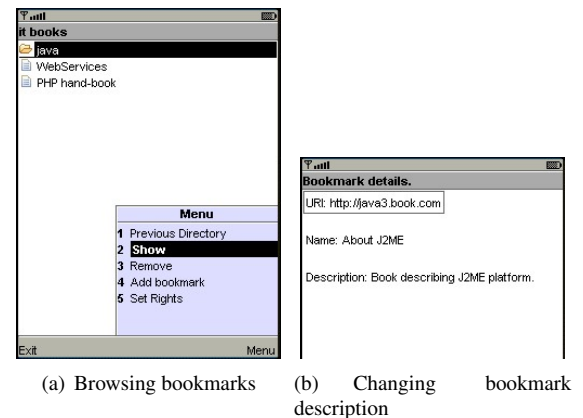


Figure 5. DigiMe Bookmarking Interface

structure and hence ensure access control to all resources and sub-directories.

5 Evaluation

The DigiMe system was initially developed as two separated modules. The search interface allows one to browse library resources and manage bookmarks. The profile management module allows a user to edit profile information his/her including list of friends and bookmarks. Both modules do not depend each other. Moreover, they implement different communication standards, and thus we can choose the more suitable solution in the final version of application.

Code size The size firmly depends on communication standard (see Table 1 and Fig 6). In the searching interface which is a web services based application (see Fig 6) only 4.01% of code is related to communication. The kSOAP library was an necessary element in DigiMe, because the server of the application utilizes Axis. The standard Sun's Wireless Toolkit(WTK) [6] provides only JSR 172 [4] specification support that is incompatible [5] with many web services servers (including Axis). Therefore, 37.38% of code is dedicated to communication.

We have compared the Java Servlet approach to SOA and observed that only 21,95% of code is dedicated to communication in the latter one. In summary (see Table 1), the size of final application does not exceeds 52 kB + communication and graphics parts. If we decide to use Java Servlets, the total size should be about 64 kB (52 kB + 22%) + graphics. Currently, we have two applications that require about 123 kB of memory.

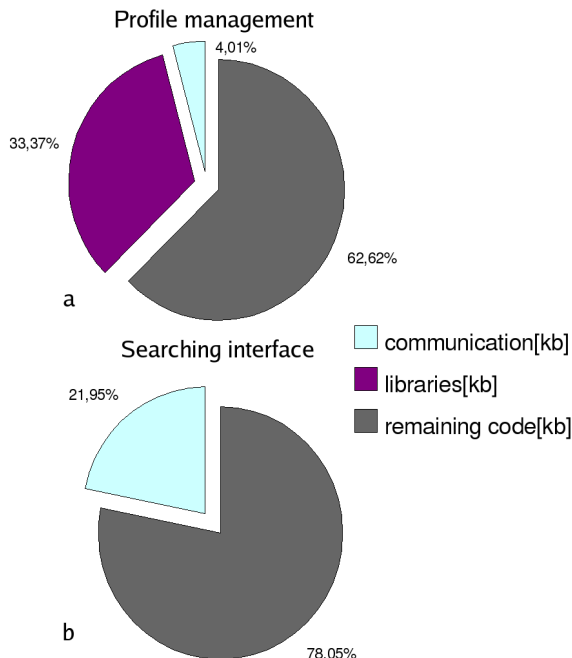


Figure 6. Code in JAR distribution

Table 1. JAR content size distribution

Characteristic	Searching	Profile
total size[kB]	48,09	34,55
graphics[kB]	34,55	8,65
communication[kB]	2,97	2,65
libraries[kB]	0	22,06
remaining code[kB]	10,56	41,39

Table 2. Communication costs of search interface

Characteristic	DigiMe [kB]	Opera [kB]
loading first page/ setting up new JeromeDL server	<i>in:</i> 12,60 <i>out:</i> 0,54	<i>in:</i> 20,3 <i>out:</i> 3,5
searching for 'Decker' in search.jeromedl.org	<i>in:</i> 20,7 <i>out:</i> 0,6	<i>in:</i> 22,3 <i>out:</i> 3,5
viewing description	<i>in:</i> 0,0 <i>out:</i> 0,0	<i>in:</i> 14,8 <i>out:</i> 3,1

Time and Cost Time Performance (compared to the web version) measures were performed in a real environment (Nokia 3230 + GnuBox [13]). Although there are no requirements with regard to the speed of the application, DigiMe meets the widely adopted accessibility standards¹. The most important issues were correct search results and a stable application, and both aspects were achieved. In order to enable users to operate the system quicker, maximal menu depth equals three. As sizes of response packets in both Java Servlet and Web Services approaches were adjusted to mobile device capabilities, the amount of time for receiving search results is even faster than a traditional Internet browser.

We have also compared (see Table 2) the costs of communication between using DigiMe dedicated search interface and OperaMiniTM to access <http://search.jeromedl.org> (starting point to P2P network of JeromeDL instances). The difference between the size of received packets in both solutions is meaningless. However they are significant when comparing the size of send packets.

6 State of art

Several solutions have been proposed to allow mobile access to Digital Libraries [21, 20]. Some of them deliver

¹<http://www.useit.com/papers/responsetime.html>

interesting features at the cost of portability, while others make high demands of device resources.

Wireless Access Protocol

The prevailing method for making services available to various devices is by making different versions, as in the case of web services for WAP (Wireless Access Protocol) [7]. WAP was designed to provide services for mobile devices equivalent to a Web browser with some mobile-specific additions. As the range of devices expands, engineering a new version for each of them quickly becomes no longer feasible. DigiMe was implemented in J2ME (Java 2 Standard Edition), which is wide-known standard for mobile applications. If the device contains only the Java runtime environment and almost all do, DigiMe will work without any modifications.

Mobile web browsers

Another popular method is to use web interfaces directly. Most up to date mobile devices run a web browser [9]. At the moment, Opera Mobile^{TM2} equipped with Opera's Small-Screen Rendering technology for Symbian OS³ is one of the most popular approaches/methods. However, this solution has its drawbacks. There is still a need to adapt the layout to small screens. The majority of web pages were not designed for rendering on mobile devices and therefore do not take advantage of device-specific features. There is also no control over how user interfaces will be presented to end-users. DigiMe was designed specially for mobile devices. It will detect mobile device capabilities and adapt itself to it automatically.

Other mobile applications

Although there are a few other mobile applications of search and browsing for Digital Libraries, neither of them is as sophisticated as DigiMe. Their functionality is often restricted and the user is forced to combine features of different applications on themselves.

Google Mobile Google Mobile Search was specially designed to port the Google search engine to mobile devices. It functions similarly to Google the major difference being the presentation of the search results on a small screen.

Caching Caching in mobile applications does not save only on cost but also on energy and time. Hayat Kara *et al* [15] propose a caching architecture for content delivery

to mobile devices, which is an attempt to provide users with efficient interaction support.

A subscription model for web directories has been proposed by Cohen *et al* [11]. The idea is to allow users to specify the parts of the directory that they are interested in. It has been shown how to synchronize the device and the directory, e.g. the Pocket Directory Browser for Palm.

Another interesting solution has been presented by Elbashir *et al* [12]. This paper describes a transparent approach to the caching of Web Services. In addition, the authors took advantage of a set of semantic tags to improve efficiency.

DigiMe was not only implemented for small devices but was also specially designed for them. A user will be never overloaded with the large amount of search results, as the semantic search provided by JeromeDL returns only precise results to a query. There is simply no point of returning more than 100 search results, because most likely the mobile user will not have time and patience to browse through them all. Returning a small number of very precise results greatly reduces the communication time between device and server.

Vodafone Live Mobile Portal Vodafone Live Mobile Portal beefs up search services by incorporating elements of Semantic Web technology. Semantic Web methods brought more precision to customer searches resulting in earning more money for downloaded content. Vodafone uses RDF to describe content which later is used during the search process on Vodafone Live Web site.

DigiMe goes one step further. It uses RDF descriptions of resources and information from a FOAFRealm user profile. Based on this information, from a user perspective the JeromeDL can return more precise search results.

7 Conclusions and Future Work

In this article we have introduced a novel approach to information management based on three main components: searching, multifaceted browsing and collaborative filtering. We have presented DigiMe, a mobile application that implements two of those components: search interface and collaborative filtering. We have demonstrated on two methods of implementing ubiquitous computing. One exemplified in DigiMe by the searching interface to JeromeDL utilizes Java Servlet connectivity. The latter method an identity management and a bookmark sharing component, is based on the Service Oriented Architecture. Finally, we have evaluated and compared both approaches.

Of the three search and browsing components - multifaceted browsing has not yet been implemented in DigiMe. The preliminary research we have done so far indicated that the mobile application would need to download, store and

²<http://www.opera.com/products/mobile/>

³<http://www.symbian.com/>

manage huge taxonomies for each facet. Alternatively, a lot of service-calls would need to be performed before defining the resulting MFB filter. Since neither of those solutions seemed reasonable we concluded to further continue our research on that part and come up with comparison prototypes soon.

The second goal of the ongoing research within DigiMe project is to provide for more robust identity management. It will be based on the FOAFRealm platform which by then should have realized the concept of Identity 2.0 [2]. DigiMe aims to provide reach-profiling with privacy issues in mind [16, 10].

8 Acknowledgments

The support of Enterprise Ireland, through its Informatics Commercialisation initiative, for the eLITE Industry Led Research Project in eLearning is gratefully acknowledged. This material is also partially based upon works supported by KBN, Poland under grant No. 4T11C00525. Authors would like to thank Brian Davis, Stefan Decker, John Breslin and Andreas Harth for fruitful discussions.

References

- [1] HyperCuP Lightweight Implementation project: <http://www.hypercup.org/>.
- [2] Identity 2.0: <http://www.identity20.com/>.
- [3] D-FOAF: <http://d-foaf.foafrealm.org/>.
- [4] JSR 172: <http://jcp.org/aboutJava/communityprocess/final/jsr172/>.
- [5] <http://www.javaworld.com/javaworld/jw-10-2003/jw-1003-wsstyles.html>.
- [6] WTK: <http://java.sun.com/products/sjwtoolkit/>.
- [7] Arehart.C. *Professional Wap*. 2000.
- [8] M. J. Bates. Information search tactics. *Journal of the American Society for Information Science*, 30(4):205–214, 1979.
- [9] O. Buyukkokten, H. Garcia-Molina, and A. Paepcke. Seeing the Whole in Parts: Text Summarization for Web Browsing on Handheld Devices. In *Proceedings of the Tenth International Conference on World Wide Web (WWW10)*.
- [10] H. Choi, S. R. Kruk, S. Grzonkowski, K. Stankiewicz, B. Davis, and J. G. Breslin. Trust models for community-aware identity management. In *Proceedings of IRW2006*, 2006.
- [11] D. Cohen, M. Herscovici, Y. Petruschka, Y. S. Maarek, A. Soffer, and D. Newbold. Personalized pocket directories for mobile devices. In *WWW2002*. ACM Press, 2002.
- [12] K. Elbashir. Transparent web service caching for mobile devices. In *Proceedings of the 2003-2004 Grad Symposium*, April 2004.
- [13] GnuBox. Share your computer's internet access to the phone via bluetooth or usb. <http://gnubox.dnsalias.org/gnubox/>.
- [14] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry. Using collaborative filtering to weave an information tapestry. *Commun. ACM*, 35(12):61–70, 1992.
- [15] H. Kara and C. Edwards. A caching architecture for content delivery to mobile devices. In *29th Euromicro Conference (EUROMICRO'03)*, page 241, 2003.
- [16] S. R. Kruk. Digime - personalization vs privacy. <http://skruk.blogspot.com/2006/02/digime-personalization-vs-privacy.html>.
- [17] S. R. Kruk. FOAF-Realm - control your friends' access to the resource. In *FOAF Workshop proceedings*, http://www.w3.org/2001/sw/Europe/events/foaf-galway/papers/fp/foaf_realm/, 2004.
- [18] S. R. Kruk and S. Decker. Semantic Social Collaborative Filtering with FOAFRealm. In *Submitted to ISWC'2005*, 2005.
- [19] S. R. Kruk, S. Decker, and L. Zieborak. JeromeDL - Adding Semantic Web Technologies to Digital Libraries. In *DEXA Conference*, 2005.
- [20] C. C. Marshall, G. Golovchinsky, and M. N. Price. Digital Libraries and Mobility.
- [21] C. N. and S. J. A. PoPS: Mobile Access to Digital Library Resources. In *Proceedings of the 2003 Joint Conference on Digital Libraries (JCDL'03)*.
- [22] D. E. Rose and D. Levinson. Understanding user goals in web search. In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, pages 13–19, New York, NY, USA, 2004. ACM Press.
- [23] K.-P. Yee, K. Swearingen, K. Li, and M. Hearst. Faceted Metadata for Image Search and Browsing. In *CHI*, 2003.