Lecture Notes in Computer Science 2954
Edited by G. Goos, J. Hartmanis, and J. van Leeuwen

Springer
Berlin
Heidelberg
New York
Hong Kong
London
Milan
Paris
Tokyo

Fabio Crestani
Mark Dunlop
Stefano Mizzaro (Eds.)

Mobile and Ubiquitous Information Access

Mobile HCI 2003 International Workshop
Udine, Italy, September 8, 2003
Revised and Invited Papers

Springer

Volume Editors

Fabio Crestani
Mark Dunlop
University of Strathclyde
Department of Computer and Information Sciences
Livingstone Tower, 26 Richmond Street, Glasgow G1 1XH, UK
E-mail: {f.crestani;mdd}@cis.strath.ac.uk

Stefano Mizzaro
University of Udine
Department of Mathematics and Computer Science
Via delle Scienze, 206, Loc. Rizzi, 33100 Udine, Italy
E-mail: mizzaro@dimi.uniud.it

Cataloging-in-Publication Data applied for

A catalog record for this book is available from the Library of Congress.

Bibliographic information published by Die Deutsche Bibliothek
Die Deutsche Bibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data is available in the Internet at <http://dnb.ddb.de>.

CR Subject Classification (1998): H.5.2, H.5.3, H.5, H.4, H.3, C.2, I.2.1, D.2, K.8

ISSN 0302-9743
ISBN 3-540-21003-2 Springer-Verlag Berlin Heidelberg New York

Typesetting: Camera-ready by author, data conversion by Olgun Computergrafik
Printed on acid-free paper SPIN: 10985229 06/3142 5 4 3 2 1 0

Preface

The ongoing migration of computing and information access from the desktop and telephone to mobile computing devices such as PDAs, tablet PCs, and next-generation (3G) phones poses critical challenges for research on information access.

Desktop computer users are now used to accessing vast quantities of complex data either directly on their PC or via the Internet – with many services now blurring that distinction. The current state-of-practice of mobile computing devices, be they mobile phones, hand-held computers, or personal digital assistants (PDAs), is very variable. Most mobile phones have no or very limited information storage and very poor Internet access. Furthermore, very few end-users make any, never mind extensive, use of the services that are provided. Hand-held computers, on the other hand, tend to have no wireless network capabilities and tend to be used very much as electronic diaries, with users tending not to go beyond basic diary applications.

This "state-of-practice" presents a dramatic contrast to the technological vision, and the emerging "state-of-the-art" devices, which are small, very powerful, wireless networked computing platforms. Providing access to large quantities of complex data on such devices while users are on the move and/or engaged in other activities poses significant challenges to the information access community and brings together many classical computing domains, such as information retrieval (IR), human-computer interaction (HCI), information visualization, and networking. This volume contains 21 papers that approach these challenges from different directions. The bulk of the papers come from the Workshop on Mobile and Ubiquitous Information Access that was held as part of Mobile HCI 2003 in September 2003.[1] Other papers were specially invited, to complement the presented papers and extend the volume.

Overview

The 21 papers in this volume have been grouped into the following four parts. Many of the papers fall into more than one category, and sometimes our choice has been somewhat arbitrary, but hopefully still useful.

Foundations: Concepts, Models, and Paradigms

The field is young, so it is not a surprise that some work is being done on basic concepts and visions of the future. In *The Concept of Relevance in Mobile and Ubiquitous Information Access*, Coppola et al. discuss the concept of relevance in the mobile, wireless, and ubiquitous information retrieval arena. In *Conversational Design as a Paradigm for User Interaction on Mobile Devices*, Leong borrows from well-established linguistics research and he presents a design paradigm for user interfaces on mobile devices based

[1] Mobile HCI 2003 was part of the Mobile HCI series (see `www.mobilehci.org`); its proceedings were published in LNCS volume number 2795.

on Grice's conversational implicatures. *One-Handed Use as a Design Driver*: *Enabling Efficient Multi-channel Delivery of Mobile Applications*, by Nikkanen, presents several practical and useful guidelines for mobile devices and applications, based on both a literature review and lessons learned at Nokia. In the last paper in this part, *Enabling Communities in Physical and Logical Context Areas as Added Value of Mobile and Ubiquitous Applications*, Pichler discusses how to provide added value to mobile users, maintaining the importance of designing services that are very specific to the context area, and how to foster communities based on both physical and logical contexts.

Interactions

Of course, interaction problems are paramount. One of the key issues when working with mobile devices is how to input data to a mobile device with very poor input devices. The other, symmetrical, key issue is how to fully exploit the small available display area. The second paper of this part discusses the former; the other ones the latter. In *Accessing Web Educational Resources from Mobile Wireless Devices*: *The Knowledge Sea Approach*, Brusilovsky et al. evaluate the use of Self-Organizing Maps (SOMs) for information access to educational resources. In *Spoken Versus Written Queries for Mobile Information Access*, Du et al. analyze IR effectiveness when the query is input via speech: they present a prototype and its experimental evaluation. In *Focussed Palmtop Information Access Combining Starfield Displays with Profile-Based Recommendations*, Dunlop et al. present two applications using starfield displays on a PDA and exploiting advanced collaborative filtering techniques: Taeneb CityGuide recommends restaurants and Taeneb ConferenceGuide presents the timetable of a conference.

Applications and Experimental Evaluations

Several approaches are used for implementing applications. Following a strong tradition in both the HCI and IR communities, evaluation is deemed a crucial issue and several papers focus on experimental studies of mobile applications. In *Designing Models and Services for Learning Management Systems in Mobile Settings* Andronico et al. propose a survey of previous systems for mobile learning, and describe an ongoing project. Cignini et al., in *E-Mail on the Move*: *Categorization, Filtering, and Alerting on Mobile Devices with the ifMail Prototype*, present a prototype allowing e-mail categorization, filtering, and alerting on mobile devices, and its first experimental validation. In *Mobile Access to the Físchlár-News Archive*, Gurrin et al. illustrate the Físchlár-News system, processing digital video and audio news stories, which is capable of segmentation, collaborative filtering-based recommendation, and delivery on mobile devices. Mai et al., in *A PDA-Based System for Recognizing Buildings from User-Supplied Images*, describe a prototype providing navigational and informational services to an urban mobile user based on GPS and building recognition achieved through image processing techniques. In *SmartView and SearchMobil*: *Providing Overview and Detail in Handheld Browsing*, Milic-Frayling et al. overview their SmartView technology, which makes Web pages with complex layout more accessible to mobile devices, and show and evaluate its integration into SearchMobil, to help the users of a small screen display estimate the relevance of retrieved Web pages. The paper titled *Compact Summarization*

for Mobile Phones, by Seki et al., deals with the very important (for mobile devices) issue of summarization: these authors present a new summarization method based on the genre of a document and they evaluate it. On the same topic, Sweeney et al. in *Supporting Searching on Small Screen Devices Using Summarisation* discuss and evaluate by means of a user test how summarization can improve IR on small screen devices. In *Towards the Wireless Ward*: *Evaluating a Trial of Networked PDAs in the National Health Service*, Turner et al. discuss and evaluate, by means of an on-field user study, several important issues on the usage of PDAs in the medical field. Finally, in *Aspect-Based Adaptation for Ubiquitous Software*, Zambrano et al. delve into software engineering issues: they propose Aspect Oriented Programming (AOP) as a solution to deal smoothly with issues that are peculiar to the design of mobile device applications and that are not found when designing standard desktop applications.

Context and Location

A hot issue in mobile device research is, of course, how to take into account and exploit the context in which the user is. In *Context-Aware Retrieval for Ubiquitous Computing Environments*, Jones et al. perform a thorough analysis of context-aware retrieval: they present definitions, links with other disciplines (IR, information filtering, agents, HCI), and a description of their own findings. Nussbaum et al., in *Ubiquitous Awareness in an Academic Environment*, propose and evaluate a prototype that, on a campus, enhances student relationships by fostering face-to-face meetings. In *Accessing Location Data in Mobile Environments*: *the Nimbus Location Model*, Roth proposes the Nimbus framework, a formal model for location information, integrating physical and semantic information. The paper *A Localization Service for Mobile Users in Peer-to-Peer Environments*, by Thilliez et al., describes a localization service based on a peer-to-peer (P2P) architecture, featuring location-based queries. Finally, in the last paper of this volume, *Sensing and Filtering Surrounding Data*: *the PERSEND Approach*, Touzet et al. present an application dealing with the issues of distributed databases, proximate environments, and continuous queries.

Acknowledgements

We thank the organizers of Mobile HCI 2003 for their support of the workshop. We also thank the members of the Program Committee (Peter Brusilovsky, George Buchanan, Keith Cheverst, Oscar de Bruijn, Marcello Federico, Matt Jones, Mun-Kew Leong, Joerg Roth, Ed Schofield, and Leon Watts) for their efforts on behalf of the workshop.

December 2003

Fabio Crestani
Mark Dunlop
Stefano Mizzaro

Organizing Committee
Mobile HCI 2003 Workshop on
Mobile and Ubiquitous Information Access

Table of Contents

Foundations: Concepts, Models, and Paradigms

Interactions

Applications and Experimental Evaluations

Context and Location

The Concept of Relevance in Mobile and Ubiquitous Information Access

Paolo Coppola[1], Vincenzo Della Mea[1], Luca Di Gaspero[2], and Stefano Mizzaro[2]

[1] Department of Mathematics and Computer Science
University of Udine
Via delle Scienze, 206 – Loc. Rizzi – Udine – 33100 Italy
{coppola,dellamea,mizzaro}@dimi.uniud.it
http://www.dimi.uniud.it/~{coppola,dellamea,mizzaro}
[2] Department of Electrical, Management, and Mechanical Engineering
University of Udine
luca.digaspero@diegm.uniud.it
http://www.diegm.uniud.it/digaspero/

Abstract. We discuss how the wireless-mobile revolution will change the notion of relevance in information retrieval. We distinguish between classical relevance (e-relevance) and relevance for wireless/mobile information retrieval (w-relevance). Starting from a four-dimensional model of e-relevance previously developed by one of us, we discuss how, in an ubiquitous computing environment, much more information will be available, and how it is therefore likely that w-relevance will be more important than e-relevance to survive information overload. The similarities and differences between e-relevance and w-relevance are described, and we show that there are more differences than one might think at first. We specifically analyze the role that beyond-topical criteria have in the w-relevance case, and we show some examples to clarify and support our position.

1 Introduction

It may surprise you, but we can hardly imagine what information overload is. Just stop one minute and think how our world will probably be in ten years or so. As soon as mobile wireless devices will ubiquitously enter our lives, the nowadays complaints about having access to too much information will be seen with a small ironic grin and perhaps some nostalgia. We are not speaking only of palm top devices, cellular phones, laptop computers, pagers, MP3 players, and similar already commonly used device; we are thinking also of networked digital cameras and video-cameras, thermometers, traffic lights, GPSs for cars and – why not – bikes, skates, pogosticks, and even walking people, game stations, and so on. Thousands of interconnected information processing devices will be available to each of us anytime anywhere. Each mobile device will sense its environment to gather information from the physical world and make it available to its user (or users). Each device will also exchange information with other (mobile and non-mobile) devices, mainly by means of some wireless communication network. Probably, users will (continue to) directly exchange information among them. Also, devices will probably change the physical environment, to

F. Crestani et al. (Eds.): Mobile and Ubiquitous Info. Access Ws 2003, LNCS 2954, pp. 1–10, 2004.

a greater extent than nowadays static and non-ubiquitous desktop machines. A similar view is expressed, for instance, in [12].

All the mobile devices can be seen, from the user point of view, as information access tools: they will filter incoming information and retrieve available information, trying to present to the user all and only the relevant information. Of course, the user will be interested in accessing information that is not only relevant in the strict sense, but also of a high quality, timely, serendipitous, of the appropriate grain size, perhaps rare, and so on. Since there is not an agreement about which of these features are relevance features, we will use the term "relevance" in a very general way, denoting with relevant information the information that the user wants.

But what is relevance in the new mobile/wireless/ubiquitous scenario? This paper is a first and preliminary attempt of answering this question. We hope both to help traditional information retrieval researchers to appreciate some complications peculiar to the mobile domain, and to persuade researchers working in the mobile and wireless field of the importance of the information access approach. Therefore, we try to stay at a level high enough to be understandable by an interdisciplinary audience.

The paper is organized as follows. In Section 2 we will briefly overview the research about the concept of relevance in classical non-mobile *Information Retrieval* (henceforth IR). We name relevance in classical IR *e-relevance* (for electronic relevance, but this is not the only reason, as we will explain in Section 5). In Section 3 we will re-analyze the relevance concept in the mobile case. In turn, we name this relevance *w-relevance* (for wireless relevance, but, again, see Section 5). We show that, from an intuitive point of view: (i) w-relevance is an extension of e-relevance; (ii) w-relevance is much different from e-relevance than one might think at first; and (iii) beyond-topical criteria, one aspect of e-relevance that has recently received a lot of attention in non-mobile IR, are both much more emphasized and much more important in the mobile case. In Section 4 we propose some simple examples and scenarios to support our position. Section 5 concludes the paper.

2 E-Relevance: The Non-mobile Information Retrieval Case

Relevance (e-relevance) is a subject that has been intensely studied for years in the IR field, and it is still a hot topic today. We will not review in detail the field, since some well known surveys are already available [14, 20, 21, 22, 23].

Classical information retrieval equates e-relevance with topicality: the query submitted to an IR system specifies the topic(s) that a relevant document has to deal with. For example, if a university professor is looking for documents to prepare her next lesson for this afternoon, she needs of course documents that deal with the matter that she is going to explain to her students. But she also wants those documents as soon as possible (if a document arrives after the lesson, it is useless), at the right complexity level (if a document is too difficult, students will not understand it), and so on. And these features go beyond the topic: they are completely independent of it.

Therefore, the topical view is short-sighted. Indeed, we have now a large amount of research that demonstrates how topic is only one of the criteria that users use when judging the e-relevance of the retrieved documents. For a review of this line of research, that started in the 60es and has received a lot of attention (especially at Syracuse University) in the 80es and 90es see [1, 14]. Since the criteria, elicited from

users or found by experts, tend to constitute a stable set (i.e., very few new criteria are found in the most recent studies), it is likely that we have an almost correct and complete list of relevance criteria.

Actually, the exploitation of beyond-topical criteria is not the only way to get closer to the "real" relevance, i.e., the relevance the user is interested in. A more general approach that takes into account this aspect has been proposed by one of us some years ago [9, 15]: the various kinds of relevance are classified in a four-dimensional space, distinguishing among them on the basis of a precise classification. The four dimensions are:

- *Information resources*, containing document, surrogate, and the information that the user receives when reading a document.
- *Representation of the user problem*, containing the real information need, the perceived information need, the request (or expressed information need), and the query (or formalized information need).
- *Time*, containing the time instants from the arising of the user's need to its satisfaction.
- *Components*, containing topic, task (what the user has to do with the retrieved information), and context (everything beyond topic and task as, for example, what the user already knows about the topic being sought, or the time that the user has to complete the search).

These four dimensions allow one to distinguish among the various kinds of relevance, and to speak, for instance, of: the relevance of a document to the query at query expression time for what concerns the topic component (the classical relevance used in IR); the relevance of the information received to the real information need at the time of final need satisfaction for what concerns topic, task, and context (the relevance the user is interested in); and so on. This classification can be used in the implementation and evaluation of IR systems.

This topic/task/context distinction has been used in some respect. Reid [18] proposed an evaluation methodology that uses the task as the starting point for building a test collection. The development of IR systems dealing with beyond-topical e-relevance has been rather slow, however some examples now exist. Researchers at MIT recently developed an IR system that, in some way, goes beyond topical criteria [13]. This system, named GOOSE (GOal Oriented Search Engine), allows the user to choose among a list of tasks (called "goals" by GOOSE authors), and uses a large common sense knowledge base to exploit the task specification for building a better query. In such a way, Liu and colleagues implemented, perhaps without explicitly noting it, an IR system that tries to work taking into account beyond-topical factors of relevance, as suggested in [15].

One can also assume that, although each search and each information need concern a different topic, there are indeed some beyond-topical components of user's needs that are more stable, i.e., the context in which the consecutive search sessions by one user take place [10, 11]. Some first experiments show that, for a given user, contexts are indeed more stable than topics, and may be used to improve the ranking of documents retrieved after a query, but the usefulness of this approach is still under investigation.

Another approach for including beyond topical criteria in an IR system is to build an *IR assistant*, namely a system that, during information seeking, observes user be-

havior and gives suggestions aimed at improving the effectiveness of the search and of the searcher [3, 4, 16]. Some of the suggestions might be of a topical nature (e.g., to add some terms to the query to better represent the topic being sought for), but also non-topical suggestions can be provided, like suggesting a paper related in some way to those judged as relevant so far (e.g., the PhD thesis by, or a short biography of, the author of a paper judged as relevant, or a references list, and so on). This line of research has still to be proven effective, but initial laboratory experiments show positive results. Also "just-in-time information retrieval agents" [19] build their queries with beyond-topical components (mainly context).

Even if the existence of beyond-topical criteria for e-relevance is not in discussion, what seems not yet recognized, or assessed, is the actual importance of these criteria in real-life IR. In the next section we discuss, on the basis of the classification in [15], how and why the w-relevance scenario is different.

3 W-Relevance: The Mobile Information Retrieval Case

One might simply repeat the above analysis in the w-relevance case, and thus just state that there are various kinds of w-relevance and there are some beyond-topical components of w-relevance that should not be overlooked. However, we believe that there are important differences between e-relevance and w-relevance. The beyond-topical criteria in the mobile IR case become more critical: they are different from, and have a higher importance than, those in non-mobile IR. Therefore, topicality is an abstraction that works in a perhaps satisfying way (even far from perfect) in the e-relevance case but, as soon as the real world comes into playthe shortcomings of this approach are manifest (examples will be shown in Section 4). Also, there are more kinds of w-relevance than kinds of e-relevance. As we will discuss in the following, the main reason for these differences is that in the e-relevance case we can comfortably seat inside the "information world", whereas in the w-relevance case we have to move into the "real/physical world".

All the e-relevance models proposed in past years need to be modified to become adequate models of w-relevance. In this section we revise and extend the model proposed in [15], in each of the four above mentioned dimensions.

3.1 Information Resources

In the non-mobile case, the user of an IR system is usually interested in retrieving information; a typical user is a scholar that needs documents on a new topic, to study them, to write a paper or book, and so on. This is obtained by retrieving a number of information sources (books, articles, Web pages, etc.), from which the user can extract the relevant information. In the mobile IR scenario, it is often the case that the user is interested not just in information, but in obtaining some (possibly material) *thing*, only partially described by information (e.g., a physical place or a pair of blue jeans): besides surrogate, document, and information, the information resources dimension should therefore include also the things, and should perhaps be renamed as *resources*. In other terms, often the retrieved information and, in general, the database

are instrumental, since they are means to reach the end of possessing, or obtaining, some thing, not the end itself. Besides the relevance of retrieved information, we also have the relevance of the retrieved thing: the user will not evaluate the information sources, but the described physical object, that in the meantime might change or disappear even without an immediate reflection on the information source content. This brings up the issue of consistency between the database and the real world.

3.2 Representation of the User Problem

Since in the mobile IR scenario, besides the real information need (that is in turn beyond the information need perceived by the user), it is often the case that the user is interested in some *thing*, we can say that the user usually has a *thing need* (that should then be added to the second dimension, namely the representations of the user problem). Therefore, if we look at the first two dimensions, we can say that from the relevance of the retrieved information to the information need, we have moved to the relevance of the retrieved thing to the thing need. Using Bateson's [2] terminology, w-relevance deals more with *Pleroma* (the physical world), whereas e-relevance deals mainly with *Creatura* (the informational world): in w-relevance we have a much stronger coupling with the real, physical world. If one photocopies an article in a library (e-relevance scenario), you can anyway read the article later. If someone buys the last item of your favorites blue jeans just after your query to a "blue jeans database", you cannot have them anymore (w-relevance scenario). In the former case you are interested in information, whereas in the latter you are interested in a thing.

3.3 Time

Another dimension of relevance that increases its importance in the w-relevance case is *time*, in two senses. First, often the user needs "quick and dirty" information: things change faster, replication is more difficult. Second, in the real world, since time is irreversible, if something is lost it is lost. In the Creatura one can often rely on backups, copies, and replication; in the real world, "carpe diem". This is perhaps the deep motivation behind the often stated claim that users of mobile devices are more interested in precision than in recall, usually justified, in a perhaps too simplistic way, by the small display area on mobile devices: having a full list of the relevant items can be useless if the list is so long that the time required for examining it is longer than the lifetime of relevant items.

Another aspect of w-relevance, related to both the strong coupling with the real-world and time, is the database change rate: since the real world changes quickly and continuously, the database has to quickly change accordingly to stay up-to-date.

The intuitive importance of time is also confirmed by a survey made last year in Singapore among users of PIRO, a commercial system developed by C5solutions [5]. PIRO presents to mobile users using WAP phones the directory listings of commercial retail relevant to user's current need. Eight users filled in a 29 questions questionnaire having the purpose to rate the importance of various relevance criteria for presenting commercial applications on a mobile device. Of course the small sample size does not allow any certain inference, but it is worth noting that two out of the three

highest rated criteria are "information is current (up to date)" and "information is about a sale or promotion or money saving opportunity", both of which concern time features.

3.4 Components

"Context" is a hot word in the mobile/wireless scene, but with a different meaning from that used above [6, 7, 8, 17]: context usually refers to the current environment, the situation, that the user of a mobile device is experiencing while using it. Let us see some examples of this usage of the term.

Location is one of the most mentioned aspects of context [26]: from the user position (derived by means of GPS, or triangulation in a Bluetooth or Wi-Fi network) other information can be inferred and exploited in various ways, for example to increase the relevance of the information accessed by the user, or to improve the interaction with the user.

Of course, location is important, but there is more to context than location [24]. Location itself is not the only information we can get from the spatial position of the user. Indeed, this feature is only as a "static" one, whereas several additional information can be inferred from the dynamic evolution of locations. For example, if the user looks for traffic information, and she is moving along a road, it is very likely that she wants information about the road she is currently on, rather than the whole national traffic news. Therefore, user's *track*, i.e., the temporal sequence of locations traveled by a user, is another aspect of context. Let us notice that context can also be predicted; for example, a full track can be inferred if the user's scheduler reveals that she has an appointment in half an hour at a certain place. Also the traveling speed that the user has while following a certain track is an important parameter: a slowly walking user can be presented more information than a running one [25].

Other common examples of context aspects are: the noise level in the environment (that can and should affect the volume of a mobile phone); the light level in the environment (affecting the display illumination); the orientation of the device (affecting the orientation of the displayed information) [24].

Therefore, "context" has a different meaning in w-relevance: in e-relevance, context concerned what was in user's mind only (perhaps mainly); in w-relevance, context is also (perhaps mainly) about the real world. We will distinguish between these two meanings by using the terms *e-context* and *w-context*. W-context is more general than e-context, is much more dynamic, and it is more likely to change during the information seeking activity. It is also worth noting that the above mentioned Göker's preliminary results on context stability [10, 11] might not hold in the mobile environment. On the other hand, whereas e-context has to be provided manually to an IR system, w-context is likely to be autonomously derived in an easier way; the reason is that many of the components that belong to w-context and do not belong to e-context can, at least theoretically, be inferred automatically (e.g., location, track, noise level, presence of other persons, and so on). Yet, the use of automatically derived w-context might result in a more inaccurate classification of items, due to the assumptions introduced in the model exploited for building the w-context. This might lead to avoid filtering out the items estimated by the system as not w-relevant, and to prefer a more careful information visualization approach, e.g., to rank the retrieved items.

4 W-Relevance Scenarios

In this section we show some realistic scenarios that support the above discussion. We are aware of the importance of privacy issues, but we do not take them into account in this paper.

4.1 Catching the Train

Let us consider a query for finding a train from Udine, Italy to Milano, Italy, made to the Italian national railways web site (http://www.trenitalia.com). At present, you fill a form with (at least) departure and arrival cities, and starting date and time. If the two cities have more than one railway station, you are also asked to select the specific stations you want. Then, you receive a list of trains since that hour, and if you want details on one of them, you have to click on its link, then go to the price link and further compile a form where you specify how many tickets, in which class, etc.

What if you are reaching the station in a hurry on a taxi, just in time for a train? You do not need to be informed about all the trains from Udine to Milano in the next two hours: you just need quickly to go where your are used to go, i.e. Milano Piazza Garibaldi railway station, in second class as usual, by the first train available. Other options include: you are not alone, but with your husband/wife (your PDA is sensing him or her around you); the ticket can be bought automatically (using the available details of your credit card); and the first train retrieved could be not useful because there are not two free seats in second class. All these data can be derived from your own w-context and an up-to-date (with respect to the real world) database.

4.2 Driving to a Conference

Let us imagine that you are driving your car, rented at Venice airport, towards Udine to attend Mobile HCI 2013 conference. Your car is of course equipped with a GPS and a driving assistant, giving you directions about the route to follow. In this situation, the information that the car 100 meters in front of you is going to Udine too is very relevant, and should be immediately notified to you so that you might follow that car without worrying about road directions. Moreover, if the driver in the other car is a good friend of yours, you should be notified about that too, since you might want to contact her (with an SMS?) for sharing the trip or just having a coffee together.

In this scenario, the topic is straightforward, being the destination of your trip (Udine); the task is given mainly by driving in a convenient way, with perhaps some subtasks given by sharing the trip for economy, avoiding pollution, just chatting. The latter case is even more interesting if the driver is a friend, but this is not expressed neither in topic nor in task, but most likely in the e-context – your address book, your last phone calls, etc. You might go on, and think of the situation if your good friend is not a good driver at all, or if her car is a very old one (and these are w-context aspects).

4.3 E-Commerce Application

Now you are a trendy boy/girl, shopping around in a commercial center, and willing to buy some fashionable trousers (“Gasoline” brand), and a newly available “Mos-

quito" shirt. You do not want to spend too much money, so you ask your PDA to look for those dresses at a good price. Some different outcomes might be considered: (i) your are using a traditional e-relevance based IR system, thus you will receive a list of offers on eBay, followed by some online shop catalogue showing very good prices for the same dresses; unfortunately, you are around for shopping, you want to have your dresses now and not to wait for their postal delivery; (ii) your PDA queries a w-relevance based IR system which, on the basis of your location, track, and walking speed, is able to infer that you probably want some specific place where to buy such dresses, possibly close to you.

So you will receive three shop addresses: the closest one is not the first because it is slightly more expensive than another one, which in turn is sufficiently near to be reached before closing time. A third shop is listed with good prices but with just one shirt of your size (as recorded in your personal profile, or communicated after request by the radio/infrared label applied on the shirt you are actually wearing); you have to run before someone else buys it, or perhaps you can book it by means of an electronic message. Again, the topic is Gasoline trousers, Mosquito shirt, good price, but the real task is to actually buy them, not to know where to buy them. Real things get sold out, usually on a first-came first-served basis: what is true now (e.g., availability of my size) could be false in some minutes, thus time matters too.

4.4 A Museum Application

Just in front of a beautiful Van Gogh picture, you want to have some more information to understand why he is painted with a bandaged ear. The wireless service available there of course does not need to show you the picture itself: you just need textual background information, as in you current w-context there is the real availability of the object which generates the topic for the query. In the wireless-enabled environment, your PDA should handshake with a radio/infrared label applied near the picture, so that, in addition to exactly know your position, it is also able to automatically generate part of the query.

5 Conclusions and Future Work

In this paper we have shown how the mobile/wireless/ubiquitous revolution is likely to bring big changes into the IR field, and how even a very foundational concept as relevance needs to be re-analyzed and re-defined.

The relevance in classical electronic environments (non-mobile ones) that we have named *e-relevance*, is actually an *irrelevance*, because many features of it are neglected, or at least not given the importance that they should have in the general case. In the mobile IR case, this generality is more easy to notice: *w-relevance* does not only mean wireless relevance, but also *double-relevance*, *world-relevance* (since the physical world is much more involved) and *double-user-relevance*, since it is a notion of relevance that is much more close to what the users want and need.

The current model for information retrieval – with one user and one system – is also challenged in a peer-to-peer wirelessly connected environment, where ultra-mobile devices are available for providing information to each other. Information will be available from many devices through many channels, either phone-like (WAP,

GPRS, UMTS) or local networking (Wireless LAN, Bluetooth, IrDA). Such devices may in turn provide also w-context information, i.e., location (in a broad sense, not only geographical coordinates), track, temperature, etc. In such a complex peer-to-peer scenario, it is likely that a single query made by my device could be answered by more than one system, and that each system could be engaged in a sort of "reverse relevance", asking to itself something like "Am I able to answer to such a query?", which in turn could be translated as "Is such a query relevant to my database?". As device answers may have a cost for the user, it is also likely that the query should involve budgetary considerations. Useful hints about how to deal with such a kind of interactions may come from the multi-agent paradigm research area [27].

Moreover, another problem needs to be mentioned. On the one side, it seems reasonable that as soon as some information is available and potentially relevant in the future, it should be stored locally on one's own device to be accessible later. This is even more reasonable if one takes into account that wireless devices are not always connected to the network, and that they use different network connections, with different transfer rates, reliability, privacy, and cost. However, on the other side, small devices are more resource constrained: low computational and storage power, low energy availability, low bandwidth also in the interaction with the user. All these features would suggest that the local storage of information is not always the best choice. Also, the locally stored data can quickly become outdated because of the quick change rate of the database, therefore rising inconsistency problems. These issues need further investigation.

In the future we plan to work on the relevance four dimensional model in order to make it more accurate and formalized. We also intend to use the revised model in the implementation and evaluation of mobile IR systems.

Acknowledgements

We would like to thank Roberto Ranon for interesting discussions and useful comments on an earlier draft of this paper, and two referees for their constructive remarks.

References

1. Barry, C. L., Schamber, L. Users' Criteria for Relevance Evaluation: A Cross-Situational Comparison, *Information Processing and Management* 34(2/3), 1998, 219-236.
2. Bateson, G. *Mind and Nature – A Necessary Unit*. E. P. Dutton.
3. Brajnik, G., Mizzaro, S., Tasso, C., Venuti, F. Strategic help in user interfaces for information retrieval, *Journal of the American Society for Information Science and Technology*, 53(5), 2002, 343-358.
4. Brajnik, G., Mizzaro, S., Tasso, C. Evaluating User Interfaces to Information Retrieval Systems: a Case Study on User Support, *Proceedings of the 19th Annual International ACM SIGIR Conference*, Zurich, CH, 1996, 128-136.
5. Insuring an Industry, Investing in People. *Career Times* 2001/11/09. http://www.careertimes.com.hk/english/employers/mgmt/features/20011109.asp
6. Chen, C., Kotz, D. *A Survey of Context-Aware Mobile Computing Research*. Dept. of Computer Science, Dartmouth College Technical Report TR2000-381, November 2000, ftp://ftp.cs.dartmouth.edu/TR/TR2000-381.ps.Z.

7. Cheverst, K., Davies, N., Mitchell, K., Friday, A. The role of connectivity in supporting context-sensitive applications, *Handheld and ubiquitous computing, Proceedings, LNCS, 1707*, 1999, 193-207.
8. Cheverst, K., Smith, G., Mitchell, K., Friday, A., Davies, N. The role of shared context in supporting cooperation between city visitors, *Computers & Graphics*, 25(4), 2001, 555-562.
9. Gabrielli, S., Mizzaro, S. Negotiating a multidimensional framework for relevance space. In S. W. Draper, M. D. Dunlop, I. Ruthven, and C. J. van Rijsbergen, editors, *Mira 99: Evaluating interactive information retrieval - Proceedings of MIRA 1999 Conference*, eWiC - electronic Workshops in Computing, pages 1-15, 1999.
10. Göker, A. Context Learning in Okapi, *Journal of Documentation*, 53(1), 1997, 80-83.
11. Göker, A. Capturing Information Need by Learning User Context, *Sixteenth International Joint Conference in Artificial Intelligence: Learning About Users Workshop*. 1999, 21-27.
12. Lettieri P., Srivastava, M.B. Advances in wireless terminals, *IEEE Personal Communications*, 6 (1), 1999, 6-19.
13. Liu, H., Lieberman, H., Selker, T. GOOSE: A Goal-Oriented Search Engine with Commonsense. In Paul De Bra and Peter Brusilovsky editors, *Second International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, Malaga, 29-31 May 2002, 253-263.
14. Mizzaro, S. Relevance: The whole history. *Journal of the American Society for Information Science*, 48(9), 1997, 810-832.
15. Mizzaro, S. How many relevances in information retrieval? *Interacting With Computers*, 10(3), 1998, 305-322.
16. Mizzaro, S., Tasso, C. Ephemeral and Persistent Personalization in Adaptive Information Access to Scholarly Publications on the Web. In Paul De Bra and Peter Brusilovsky editors, *Second International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, Malaga, 29-31 May 2002, 306-316.
17. Pascoe, J. Adding generic contextual capabilities to wearable computers, in *Proceedings of 2nd International Symposium on Wearable Computers*, October 1998, pp. 92-99.
18. Reid, J. A Task-Oriented Non-Interactive Evaluation Methodology for Information Retrieval Systems, *Information Retrieval*, 2(1) , 2000. 115-129.
19. Rhodes, B. J., Maes, P. Just-in-time information retrieval agents, *IBM Systems Journal*, 39(3&4), 2000, 685-704.
20. Saracevic, T. Relevance: A review of and a framework for the thinking on the notion in information science. *Journal of the American Society for Information Science, 26,* 1975, 321–343.
21. Saracevic, T. Relevance Reconsidered '96. In P. Ingwersen and N. O. Pors, editors, Information Science: Integration in Perspective - Proceedings of CoLIS2, pages 201-218, Copenhagen, Denmark, October 1996. The Royal School of Librarianship.
22. Schamber, L. Relevance and information behavior. *Annual Review of Information Science and Technology,* 29*,* 1994, 3-48.
23. Schamber, L., Eisenberg, M. B., Nilan, M. S. A re-examination of relevance: Toward a dynamic, situational definition. *Information Processing & Management,* 26(6), 1990, 755–776.
24. Schmidt, A., Beigl, M., Gellersen, H.-W. There is more to context than location, *Computer & Graphics Journal*, 23(6), 1999, 893-902.
25. Wahlster, W. Resource-Adaptive Interfaces to Hybrid Navigation Systems (Keynote Talk), In Paul De Bra and Peter Brusilovsky editors, *Second International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, Malaga, 29-31 May 2002, 12-13.
26. Ward, A., Jones, A., Hopper, A. A New Location Technique for the Active Office. *IEEE Personnel Communications*, 4(5), 1997, 42-47.
27. Wooldridge, M. *An Introduction to Multiagent Systems*. John Wiley & Sons, 2002.

Conversational Design as a Paradigm for User Interaction on Mobile Devices

Mun-Kew Leong

Institute for Infocomm Research (I^2R),
21 Heng Mui Keng Terrace, Singapore 119613
mkleong@i2r.a-star.edu.sg

Abstract. This paper borrows the Cooperative Principle and the idea of conversational implicatures from H.P. Grice's *Logic and Conversation*. We apply it to user interface design, specifically for mobile devices. We introduce the idea of *conversational design,* which is that user interfaces should be designed to flow as if they were a conversation between two cooperative entities. We present a case study on mobile phone user interface and show where conversational design provides a new perspective on interface design and how such an analysis helps create better interfaces for mobile devices.

1 Introduction

This paper introduces the idea of conversational design inspired by readings in the Philosophy of Language, specifically by the work of H.P. Grice[1]. By definition, human conversation is a cooperative endeavor with both parties working to exchange information. Otherwise, it would merely be two simultaneous monologues.

Grice proposes what he called *Conversational Maxims*, which are non-prescriptive rules of what each party[2] in a conversation expects from the other. What is remarkable about the maxims is the way they generalize out of the linguistic domain into any form of communication between two (or more) parties where there it is possible to distinguish a series of interactions between them. We propose to use these maxims as design principles in computer human interaction, and in formulating test suites for judging the efficiency and completeness of the interaction.

What is also important is that, linguistically, human beings are not slaves to the maxims. They are often broken (maliciously or inadvertently) and this results in either a partial or complete breakdown in the communication flow. The conversation, however, does not end in the case of a breakdown; rather there is a (series of) correction(s) which automatically puts the conversation back on track. One may suggest that the maxims codify certain assumptions on the part of the speaker and hearer which allows conversation to occur fast (the assumptions reduce the search space of the context), fluidly (shared assumptions), but not always accurately (misunderstandings).

---

[1] H. Paul Grice was an influential philosopher of language in the middle part of the 20th century. He is best known for his work on conversational implicatures, first presented at the 1967 William James Lectures.

[2] For simplicity, we assume our conversations occur between two parties and not more.

F. Crestani et al. (Eds.): Mobile and Ubiquitous Info. Access Ws 2003, LNCS 2954, pp. 11–27, 2004.

This has interesting ramifications for user interface design as well. If we, as humans engaged in conversation, sometimes breakdown and automatically repair, then can we not use that as a model for user interface design? That is to say, we design a UI that works fast and fluidly but which may occasionally makes mistakes (but which can be repaired) rather than trying to achieve error-free communication.

The following section introduces Grice's Conversational Maxims and provides examples of how a language motivated idea translates into other (non-linguistic) modes of communication. In particular, we will argue that mobile device interaction takes the form of a conversation between the user and the device. Section 3 defines *conversational design* including the idea of *breakdown and repair* and provides a case study analyzing the user interface of some prototypical mobile devices. Section 4 validates the intuitions of the previous section with some preliminary experiments specifically on mobile phones. Section 5 ties the idea of conversational design to the theme of this collection of papers, i.e., to mobile information access, and also considers some implementation issues. We conclude with a summary and some discussion of future work.

2 Background and Motivation

Cross-disciplinary approaches to user interface design are common, but they mostly borrow from psychology and cognitive science, e.g., the papers in [1] and [2]. In this paper, however, we are borrowing from the Philosophy of Language, specifically from the work of H. Paul Grice in *Logic and Conversation* [3]. We introduce Grice's work on conversational implicatures, leading to his conversational maxims. We show how these maxims apply in non-linguistic modes of communication, ending with an example on a mobile device interface.

2.1 Conversational Implicatures and the Cooperative Principle

In *Logic and Conversation* [3], H.P. Grice distinguishes between conventional and unconventional implicature[3] in language. Conventional implicature occurs when one part of a sentence (or utterance) explicitly implies another, e.g., "He's fat therefore he likes to eat". Unconventional implicature occurs when what is implied by a sentence is different from the lexical content of the sentence itself. A common example comes from John Searle's *Speech Acts* [4] where "Can you pass the salt?" implies a request to pass the salt and is neither a question (despite the grammatical form) nor a query on one's ability to move the salt shaker (which is the lexical content). Other examples include uttering "How do you do?" which is just saying hello or "I could eat a horse!" which merely expresses a high degree of hunger rather than a desire for an equine dinner.

Grice, however, was interested in a subclass of non-conventional implicatures which occur in the context of a conversation. A conversation takes place between two or more parties over time. There is an exchange of utterances (interactions) with intent to accomplish a common task. This is in contrast to a command or an acknowl-

[3] *Implicature* is just a fancy way of making a noun out of *implies*.

edgement which is strictly unidirectional. So, a conversation has to be a cooperative effort between the parties concerned. Thus one may formulate a *Cooperative Principle*, specifically:

> Make your conversational contribution such as is required, at the stage at which it occurs, by the accepted purpose or direction of the talk exchange in which you are engaged.

In other words, a conversation *implies* that the utterances of the parties involved have two components: (a) the lexical content which explicitly exchanges information between the parties, and (b) the unspoken structure which explicitly works to keep the conversation going smoothly and efficiently.

It is this second component which Grice calls *conversational implicature*. This may be thought of as a moving set of expectations by each party about the utterances of the other party. Grice's major contribution in *Logic and Conversation* was to codify the rules (or *maxims* as he calls them) which generate these expectations over the course of a conversation. These maxims are presented in the next section.

2.2 Conversational Maxims

There are four categories of maxims which result from the Cooperative Principle: Quantity, Quality, Relation and Manner[4]. These are listed below and abstracted almost verbatim from [3], and should be self-explanatory.

The Maxims of Quantity
This category of maxims deals with the amount of information a conversational party would provide:

a. Make your contribution as informative as is required (for the current purpose of the exchange)
b. Do not make your contribution more informative than is required.

The Maxims of Quality
This category has a supermaxim:

a. Try to make your contribution one that is true

As well as two more specific maxims:

a. Do not say what you believe to be false.
b. Do not say that for which you lack adequate evidence

The Maxims of Relation
This category has a single maxim:

a. Be relevant.

[4] These four categories are named for the categories of perceptual judgment in Immanuel Kant's *Critique of Pure Reason* [5]

The Maxims of Manner
Unlike the previous maxims, this category relates not to *what* is said but *how* it is said, and also has a supermaxim:

a. Be perspicuous, i.e., clearly expressed and easy to understand

And various specific maxims:

a. Avoid obscurity of expression
b. Avoid ambiguity
c. Be brief (avoid unnecessary prolixity)
d. Be orderly

2.3 Examples of Conversational Implicature

Conversational maxims attempt to codify the implicit implicatures in normal conversation. As such, positive examples of the implicatures would be trivial and not particularly illuminating. What Grice does, and what we will do here as well, is to examine examples where the conversation breaks down to illustrate specific maxims being violated.

We will provide examples of maxims being violated in spoken conversation (the "norm"), in non-verbal exchanges, in a purely graphic exchange, and last, in the context of mobile devices. We will assume that, in each scenario below, the Cooperative Principle holds.

Spoken Conversation
Scenario: My car is running out of petrol; I stop a passerby to ask where is the nearest petrol station. Here are examples of the maxims being violated. No explanations are provided as it should be obvious which maxim or sub-maxim is being violated:

- *(Maxim of) Quantity*:
  - I: Excuse me! Hello?!
  - Stranger on the road: Yes? Can I help you?
  - I'm looking for a petrol station. Is there one nearby?
  - Yes, just two blocks down this street; it's a new one, just built last year and they have 7 pumps, two of which sell premium gas, and 1 which is for diesel. They don't really sell much diesel, but the law requires them to carry it.
- *Quality*:
  - I'm looking for a petrol station. Is there one nearby?
  - No, you'll have to drive to the next town.
- *Relation*:
  - I'm looking for a petrol station. Is there one nearby?
  - That's the problem with the world today! Too much reliance on fossil fuels! You should be walking; it's good for you!
- *Manner*:
  - I'm looking for a petrol station. Is there one nearby?
  - Yes (then walks away)

Non-verbal Communication
Scenario: You're helping me to fix a broken door. Similarly, it should again be obvious which maxim or sub-maxim is violated:

- *Quantity*: I ask for two screws, you give me four.
- *Quality*: I ask for a screwdriver, you pass me a wrench.
- *Relation*: I'm hammering a nail; you pass me a good book to read.
- *Manner*: I'm on top of a ladder and ask for 2 screws; you go to the kitchen for a glass of water then come back and give me the screws.

Graphical Communication:
Scenario: You see the *all* of following official road signs on the same lamp post as you drive by:

- *Quantity*:  - in this case, once you have the straight-only road sign, there is no need for the other two.
- *Quality*:  - an obvious contradiction.
- *Relation*:  - the give-way sign is irrelevant given the no-entry sign.
- *Manner*:  - the speed limit is ambiguous.

Mobile Device Interaction
Scenario: I pull out my trusty PDA (personal digital assistant) and select the address book function, intending to call my friend, Gareth, at his office. Basically, I'm "asking" for his office phone number, and this is what happens:

- *Quantity*: I ask for his office phone number, the PDA returns his entire v-card, and I have to scroll to the bottom to get his phone number.
- *Quality*: I ask for his office phone number, the PDA returns his home phone number (because it's sorted "h" before "o").
- *Relation*: I ask for his office phone number, the PDA (LAN-enabled, of course) pops up a message telling me I have email from Gareth. Right thing, wrong time.
- *Manner*: I ask for his office phone number, the PDA reboots…

2.4 Examples of Breakdown and Repair

Assume that each of the scenarios above takes place within the limits of the Cooperative Principle, then each of the examples above is a violation of the respective conversational maxims. Each example is also an illustration of breakdown in the communication process.

In many of the cases, repairing the breakdown is also straight forward. This should not be surprising since we claim that breakdown and subsequent (and often automatic) repair is a natural part of having a conversation. One could easily imagine being in a scenario similar to the one described of looking for the petrol station. If we do meet passerby's who have proclivities towards violating conversational maxims, we can also imagine continuing the conversation after the breakdown occurs. For example in the first case (where the passerby violates the maxim of quantity), we could simply ignore the verbiage and continue with another question, e.g., "It's 10:00pm. Do you think they are still open?"

The exception is, of course, when the passerby (or other party in general) violates the maxim of quality. Unless the reply is blatantly impossible or untrue, we would not be aware of a breakdown, and there would be no repair.

Breakdown and repair over a longer exchange is also possible in a non-speech scenario. The examples above were chosen for simplicity. We have analyzed the card game of bridge where both the bidding sequence and the subsequent play of the cards are very much subject to the Cooperative Principle and part of the allure of the game is in trying to communicate the structure and strength of one's bridge hand to one's partner. In bridge competitions, verbal communication between partners is not allowed. This results in occasional misunderstandings (breakdowns) since the players are restricted to a limited bidding vocabulary to describe their position, but it is also interesting in how having the right shared context (bidding rules and conventions) allow the communication to occur, and breakdowns to be repaired.

2.5 Mobile Device Interaction as a Conversation

A full philosophical discussion of whether mobile device interactions are conversations is out of the scope of this paper. It is sufficient that we can show that the idea is not ludicrous and that intuitively, Grice's Cooperative Principle applies, and that the conversational maxims do capture the human user's expectations when he or she interacts with the device.

We will therefore list what, intuitively, are the characteristics of a conversation, and show that how we interact with a mobile device satisfies these characteristics.

Characteristics of a Conversation
From Grice, and with appeal to our own intuitions, we have the following characteristics:

1. Conversations take place between two or more parties. At least one of these parties should be human, conscious, and cognitively mature (this is to exclude talking in one's sleep, the babbling of small children, etc.)
2. Conversations result in an act of communication, i.e., that there is an information flow. The flow may be in one direction only.

3. Conversations must be cooperative. The parties involved want to maintain the conversation (until a logical end) and both will work towards a common goal. This distinguishes, for example, from command and control interactions.
4. The *event* of a conversation need not be at a single time and place. This allows instant messaging, SMS's and IRC chat to count as conversations. It also allows, for example, an advertising campaign to communicate a message that is distributed either or both temporally and geographically.
5. The medium of a conversation is not important. We mostly think of interactive speech as conversation, but new modes such as SMS and internet chat are becoming pervasive. It is also true that in human-human communication, there are communication channels such as body posture, movements, gestures, eye contact, etc., which complement, supplement, and can often replace human speech. Thus, this last condition is a liberal one but does try to capture how conversations proceed. It allows alternatives such as art, advertising, human-computer conversation [6, and other papers in the same workshop series], etc., to count as conversations.

There are many more (and different) characterizations of a conversation, but the ones listed are particularly relevant to the area of human-interface design.

Mobile Device Interaction

The first characteristic, that at least two parties are involved, is trivially true in our interactions with a mobile device.

The second is not so straight forward. Many of our interactions with a mobile device (e.g., calling a number on a cellular phone) do not seem to require an information flow; and would simply fall in the realm of command and control. However, in almost all cases, there is some form of a feedback and the ability to change, correct, or abort the interaction. For example, in calling a number, there is visual feedback of the numerals on the screen during the input phase and of the number in its entirety before we confirm and activate the call. The interesting thing is what comes next. In a conversational setting, there is the concept of etiquette[5] in the flow of a conversation. This includes unwritten rules for barge in, turn taking, etc. All of these are violated when we receive a phone call. If we are in the middle of another activity, we have to apologize and excuse ourselves to answer the phone. From the caller's point of view, would our actions be different if we knew the called party was in an important meeting? If yes, then it implies a breakdown in the communication act, i.e., it is consistent with the idea that calling somebody has the etiquette of a conversation. Repair is immediate in many cases; either the caller party asks "are you free to talk now?", or the called party asks that you call back later.

The third condition, which is that the two parties are in cooperation, can be recast in two directions. First, is the user cooperating with the device (or the device interface)? The answer is probably affirmative in that without cooperation (the user adapting to the interface), it is unlikely the device can be used. Second, is the device cooperating with the user, or rather does the user expect the device to cooperate with him? Reeves and Nash in [8] suggest that we unconsciously anthropomorphosize computers and other new media devices. This can be seen, for example, in how users are

[5] See [7] for example, and *politeness* is also a theme in the 3rd International Workshop in Human-Computer Conversation, 2000

frustrated when the system doesn't do what we expect. That implies that, consciously or unconsciously, we expect the device to cooperate with us.

The fourth condition, that a conversational event can be distributed over time and space is not an issue for user interaction with a device. This is trivially satisfied.

The last condition, that the medium of communication is not important, is also trivially satisfied.

We have shown that our intuitive list of conditions can be met by human interaction with mobile devices. We could also have added the condition that conversations take place over time, i.e., to try and capture the intuition that a conversation between two people is a series of exchanges and not just a single exchange. The problem comes then in, first, fixing the lower bound for when it is no longer a conversation, and second, if it is not a conversation, then what is it? To keep things simple, we have removed this condition.

Nevertheless, we should note that a human performing a task such as doing an internet search on a mobile device probably has more exchanges (interactions) than doing the equivalent task on a desktop computer. This is due primarily to the much smaller screen real estate of typical mobile devices (especially mobile phones) and the lower bandwidth channels of many of them (GPRS on a cellular device vs. LAN access on a desktop). For most mobile phones, there are the added limitations of query input mechanisms (no keyboards), manipulation mechanisms (limited scroll buttons vs. direct manipulation using a graphical interface), etc. Thus our interactions with a mobile device and a mobile phone in particular, need to be as efficient as possible.

3 Conversational Design

This section introduces the idea of conversational design. In particular, it suggests that we should use the Cooperative Principle to create interfaces which are as efficient as conversations. We look at the idea of designing for breakdown and repair, and look at a case study on the design of a mobile phone interface.

3.1 What Is Conversational Design

Conversational design simply means that we should design device interactions so that they proceed smoothly and efficiently just like conversations between two people. From our arguments earlier, we showed that human to device interaction counts as a conversation. As such, it should obey the Cooperative Principle.

The cooperative principle suggests that there are two aspects to a shared context in a conversation. The first is the conventional shared domain of the conversation. This is the one that is leveraged by interface designers currently to streamline the interaction. For example [9, 10], if we know that a user desires to retrieve an address on his or her mobile phone, we can customize the menus to restrict the number of options and the order in which the options are presented to minimize the number of button presses to accomplish the task. Note that here we are optimizing for efficiency (number of button presses) and not for, say, precision in retrieval, since the result should be binary (retrieved address or failed).

The second (unconventional) aspect of the shared context is that which gives rise to Grice's conversational maxims, and which is mostly ignored by traditional user

interface designers. This is that the two parties are sharing the same task, i.e., they are cooperating to accomplish the same task. In a non-cooperative command and control situation (party A tells party B (which could be a device)) what to do, there is no need for the device to "maintain" the conversation. All the communication in the interaction goes from A to B. In a shared task, however, both parties are committed to continuing and maintaining the conversation until its logical end. Information flow (communication) is bi-directional; both parties advance the conversation until the task is accomplished.

Related Work

The idea of designing interfaces to act (or behave) as if in a conversation is not new. R.S. Nickerson proposed such a paradigm in his seminal paper in 1976 [11] where he explored the elements of conversation including rules for turn-taking, non-verbal communication, etc. Later work in conversational design can be broadly divided into three areas. The first area is from the speech perspective, i.e., to build computer systems which interact with the users using speech recognition and text-to-speech technologies. One of the more successful versions of these efforts was the Portico voice portal from General Magic, now existing only as General Motor's OnStar service [12]. Portico carefully constructed its speech dialogues to take conversational cues and user expectations into account[6]. The second area is from the discipline of human computer interaction, where the extended interaction between human and computer is modeled as a dialogue, and hence a "conversation". Much of the work is on how the computer system will react (or feedback) to the user in response to a command, .e.g, [13], and ignores the human characteristics of a conversation. The third area continues in the direction started by Nickerson, and is epitomized by the work of Justine Cassell at MIT [14, 15]. Cassell's work on building conversational interfaces relies on knowing and understanding the various social and linguistic characteristics of conversation including turn-taking, feedback, repair, and generating and responding to verbal and non-verbal interactions.

Conversational Design and Cooperation

Our particular contribution to the notion of Conversational Design is therefore a paradigm in which the interface designer assumes that the user and the device will be *cooperating* in a conversation. As such both the user and the device have the right to expect certain behaviour of the other party, specifically, behaviour which obeys the conversational maxims.

This has ramifications in both directions. For the interface designer, he can assume that the user is cooperative (predisposed to providing the right input) and create a system which does not assume an idiot user. One very successful example of this is the motorcar. The interface designer can assume that the driver has passed a driving test and knows the difference between the accelerator and the brake, and that he or she is competent to engage the cruise control, etc. In other words, that the driver and the vehicle are cooperating to accomplish the task of transportation.

In the computing domain, Palm made that same assumption by providing only Graffiti as an input mechanism on its highly successful PDAs. They assumed that the

[6] From personal conversations with Dr. George White, formerly Director of Speech Technologies at General Magic.

user will cooperate to learn the Graffiti writing system which allowed the device to recognize written input with a much higher accuracy than previous devices. This allowed the device resources to be concentrated on what the user wants to accomplish, rather than on how to accomplish it. The various conversational maxims are epitomized in the early Palm Pilot devices. If you analyze the interface, you would be hard put to find many violations of the maxims.

Coming back to the ramifications of mutual expectations, in the other direction, the user should be able to assume that the device is cooperating to accomplish the task as well. The device should not put up roadblocks or diversions in the path of the task, e.g., confirmation dialogues when it is what the user wants to do. So, contextual menus, adapative interfaces, etc., are all methods by which a device can cooperate with the user. What is important, however, is that the implementation of the contextual menus must satisfy the conversational maxims. For example, the menu items should not have too much or too little options (maxim of quantity), should not have false options (maxim of quality), should not have irrelevant options (maxim of relation), and should not be ambiguous, inconsistent, or disordered (maxim of manner).

3.2 Designing for Breakdown and Repair

One of the characteristics of human conversation is the potential for breakdown. Breakdown is defined as a failure of the underlying communication act which is carried out in the course of a conversation. It is possible for both parties to converse and satisfy all the characteristics of a conversation and still be subject to breakdown. A simple example may be when two people are having an interesting conversation supposedly about a mutual acquaintance called Bob, but in fact it turns out that they are talking about two different Bob's.

Most of the time, however, the breakdown is either immediately obvious, or becomes obvious over time, and steps are taken by the parties involved to repair the breakdown. For example, party A mentions Bob's wife, and party B says, "Wait a minute! Bob isn't married, is he?" thus leading to further exploration and the discovery that they were discussing two different people. This is often followed by a good laugh and a switch of topic. This nullifies the breakdown, repairs the conversation, and initiates a new communication act. In almost all cases, the repair is automatic and takes no conscious effort.

In a similar way, user interfaces should share the same characteristics. A conversation works efficiently (fast and fluently) because the parties involved create a hypothetical model of a shared context [16, 17] (e.g., the idea that they are talking about the same person named Bob). Each person's model will be different depending on their perspective on the context (e.g., Bob's childhood friend would have a very different model of Bob than a current working colleague) but they necessarily cooperate to create their own model based on the elements of the conversation. The model continues to evolve as the conversation progresses. You can say that a communication act has occurred when one party's model is changed or augmented causally by the act of conversation with the other party.

One common example of breakdown and repair in interface design is the `Undo` and `Redo` buttons in many interfaces. This is in contrast to the `Confirmation Dialogue` which pops up, often in the same interface as well. If you were using a

drawing package, you do not want a confirmation dialogue every time you performed an action (draw a box, whatever). If you make a mistake, you can repair either by an `Undo` command, or by erasing and starting over. In contrast, a popular operating system, by default, pops up a confirmation dialogue each time you wish to delete a file. This happens even though the file goes to a special holding area (a `Recycle Bin`) from which an `Undo` (or `Undelete`) command can be used to repair a mistake. In which of the above cases would you consider that the system is cooperating with you to achieve your task (whether it is to draw a box or to delete a file)?

So, the idea of designing for breakdown and repair is to focus on the task and make the interface cooperate to achieve that task most efficiently. We propose that checking the interface to ensure that they do not violate the conversational maxims would help considerably in creating such an efficient interface.

4 Mobile Phone Design Case Study

In this section, we present a case study on the user interface of a mobile phone. The intention is to identify real user interface problems with the mobile phone, and to use the conversational design paradigm to analyze the problems.

The following case study follows from work done previously in [18]. The study was conducted in two phases. The first phase was to identify real problems users had with their mobile phone interfaces, and the second phase was an analysis of relevant problems using the conversational design methodology to suggest an alternative design.

4.1 Phase 1: Real Problems

We found 12 users which had personally owned and used the same phone interface. In this case, there were 3 models of the phone made by the same European manufacturer, all of which were based on the same platform (screen, buttons, controls), firmware, and hardware. There were external differences in the model including keyboard layout (one was unconventional), and only two of the models had built-in FM radios. The screen was 128x128 pixels capable of displaying 4096 colours. All of the subjects were well acquainted with the phone, being current users and owners or only recently having changed to a new phone.

Demographically, we had ten male and two female subjects, ages ranged from late 20's to mid 50's. All were local residents, but from various nationalities (Singapore, People's Republic of China, India, etc.). Seven of the subjects were technically inclined (computer science, engineering, etc.), the remainder ranged from housewife to marketing and sales.

Each user who was a current user of the phone was asked to consider his or her use of the phone over the course of a day, and at the end of the day, to email a ranked list of problems with the user interface which annoyed the user. Those who had changed to a different phone were just asked to send a ranked list of their problems. We did not ask for explications of the problems, though several of the subjects provided that as well.

The lists were normalized as follows. Each top problem was awarded 5 points, the 2nd problem 4 points, and so on down. The fifth and subsequent problems were each awarded 1 point. The median number of problems reported was 3. Here are the results in descending order of points. Only problems with at least 3 points are listed:

1. Difficult to switch on and off; button too small; have to press a long time
2. Too many button presses for actions (several examples were given)
3. Volume buttons too small
4. No voice dialing
5. Vibration mode too weak
6. Difficult to find out details (phone no.) of received calls
7. Very slow/difficult/too many button presses to delete individual SMS's
8. Cannot share information (pictures or ring tones)
9. Inconsistent buttons for answering options (examples given)
10. Cannot assign ring tones or pictures to particular numbers
11. Synchronization software (for PC) not included with phone (have to download from website)
12. No bluetooth
13. Caller name sometimes doesn't display even though caller number is in the address book
14. Alarm sometimes cannot be turned off (alarm is part of calendar function)
15. Not enough memory on the phone
16. Unusable keypad (relevant only to one of the models)
17. FM radio needs headphones plugged in to be used even with loudspeaker
18. Screen gets dirty very easily; have to take apart to clean

4.2 Phase II: Conversational Design Analysis

Most of the problems listed above are not issues of user interface design; but generically phone design. So, filtering away the poor hardware design (e.g., small buttons) or limitations (e.g., weak vibrations, lack of memory) and the features (or lack of them, e.g., voice dialing), we are left with the following:

1. Too many button presses for actions (several examples were given)
2. Difficult to find out details (phone no.) of received calls
3. Very slow/difficult/too many button presses to delete individual SMS's
4. Inconsistent buttons for answering options (examples given)

The first three problems above seem to be very similar, but have different solutions from a conversational design perspective.

Too Many Button Presses for Actions

Here, the problem is that the device has a reconfigurable button (labels change to suit the activity) which is either an action (`select` or `open`) or else `option`. However, in a few cases, the first selection in the `option` list is to open. The interface design philosophy seems to by default: provide an option list to select; if there is only one action in the option list then replace the option list with the action. E.g,, the default interface is `click option` → `click open` to open a folder.

This immediately violates the maxim of quantity. Most of the time, the user expects to open the folder; so that should be the default. Providing all the options is

giving unnecessary information. The interface could be redesigned to be `click open` → `click options` only if desired.

This could also be construed as a breadth-first (show all the actions first) vs. a depth-first (show all the objects for the most common action first) display of the menus. Conversational design suggests that in a task-driven (i.e., user) context, depth-first will be the appropriate model. Breadth-first would be appropriate only if viewing the options (e.g., exploring the interface) was the task.

Difficult to Find Details (Phone No.) of Received Calls

To find out the details of received calls, the user has to click `menu`→ `Call Register` → `select` → `Received Calls` to get to the list of calls. In contrast, to find out details of dialed (outgoing) calls, the user only has to click the `Answer` button. This is a violation of the maxim of manner; specifically that unnecessary prolixity should be avoided.

One subject who had rated this as his most important problem clarified that as he was in sales, he would be in many meetings and would receive calls he would answer but which he could not process immediately, and would promise to return the call. Hence, he wanted an easy way to list called numbers so that he could call them back. When asked, he also indicated that he used the list of dialed calls very often as well to call clients again if they were busy.

However, the phone only has one `Answer` button and no spare buttons available for a 1-click access to received calls. We examined other makes of phones to see if this problem was there as well, or if there was a solution. We found that another manufacturer merges the list of received and dialed calls together, sorted by time, which is immediately displayed when the `Answer` button is pressed. If differentiated lists are desired, then the user can retrieve them individually through the long-winded menu path. This seemed like an elegant solution. There was no ambiguity (violation of the maxim of manner again) because each call was prefaced with an icon denoting either a received or a dialed call.

Very Slow/Difficult/Too Many Button Presses to Delete Individual SMS's

The phone has a single command to delete all SMS's in the inbox via `menu` → `Messages` → `select` → `text messages` → `select` → `delete messages` → `all messages`. While this sounds rather long, it can be done in 7 clicks.

However, if you wanted to delete specific SMS's only and to leave others untouched, the sequence becomes `menu` → `Messages` → `select` → `text messages` → `inbox` → `select` (select and open a specific SMS by sender name) → `options` → `Delete` → `OK` (confirm dialogue) before returning to the inbox. This is a total of 9 clicks for one SMS, plus a further 4 clicks more for each additional SMS. The reality is worse as the phone is slow to retrieve and open each SMS.

This sequence violates the maxims of quantity (too much information returned to delete an SMS) and quality (asking for confirmation when there is no evidence that confirmation is desired). If we design for conversational efficiency (i.e., catering for breakdown and repair) then we should drop the confirmation dialogue and substitute it with an option for `undo`. This removes the violation of the maxim of quality. For the maxim of quantity, we notice that, while in the inbox, the `Answer` button is unused, and propose the following change. The current select function could be done by clicking the `Answer` button, and the current `select` button be changed to `option`,

one of which is to delete message. This makes the entire branch of this menu one level shorter.

To delete a single SMS, the sequence now becomes menu → Messages → select → text messages → inbox → options → delete which is 7 clicks (reduced by 22%) with each additional SMS requiring only 2 clicks (reduced by 50%). So, for example, to delete five specific SMS's, the original interface required 25 clicks plus the waiting time to retrieve and open the 5 messages. The proposed method requires only 15 clicks (reduced by 40%) plus no waiting time.

Inconsistent Buttons for Answering Options

The phone device in question has a built in loudspeaker. In addition to the Answer and Hang-up buttons, there are two buttons which we will call Left and Right which are located above the Answer and Hang-up buttons respectively. In the analyses above, the Select and Option buttons is always the Left button, and the Right button is reserved for the back (or return to previous screen) function.

The problem comes when answering a call. If the Answer button has *not* yet been clicked, then clicking Left/option → loudspeaker (2 clicks on the Left button) both answers the call and activates the loudspeaker. The loudspeaker option is at the very top of the list. Clicking Right/Silence turns off the ring tone but does not answer the call. Clicking Right again rejects the call.

However, if the Answer button has already been clicked, then clicking Left/option provides various other options but now with the loudspeaker option at the very bottom of the list; you have to click Right/loudspeaker to activate the loudspeaker.

This is a violation of the maxim of manner (avoid ambiguity). If it sounds confusing to describe above, then the inconsistent interface also makes it difficult to use. We checked with one of the subjects who had listed this as a problem and he said that he uses the loudspeaker function as a substitute for a hands-free earpiece while he's driving. He is however used to clicking Right/loudspeaker to activate the loudspeaker that he often clicks Right/silence→reject (2 clicks) rather than Left→loudspeaker (also 2 clicks) when he tries to answer the call on the loudspeaker, thus ending up rejecting the call.

Conversational design tells us to follow user expectations, in this case, to be consistent, so we suggest the interface should be changed so that the Left and Right button functions are swapped when the Answer button has not yet been clicked on an incoming call, i.e., clicking twice on the Right button results in answering the call and activating the loudspeaker. The problem is that there are now two ways of answering the call and there is the possibility that a call is rejected rather than answered if a finger slips since the Hang-up button is right below the Right button.

A better way may be to remove the Right/loudspeaker function to avoid the ambiguity of having two ways of answering a call, and just make Left→ loudspeaker (2 clicks) a consistent option regardless of whether Answer has been clicked or not.

5 Mobile Information Access

There are many aspects of mobile information access. This includes issues such as the following:

- how to enter information, e.g., Blackberry style keyboards, handwriting recognition, speech, etc.,
- how information is output on the mobile device, whether on a limited screen or through speech or any other modality. How to compress results, or summarize, or improve retrieval precision,
- how to navigate within the device, and within applications, and how to add new functionality onto the device. For example, if you download a java application to a java-enabled mobile device, you also import that application's user interface which is independent of the user interface of the inbuilt mobile device functionality. There are issues of efficiency and robustness, ease of use, etc.
- how to leverage the mobility aspect, leading to work in location based services and equivalent.

All of these, while valid areas of research, can be tritely summed up as *getting the right information at the right time in the right form*. Thus if I were driving a car, and I wanted to get to someplace new to me, there are many ways that information could be provided:

- if the information was a map of the area, I would have the right information, but neither at the right time (which would be before I started driving) or in the right form (it's not safe to use the map while driving),
- if the information was a set of instructions to follow on a website, I would have the right information and in the right form, but not at the right time,
- ideally, I want to be told when to turn left or right, with sufficient notice to slow down or filter to the correct side of the road, etc. And the instructions should be a pleasant clear speaking voice which I can hear without having to take my eyes off the road.

All this is not new and there are car navigation systems that attempt to do precisely what I mentioned. But currently, each application has its own interpretation of what is the right information at the right time and in the right form, and none of them take into account the situation of the user, i.e., the user has to learn or adapt to the navigation system. There are no "rules" to decide if an output or demand for input is right or wrong, or even what it means to be "right" or "wrong".

Conversational Design, following Grice, is an attempt to codify what it means to be right, and what kind of questions we need to ask to ensure that, subjectively, as in a conversation, the system provides the user the information that meets his needs, both at the explicit (lexical content) level and at the implicit (structural) level. This was illustrated in the previous section.

Unlike similar work based on conversations [e.g., 19, 20], Conversational Design does not try to implement conversational behaviour in systems. Rather it is a design paradigm that can be integrated into a design workflow to test the interface design. Here are two possible implementation strategies based on the standard ethnographic-type design approaches[7]:

[7] There are many flavours of ethnographic design, but essentially they have the same guiding principles, and key design activities. For innovation design, see [21]. More generically, see The Better Product Design website, at http://www.betterproductdesign.net/.

- Before designing an interactive information access system (mobile or otherwise), put a human intermediary between the system and the user. Assume that the intermediary is totally dedicated to that single user (as a personal mobile device would be). Observe the interaction between the intermediary and the user; analyze and replicate on the final product.
- In the course of a normal ethnographic design process, after the qualitative phase and before the quantitative phase, insert an extra step which uses Conversational Design as a test suite for the mobile device interface.

Of course, it also helps if the interface designer is familiar with Conversational Design; as a paradigm, Grice's maxims should be a constant watchdog on the design process.

6 Concluding Remarks

In this paper we have borrowed the Cooperative Principle and the conversational implicatures from Grice's *Logic and Conversation*. We have applied it to the idea of user interface design, specifically for mobile devices and introduced the idea of conversational design, and of designing to cater for breakdown and repair during efficient conversations. We presented a case study on mobile phone user interface and showed where conversational design can provide a new perspective on interface design and how such an analysis could help create better interfaces for mobile devices.

Future work for conversational design involves experimental validation. We had performed a few simple experiments based on the changes in user interface mentioned above. None of the subjects made any errors using either the original or the proposed interface. As such, no significant conclusions could be determined. If we measured the experiments by efficiency (i.e., by the number of button clicks), then the conversational design interface would be much more efficient, but since the experiments were simulated on a laptop rather than "live" on actual mobile phones, the results have no particular validity.

References

1. Guidon, R. (ed.): Cognitive Science and its Application for Human-Computer Interaction, Lawrence Erlbaum Associates, Inc., Publishers, New Jersey (1988)
2. Carroll, J. (ed.): Interfacing Thought: Cognitive Aspects of Human-Computer Interaction, MIT Press, Cambridge, MA (1987)
3. *Logic and Conversation* was first presented at the 1967 William James Lectures, and reprinted in Grice, P.: Studies in the Way of Words, Harvard University Press, Cambridge, MA (1989)
4. Searle, J.: Speech Acts: An Essay in the Philosophy of Language, Cambridge University Press, Cambridge, MA (1969)
5. Kant, I.: Critique of Pure Reason (1787), English translation by Kemp-Smith, N., MacMillan Press Ltd, (1929), and available on-line in e-text version at http://www.hkbu.edu.hk/~ppp/cpr/toc.html (1985)
6. Levy, D., Catizone, R., Battacharia, R., Krotov, A., Wilks, Y.: CONVERSE: a Conversational Companion, In Proceedings 1st International Workshop on Human-Computer Conversation, Bellagio, Italy. (1997)

7. Schmidt, A., Stuhr, T., and Gellersen, H.: Context-Phonebook - Extending Mobile Phone Applications with Context, in Dunlop, M. & Brewster, S (eds), Proceedings of Mobile HCI 2001: Third International Workshop on Human Computer Interaction with Mobile Devices, IHM-HCI 2001 Lille, France (2001)
8. Reeves, B., Nass, C.: The Media Equation, CSLI Publications, Cambridge University Press, Cambridge (1996)
9. Miyamoto, M., Makino, T., Uchiyama, T.: Menu Design for Cellular Phones, Proceedings of the Workshop on Mobile and Personal IR, ACM SIGIR Conference, Tampere (2002)
10. Jones, G.J.F., Brown, P.J.: Information Access for Context-Aware Appliances, Proceedings of ACM SIGIR Conference, Athens (2000), pp. 382-384
11. Nickerson, R. S.: On Conversational Interaction with Computers, Proceedings of User Oriented Design of Interactive Graphics Systems: Proceedings of the ACM SIGGRAPH Workshop, 1976. Reproduced in Baecker, R.M. and Buxton, W.A.S (eds), Readings in Human Computer Interaction, Morgan Kaufman, Los Altos, California (1986) pp. 681-693
12. General Magic's website, www.genmagic.com, now redirects to OnStar's website at www.myonstar.com.
13. Pérez-Quiñones, M.A., Sibert, J.L.:A Collaborative Model of Feedback in Human-Computer Interaction, Human Factors in Computing Systems Conference Proceedings CHI96, Vancouver, BC. (1996) pp. 316-323
14. Cassell, J., Bickmore, T., Campbell, L., Vilhjálmsson, H., and Yan, H.: Human Conversation as a System Framework: Designing Embodied Conversational Agents, in Cassell, J. et al. (eds.), Embodied Conversational Agents, MIT Press, Cambridge, MA (2000), pp. 29-63
15. Cassell, J.: More Than Just Another Pretty Face: Embodied Conversational Interface Agents, Communications of the ACM, Vol. 43 No. 4 (2000), pp. 70-78
16. Clark, H., Clark, E.: Psychology and Language: An introduction to psycholinguistics, Harcourt Brace Jovanovich, Inc., San Diego (1977)
17. Clark, H.: Using Language, Cambridge University Press, Cambridge (1996)
18. Loudon. G., Sacher, H., Leong, M.: Design Issues for Mobile Information Retrieval, Proceedings of the Workshop on Mobile and Personal IR, ACM SIGIR Conference, Tampere (2002)
19. Taylor, A. S: Teenage 'Phone-talk' and its Implications for Design, NordiCHI 2002, Bridging the Gap between Field studies and Design Workshop, Aarhus, Denmark (2002)
20. Berg, S., Taylor, A.S., Harper, R.: Mobile Phones for the Next Generation: Device Designs for Teenagers. CHI 2003 Conference on Human Factors in Computing Systems (2003), http://www.appliancestudio.com/publications/external/nextGenMobilesCHI.pdf
21. Ulwick, A.W: Turning Customer Input into Innovation, Harvard Business Review, Vol 80. No 1. (2002), pp 91-97

One-Handed Use as a Design Driver: Enabling Efficient Multi-channel Delivery of Mobile Applications

Mikko Nikkanen

Nokia Enterprise Solutions, P.O. Box 407, 00045 Nokia Group, Finland
mikko.ju.nikkanen@nokia.com

Abstract. This paper examines user interface issues in mobile services. Experiences from the development work of a mobile connectivity service are compared to published recommendations for small interface design. It is concluded that for a multi-channel mobile service, it is crucial to provide similar content with different access methods. By designing applications to enable easy one-handed navigation, applications can be kept simple enough to ensure that multi-channel delivery – porting to different environments, screen sizes and devices – does not require unreasonable effort.

1 Introduction

This paper examines software user interface issues in mobile communication applications, with a special emphasis on mobile office solutions. Findings from a literature review on the subject are compared to experiences from the development work of a mobile connectivity service.

Mobile telephones have been a success in the mobile market, establishing wireless phone calls and short messaging through SMS as a means of communication. In addition to using fixed-line phone calls and e-mail, more and more people are moving to mobile communication. Mobile e-mail is predicted to be one of the next big things in mobility, and signs of its business potential have already been seen in Japan where mobile Internet has made its breakthrough with tens of millions of users.

Mobile communication applications may be used with devices like mobile phones, personal digital assistants (PDAs), or pagers. Some of the devices enable wireless communication with other devices or with servers through some built-in software and over a protocol like SMS, WAP or HTML.

Typically mobile communication applications are used by people "on the go", meaning that the users do not reserve separate time to use an application, but use it as they are simultaneously doing something else. The devices the applications are used on have size, interaction, and processing power limitations, but despite the limitations, they do however offer some advantages over desktop computers, like portability and instant access to time-critical information.

Usability research on "large" interfaces like desktop computers is an established practice, and various design guidelines for this kind of applications exist. It is however not obvious that all of these widely recognized design principles apply as such for the design of small interfaces [10,11]. Only in recent years has the rise of mobile phones increased the research effort invested in small interface design, and guidelines for small interfaces have started to emerge.

F. Crestani et al. (Eds.): Mobile and Ubiquitous Info. Access Ws 2003, LNCS 2954, pp. 28–41, 2004.

1.1 Comparison of Mobile Applications and Desktop Applications

Mobile applications differ in various ways from their desktop counterparts. Along with the characteristics of mobile devices and the connecting network come certain limitations [10,14,18]:

- Low computational power, small memory and cache, and usually no mass storage devices like hard disks.
- Small display size, and a lot of variation in display dimensions.
- Restricted color display – e.g. for mobile phones, the number of color displays has only recently started to grow.
- Limited fonts and text size.
- Restricted input methods make text input slower than on a full PC keyboard.
- Often there is no pointing device for activating objects, which limits the possible user interface components and slows down object activation.
- Some devices support only vertical scrolling
- Network connections to handhelds have low bandwidths and are considerably unstable.
- Handheld operating systems do not offer the same variety of services as desktop operating systems. For instance, many operating systems do not support threads or processes for background tasks, a common technique for desktop computer applications.

Mobile applications follow a different usage paradigm than desktop applications: they are designed for a small display, have to provide short start-up and response times and are developed for gathering and presenting small pieces of information rather than processing large amounts of data [14]. Mobile users can access the mobile Internet or application at any time and anywhere, e.g. to kill short periods of time when they are not busy with something else. They play games, check their e-mail, or read the daily news headlines e.g. while they wait for an appointment.

Users also often access the applications while doing something else, either to help performing another activity, or completely unrelated to the other activity. Therefore they expect the services to be accessed easily by clicking a few buttons. Weiss [19] calls this approach “hunting” for information, as compared to “surfing” on the desktop web.

The initial position for designing a mobile application is very different from that for designing a desktop application. Design issues that specifically challenge a mobile application designer include the following [1,10,18]:

- *Information visualization*, due to the small screen.
- *Information navigation*, i.e. finding the path and actions necessary to find a piece of information on a site, and getting back when needed.
- *Interaction constraints*. For instance, requiring the use of both hands to operate a device when standing in a bus may not be a good idea. Ideally devices and services should enable easy one-handed use.
- *Context of use*. The context of use is harder to predict than with an office PC application. Since mobile devices rarely have the capabilities of stationary computers, they are not likely to be the complete solution to the user’s problems. Instead, they are more of a support in activities, where, ideally, the user’s main focus is on the activity taking place rather than on the technology supporting it.

- *Access speed.* The users often fill short gaps of unproductive time with mobile applications. Therefore the possibility to access pertinent information quickly is crucial.
- Cost. *The user may have to pay for each piece of data transferred over the network.*

Besides limitations, mobile applications provide unique opportunities for their content [18]:

- *Personalization.* The content of applications can be personalized, and content and services can be billed e.g. via mobile phones. A mobile application can, for example, allow users to purchase public transportation tickets electronically via their mobile phones.
- *Location-sensitive services.* A mobile phone can be used both independently (anywhere) and depending on its location. For instance, making phone calls is normally location-independent, but routing information for public transportation is location-dependent.
- *Timeliness of content.* Mobile service users can access content precisely when they need it, and can receive and retrieve timely information. For example, mobile services can employ alerts for last minute concert ticket sales or up-to-the-minute stock-trading information.

According to Wallace et al. [18], the most successful mobile services try to use at least two – if not all – of the above listed characteristics.

2 Guidelines for Mobile Application and Service Design

As noted above, it is clear that for the development of user-friendly small interface services and applications on mobile devices, a revision of guidelines originally meant for large displays and interfaces like PCs is necessary. An overview to existing guidelines for small interface mobile devices in the literature is presented in this section. Guidelines that apply to small mobile interfaces in general are presented, followed by a collection of guidelines more closely addressing the mobile content and navigation.

General design guidelines for mobile devices include the following:

- *Design for users on the go*. The design for mobile devices must include context and forgiveness [19], and provide time-critical information [15].
- *Enable fast use*. Two major considerations for the users of a mobile service are the cost of access and the speed of downloading content [18]. Many users are paying for mobile services by the minute, so if they cannot get the information they are looking for within a short period of time they will stop using the service [12,17].
- *Keep it simple*. The old adages about keeping a system simple stupid and about "less being more" certainly apply for mobile devices and services. For instance, the most successful PDA devices do not attempt to replace the PC, but to complement the PC use, and the use of some other traditional tools [13].
- *Provide feedback and navigation cues*. It should be obvious what the application is, and how one can navigate from the page [6,19].

- *Include self-recovering capabilities*. Even if the network goes down, the service or application need not [13,19]. There should be means to restore the values or written text, or to have them restored automatically.

Content design guidelines for mobile devices include the following:

- *Present the most important content first*. The most important content should appear at the top of the page [2,7,13,15,19].
- *Keep content compact*. It is recommended to keep the pages short [2,7,9,10,12,13].
- *Don't make the page layout complicated*. It is recommended to keep pages simple and task-oriented, possibly text only, and to avoid elements that don't add direct value to the content [2,9,12,13].
- *Use simple text elements and styles*. The elements used in text layout should be clear and simple [2,12,18,19].
- *Pay attention to page titles*. It is important that the page title elements are descriptive, since they enable bookmarking and knowing where one is [10,15,17]. The titles should however be short, preferably less than 15 characters [12,13].
- *Keep documents small*. Because there are various memory restrictions in mobile devices, the documents should be kept as small as possible [12,18].
- *Use compact link names*. Long linked text can make a page difficult to read and time consuming to scroll. It is recommended to use only one or two words as the title of the link [18,19].
- *Design clear forms*. Forms should not be too long [10]. A clear way to cancel the form filling and for going back should be provided, but attention should be paid to form resets, since on small devices, forms are laborious to refill if all values are reset by accident [18].
- *Use smart graphics*. If graphics are used at all on small devices, they should be made informative, small and simple [13].

Navigation design guidelines for mobile devices include the following:

- *Minimize steps in navigation*. With small screen devices, it is very important to design for economy of navigation [2,6,10,15,18]. Users will be frustrated by scrolling through long lists of options, filling out complex search forms, and seeing needless pages along the navigation path.
- *Selecting instead of typing*. It is recommended to consider whether it is possible to ask the user to choose from a default list using select lists, checkboxes or radio buttons rather than typing in a selection [2,12,13,17,18,19]. Alternatively one can offer a default list together with an input box.
- *Keep the navigation consistent throughout the service*. The way in which a user makes his or her way through the pages that constitute a service, interacting via links, menus and data input should be kept consistent throughout the service [12,19].
- *Design flat menus*. It is recommended to keep menus flat, because it is often difficult to form an overview of a service containing too many layers, and because a deep hierarchy makes the use more difficult [2,12,15,19].
- *Cross link*. The Back functionality is the most important way to go back. However, when users need to go back several levels, links to the starting page and subsection main pages are useful [10,12,15,19]. A simple tree design is efficient, but the deeper the navigational hierarchy gets, the more necessary it becomes to get back to the starting point, and also to other pages.

- *Provide confirmations for important actions*. Confirmations must be there for actions like changing important values or deleting items. Even though the user needs to click OK on the confirmation page, that requires much less effort than e.g. returning to a list to check if an item was really removed [10].
- *Searching should be intuitive*. Searching should be a step-by-step, logical process [15]. Once the search is performed, the results must be easy to scan, and the information should enable making good, informed choices within the results [6,10,15].

3 Experiences from Development Work

This section presents usability-related experiences from the development work of the SMS, WAP, Web and Voice accesses to corporate information provided in the Nokia One Mobile Connectivity Service. Guidelines presented in the previous section have been used to make design decisions and for evaluations during the various stages of development work with Nokia One applications. The guidelines have proven useful in development iterations.

3.1 Presentation of the Service

The Nokia One Mobile Connectivity Service is an application service that provides access to corporate e-mail, calendar and directory information from a GSM phone, a PDA, a PC or a fixed-line phone. The service enables sending and receiving e-mail, scheduling meetings and appointments and accessing corporate directories, e.g. while traveling or out of the office. It is targeted for business users. Out of the three characteristics that Wallace et al. [18] relate to successful mobile services, Nokia One applies two and leaves one out: it has personal information and timeliness, but is independent of location as it provides the same information to all locations.

Access Methods and Applications. The Nokia One service has four different access methods based on the SMS, WAP, Voice and Web protocols. Table 1 presents the applications provided with each access method at the time of writing. For Web access, large screen (PC) versions of e-mail and calendar exist, but as this paper concentrates on small interfaces, they are left out of the table.

Table 1. Nokia One applications by access method at the time of writing. Large screen e-mail and calendar are left out, as they are not within the scope here.

	Nokia One applications per access method		
Access method	E-mail	Calendar	Directory information
SMS	Yes	Yes	Yes
WAP	Yes	Yes	Yes
Web	*Under development*		
Voice	Yes	Yes	No

The applications in each access method are presented in more detail below.

The SMS access. The SMS access is based on sending short commands like "m" (for mail), "c" (for calendar), or "find" followed by a name (for the directory service) to a service number, which sends shortly a response. The responses come usually in the form of numbered lists, which then enable viewing items and navigating between them. If multiple items are presented in a list, items can be viewed by sending the number of the item (e.g. "1" for the first e-mail message, calendar event or directory service item). The items can be e-mail messages, calendar events, or items found from the directory service. Figures 1, 2, and 3 present examples of SMS commands sent to the service through SMS and responses given by the service.

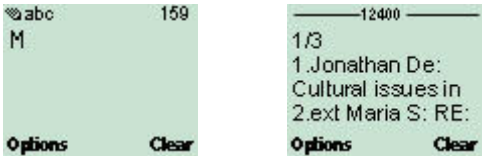


Fig. 1. An example of e-mail use through SMS. On the left, a request for new mail, and on the right, a response that shows that there are three SMS pages of message headers, out of which the first one is displayed. More of the response message is to be found by scrolling down. By sending the number of a message (e.g. "1" for the first message), the user can read the message content.

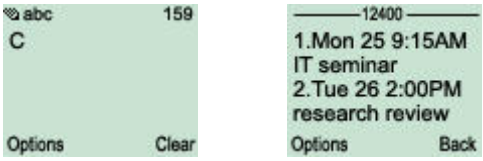


Fig. 2. An example of calendar use through SMS. On the left, a request for calendar events in the near future, and on the right, a response displaying a list of two events.

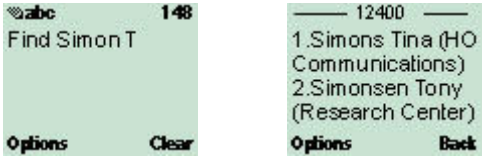


Fig. 3. An example of directory service use through SMS. On the left, a request for information on people whose names match the input, and on the right, a response displaying two people who match the criteria.

The item is split into several SMS messages in case the length of the retrieved item is more than the SMS length supported by the GSM phone in question. This is indicated by displaying a "page count" in the beginning of the message (see the response message in Fig. 1). Moving to the following page is enabled by sending an empty message, or a message containing just a space character.

Also several other e-mail functions are supported by the SMS access. Possibilities for e.g. sending, replying to, and forwarding e-mail, as well as receiving notifications of arriving messages and browsing older messages and messages in other folders besides inbox exist. The calendar application enables also e.g. browsing time periods

selected by the user, adding calendar events, and using a mobile phone's calendar together with the service. In addition to the name search, the directory service supports also searching by phone numbers and business units.

The WAP access. The WAP access provides interfaces for e-mail, calendar and corporate directory. The navigation is based on links, and is thus more intuitive to most users than the "command line" type of interaction in SMS. A starting page provides access to all applications, and to WAP settings that affect the WAP browsing. The applications are also cross linked with WAP's Options menu, so that returning to the starting page is not obligatory for moving between the applications. Moving up in the navigation hierarchy is made easier by providing links to one level up, and to the starting page at the bottom of each page.

The WAP e-mail application enables navigating within and between e-mail messages and folders, sending, replying to, and forwarding e-mail, searching and sorting messages, and viewing attachment files. E-mail in folders is divided to unread (new) and read (old) messages. If one of these links is selected, the user gets to an e-mail list. The list is divided into five message headers per WAP page. When the user selects a header of an e-mail message, the message in question is opened. If the message is long, it is divided into two or more pages. The next part of the message can be reached by selecting the link More. Examples of a WAP e-mail list and message screens are presented in Fig. 4.

Fig. 4. Examples of a WAP e-mail list and message screens. On the left, the list, and on the right, the message.

The WAP calendar application enables listing calendar events by day, week, or month, viewing, searching and editing them, creating new events, and requesting events to be sent to the phone as vCalendar notes. Calendar event lists are divided to five events per WAP page. When the user selects a header of an event, the event in question is opened. If the message is long, it is divided into two or more pages, similarly as e-mail messages. Examples of a WAP calendar event list and event detail screens are presented in Fig. 5.

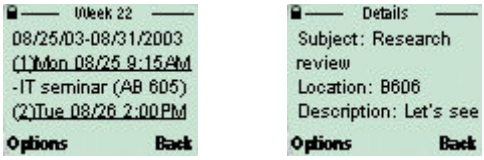


Fig. 5. Examples of a WAP calendar event list and event detail screens. On the left, the list, and on the right, the event details.

The WAP directory application enables searching contacts from the corporate directory, viewing contact details, and saving them on the phone as business cards (vCards). Examples of a WAP directory search response list and contact detail screens are presented in Fig. 6.

Fig. 6. Examples of a WAP directory search response list and contact detail screens. On the left, the list, and on the right, the contact details.

The Voice access. The voice access provides an interface to e-mail and calendar. It is used by calling a service number, where a speech synthesizer reads out the e-mail messages or calendar events that the user requests to hear. The navigation is carried out through the speech engine providing guiding prompts suggesting what the user may want to do next. The voice access includes two alternatives for commanding, speech commands that the user speaks out, and DTMF keypad commands that the user gives on a phone's keypad. The speech and DTMF interfaces provide the same commands. In addition to listening to messages or events, the voice interface enables replying to e-mail messages by recording a voice reply file (in WAV format) that is sent with the message as an attachment file.

The following is an example of a possible excerpt from e-mail use via voice access:

> *[Speech synthesizer] "… Message one from John White at Nokia dot com, subject project meeting. Say read message, next header, previous header, or goodbye."*
>
> *[User] "Read message."*
>
> *[Speech synthesizer] "Reading message number one. Press zero to interrupt at any time. Hello all, I think we should continue our…"*

The interaction with the voice access to calendar is similar, for instance:

> *[Speech synthesizer] "… The appointment is at 11 AM and it's about conference call. Say give details, browse calendar or goodbye."*
>
> *[User] "Give details."*
>
> *[Speech synthesizer] "The appointment is today at 11 AM and it lasts one hour. It's at E727 and it's about conference call. Here is a more detailed description…"*

The Web access. The Web access provides an interface to e-mail, calendar and corporate directory. At the time of writing, the small screen versions of the applications were under development, and thus they are not presented in detail here. As the large screen HTML browser applications are not within the scope of this paper, they are not presented here, although large screen (PC) versions of e-mail and calendar are fully functional.

The functionality for the small screen browser applications will resemble closely that of WAP applications, but as HTML/XHTML enables the use of more advanced formatting and use of graphical elements like icons, some views are completely redesigned to provide more value to the user. For instance, the week and month views of the calendar application benefit from the use of tables to present the time periods in a way users are used to see them in other calendar applications, and view selection between week, day and month views can apply icons that help users in quickly recognizing what the views are about.

3.2 Experiences

This section presents experiences learned in the development of the Nokia One service. Experiences have been gathered from end users through spontaneous e-mail feedback and through user studies, from customer meetings where end users have been present, and from development work. Performed user studies include 3 interview studies with 16, 3 and 4 participants respectively, and one usability test with 3 participants. The studies have involved users from three different companies.

The objectives of the user studies were to gather user needs and feedback from Nokia One users, and to gather information about current usage methods and the context of use. The intention was to get rapid, grounded input for development work. Business users with different profiles were selected from client companies based on their work profile, Nokia One use experience and their availability at the time of the studies.

In the first study, the aim was to cover the SMS and WAP accesses, and to get feedback from long-term users by conducting semi-structured interviews. 16 users from two different companies were interviewed. 8 of the users worked for one of the two companies, 8 for the other one. In one company, the users had in average 5 months of experience in using the service, while in the other company the average experience was 1.4 years. However, little information was obtained of WAP use, and thus another study was conducted to cover WAP use specifically.

In the second study, the research method applied was field usability testing, which included an interview and performing test tasks with the WAP interface to e-mail. 3 Nokia One WAP users participated in the study. All of them worked for the same company. Their experience of the Nokia One WAP e-mail use ranged from 2 weeks to 4 months.

A third study was conducted to cover the Voice access. 3 Nokia One Voice users from the same company, with 2 to 3 months of use experience, participated in semi-structured interviews.

In order to ground the design of the WAP calendar application in user data, a focus group session with 3 Nokia One SMS calendar users was held. In addition to the focus group session, one “power user” of the SMS calendar application participated in a single semi-structured interview. The focus group participants had used the SMS calendar for 2 to 3 months, and the power user for 9 months.

The most important findings from the studies are presented in this section, along with experiences from other development work. As the studies had similar objectives, and as important lessons have been learned also outside them, all the results and experiences are presented together, and not separated by study or source.

Mobile Applications in General. We have found that for a mobile service, it is beneficial to provide the same applications on various access methods and for various devices. This provides flexible access and minimizes the “gulf” between devices, while it also helps leverage demand for existing services not yet available on new devices, as once a mobile service gives access to some of the PC world's functionality, users quickly start to expect also other functionality familiar from the large display and fixed-line connection. A mobile service can nicely complement the use of a PC application, if the use of the service is fast and easy enough. For instance, a mobile service that "gets to the point" fast can reduce the need for establishing a laptop connection. If a mobile service is easy, fast and efficient to use, users can and will use it often, even during very short breaks. Users can get "hooked" to the service – in a positive sense. Easy authentication is an important part in creating a feeling of fastness and efficiency.

Flexibility of use is important. Users like it that there are several ways to use a service. Enabling users to switch between different access methods easily and efficiently, without losing the thread, is important. Moreover, multi-device support is crucial. Users, and especially large companies, don't want to buy many devices to be able to use mobile services. Once different devices are supported, tailoring the content for different devices is appreciated, as users get content optimized for their device.

Different levels of information should be available on a mobile device. As recommended in several design guidelines [2,13,15,19], the most important information should be presented first, but more detailed or less important information should also be available. The default values for all service settings must be appropriate, but low effort for user-initiated customization is appreciated by those who want to change the settings. The service interface should enable customization in the same application that is affected by the settings. If the settings are placed outside the application, the users will not change them. For an SMS interface that cannot intuitively present the settings, a credit card size quick reference card has turned out to be an efficient aid.

We have experienced that navigation is crucial for the user experience. This is not surprising, as the importance of navigation is heavily emphasized in literature [2,6,10,12,13,15,17,18,19]. Being able to tell how to get to where one wants to go and to accomplish what one wants to do, being able to tell where one is, distinguishing the device's in built features from those provided by the service, and removing unnecessary steps from navigation were noted as important.

Moreover, we have found confirmations of important actions to be valuable, and that progress indicators are appreciated when actions take long.

Application Specific Remarks. The following remarks were made about specific applications.

E-mail. Mobile e-mail users were found to primarily read their e-mail, and only secondarily take any action, like replying to the message. Many users just want to know if they have new e-mail or not. With WAP and Voice, users read longer messages than through SMS, and with WAP they write more than through SMS. Some users use the voice reply functionality in Voice e-mail.

Automatic notifications of new e-mail as SMSs are popular. Together with the fact that the mobile phone is almost always on the user, automatic notifications enable users to react to e-mail messages in real time. This enables users to choose if they want to be active in checking their e-mail themselves, or if they prefer the system to

tell them when new e-mail arrives. Automatic notifications however bring with them the need for filtering, as many business users receive huge amounts of e-mail.

Calendar. Mobile calendar users are mostly interested in quickly checking the events in the near future, especially their time and location. Viewing the current day's events is the most important function of the calendar, and viewing the current week's coming events the second most important one. Mobile calendar users appreciated the most the fact that their calendar was online, without a separate need to synchronize it.

Moving events ahead is the most often occurring action on the calendar events that are already entered in the calendar. There is little need to change the contents of an event, and the past is viewed very seldom. Most users use alarms to remind of events. The most often used alarm time is 15 minutes before the event. For appointments taking place out of office, this has to be tuned.

Getting events as calendar notes to the mobile phone is useful, as well as being able to enter events directly from the phone's calendar to the office solution's calendar.

Directory. The corporate directory users mainly use the application to get and store contact information on their mobile phones. Providing content as vCards, which enables saving directly on the mobile phone, was appreciated. Text format is also important however, since not all details can be included in a vCard. Since the directory application is fast and easy to use, some users even use it to get information about people who are in the same meeting with them.

Voice applications. For voice applications, providing the user interface in the user's native language can greatly improve the user experience even if the user has relatively good skills in a certain foreign language. In voice interaction, the guiding prompts need to get shorter as the user gets more experienced, and it must be possible to interrupt the speech synthesizer. The possibility to set the synthesizer's speed is important, along with the possibility to navigate forward and backward within a message.

One-Handed Navigation. WAP applications become almost naturally designed for easy one-handed navigation, as WAP devices are typically used with one hand only. Most WAP-enabled devices have no stylus, and thus the cursor stops on every link. This means that it is best to present the content first on the page, and the navigation tools only after it. This makes accessing content fast on devices that rely on moving from one link to the next in the order provided by the application, as opposed to presenting navigation links at the top or side of every page, as then the users would have to navigate through these links on every page. Presenting navigation tools first works well with large screen interfaces, though, since there the tools can always be visible.

Enabling easy one-handed navigation is a good design driver for all small interfaces, as it forces the interfaces to be simple and fast to use, and to provide the most important content first without any unnecessary scrolling. Navigation bars are useful on large screens, but very painful to scroll through at the top of every page on a small screen device – this is because one almost never wants to use the navigation links before having seen the actual content on the page, and thus they only slow the use down significantly. However, interfaces that enable easy use with one hand are easy to use also with two hands, e.g. with a touch screen and a stylus.

4 Discussion

A literature review on suggested guidelines for mobile devices and applications was presented, followed by experiences from the development work of a mobile connectivity service.

Designing for people who are on the move is a good design principle, as people use a mobile service during even very short breaks if it is easy and fast [15,18,19]. Similarly as Weiss [19] notes about mobile commerce on the wireless web becoming successful only after it is more convenient than making a phone call, it is to be noted that people only use mobile e-mail if it as a whole – with its response times, access speed etc. – is more convenient than waiting to get to use the PC for instance in the office.

The mobile service described in this paper presents specified sets of contextual information, e.g. only new e-mail messages instead of all messages in inbox, and provides search and sort possibilities on various levels, which has been observed to enable fast use. Approaches closely related to this kind of implementation exist in literature: for large information structures, it has been suggested to first give an overview, then to enable narrowing the scope, and to give the details only when the user requests them [16], and for Internet use on small screen devices, pre-processed summarization views that provide context information and enable view specific searching have been shown to be useful [3,4,5,6,8]. We have found that in addition to visual interfaces, this kind of approaches are useful also in voice services, like voice e-mail and calendar.

It was noticed that for a multi-channel service, it is crucial to enable easy switching between access methods, and to provide similar content across different access methods, thus enabling users to use what they have available at a time. This is in line with recommendations for e-commerce services [19]: if users cannot use what they have at hand or will lose the thread of what they are doing, they may very well not perform the action at all or move to using another channel or service. Good ways to enable use over different access methods include supporting the same simple, such as numeric-only, passwords over different mobile platforms, and making the authentication easy and fast. Providing various ways to use a service makes the service useful and motivating for a broad audience. Providing similar content across different media is challenging, though, as for example, SMS, WAP and Voice as access methods provide very different interaction design possibilities, each with their own particular limitations.

5 Conclusions

Providing similar content across different access methods is crucial for mobile communication applications. Designing to enable easy one-handed navigation is a good way to keep the applications simple, and thus scalable for different screen sizes and devices. These issues are important for multi-channel delivery on future handheld devices, as soon it will be possible to use the same content almost as such for various devices, and the device-specific modifications, when necessary, can be made for example with different style sheets. For instance, XHTML MP, the language of the future version 2.0 of WAP, can be viewed also with large screen browsers, and thus "upgrading from the small screen applications", i.e. taking the small screen applications as the starting point for the larger interfaces, will be a feasible strategy.

Tailoring only the most important views of the application to take full advantage of the specific device type's (e.g. mobile phone, Pocket PC, etc.) capabilities, while leaving the other views as simple as possible, enables high usability on various devices, without the need to make too many different designs. Enabling easy one-handed navigation is obviously an efficient design principle also e.g. when designing for the emerging phone clients that run on the Java or Symbian platforms, or when porting existing mobile applications to these new environments.

Acknowledgements

Big thanks to Virpi Roto, Jaripekka Salminen and Ingrid Schembri for review comments and suggestions for this paper, and to Heidi Wahl and Pekka Jussila, who performed user studies with me.

References

1. Björk, S., Redström, J., Ljungstrand, P. & Holmquist, L. E. (2000). Power View. Using Information Links and Information Views to Navigate and Visualize Information on Small displays. In Gellersen, H-W & Thomas, P. (Eds.). Proceedings of HUC 2000, Second International Symposium of Handheld and Ubiquitous Computing. Bristol, UK, September 2000. Springer-Verlag, pp. 46-62.
2. Buchanan G, Jones M., Thimbleby H., Farrant S. & Pazzani M. (2001). Improving mobile Internet usability. In Proceedings of the 10th International Conference on World Wide Web, 2001, pp 673-680.
3. Buyukkokten, O., Garcia-Molina, H. & Paepcke, A. (2000). Focused Web Searching with PDAs, In Proceedings of the 9th International Conference on World Wide Web, 2000, pp. 213-230.
4. Buyukkokten, O., Garcia-Molina, H., Paepcke, A. (2001). Accordion Summarization for End-Game Browsing on PDAs and Cellular Phones. In Proceedings of CHI 2001, pp. 213-220.
5. Buyukkokten, O., Garcia-Molina, H., Paepcke, A. & Winograd, T. (2000). Power Browser: Efficient Web Browsing for PDAs. In Proceedings of CHI 2000, pp. 430-437.
6. Jones M., Buchanan, G. & Thimbleby, H. (2002). Sorting Out Searching on Small Screen Devices. In Paterno, F. (Ed.), In Proceedings of the 4th International Symposium on Mobile HCI, Pisa, Italy, September 2002, LNCS 2411, pp 81-94.
7. Jones, M., Marsden, G., Mohd-Nasir, N., Boone K. & Buchanan, G. (1999). Improving Web Interaction on Small Displays. In Proceedings of the 8th International Conference on World Wide Web, 1999, 51-59.
8. Jones M., Mohd-Nasir, N. & Buchanan, G. (1999). Evaluation of WebTwig - a Site Outliner for Handheld Web Access. In Gellerson, H-W (Ed.), Proceedings of the International Symposium on Handheld and Ubiquitous Computing, 1999. LNCS 1707, pp. 343-345.
9. Kaasinen, E., Aaltonen, M., Kolari, J., Melakoski, S. & Laakko, T. (2000). Two Approaches to Bringing Internet Services to WAP devices, In Proceedings of the 9th International Conference on World Wide Web, 2000, pp. 231-246.
10. Kaikkonen, A. & Roto, V. (2003). Navigating in a Mobile XHTML Application. In Proceedings of CHI 2003, pp. 329-336.
11. Kuutti, K. (1999). Small interfaces - a blind spot of the academical HCI community? In Bullinger & Ziegler (Eds.) Human-Computer Interaction: Communication Cooperation and Application Design. Proceedings of the 8th International Conference on Human-Computer Interaction. Lawrence Erlbaum Ass. Mahwah, NJ, Vol. 1 pp. 710-714.

12. Nokia Corporation (2002). Nokia Mobile Internet Toolkit XHTML Guidelines. http://www.forum.nokia.com
13. Pearrow, M. (2002). The Wireless Usability Handbook. Charles River Media.
14. Roth, J. & Unger, C. (2000). Using Handheld Devices in Synchronous Collaborative Scenarios. In Gellersen, H-W & Thomas, P. (Eds.). Proceedings of HUC 2000, Second International Symposium of Handheld and Ubiquitous Computing. Bristol, UK, September 2000. Springer-Verlag, pp. 187-199.
15. Serco Usability Services (2000). Designing WAP Services: Usability Guidelines. http://www.usability.serco.com/research/suswapguide.pdf
16. Shneiderman, B. (1996). Advanced graphic user interfaces: elastic and tightly coupled windows. ACM Computing Surveys, 28(4es), Article no. 144, December 1996.
17. Singhal, S., Bridgman, T., Suryanarayana, L., Mauney, D., Alvinen, J., Bevis, D., Chan, J. & Hils, S. (2001). The Wireless Application Protocol. Writing applications for the mobile internet. Addison Wesley.
18. Wallace, P., Hoffmann, A., Scuka, D., Blut, Z. & Barrow, K. (2002). i-mode Developer's Guide. Addison-Wesley.
19. Weiss, S. (2002). Handheld Usability. New York: John Wiley & Sons.

Enabling Communities in Physical and Logical Context Areas as Added Value of Mobile and Ubiquitous Applications

Mario Pichler

Software Competence Center Hagenberg (SCCH)
Hauptstrasse 99, A-4232 Hagenberg, Austria
mario.pichler@scch.at

Abstract. This paper tries to address the question of how to provide added value to mobile people through mobile applications. Our suggestion for the next generation of value added mobile applications following the support for (i) communication and (ii) information access, is (iii) to provide mobile people with services that are very specific for the context area - an area representing a specific context - these people are currently in and (iv) to support the networking of individuals to form communities. We envision communities being built of humans, who come together in physical proximity, reside in equal or similar situations (e.g. people waiting on train- or tram stations), or do have the same interests. Spoken more general these communities will be built of humans, who come together in the same context area, where a context area can be constrained physically or logically. Therefore, this work introduces the notion of wireless context area networks (WCANs) as enabler of a ubiquitous information access.

1 Introduction

The current most ubiquitous mobile application[1] is mobile telephony. Looking back to the past, the objective of this application was to satisfy the *communication* needs of humans. People wanted to stay-in-touch with their families, relatives, colleagues etc. at anytime and from anywhere. This is still the objective at the present, but we nowadays can observe another goal of mobile applications too: people increasingly want a seamless *access* to required *information* like personal data or context (e.g. location) dependent information. Like mobile telephony most of the current developed mobile applications addressing this need, follow a *one-to-one communication* paradigm.

However, there still exists the question for mobile operators how to provide added value to their customers through mobile applications. This is especially true for Europe. A number of consortia address this problem and aim at developing scenarios of innovative mobile applications or to lay the foundations for building them [1–3]. Our suggestion for the next generation of value added mobile applications following the support for (i) communication and (ii) information access, is (iii) to provide mobile people

[1] By a *mobile application* we understand an application, where at least a part of the application executes on a mobile device and this device in turn communicates at least with one another stationary or mobile device via a wireless connection.

F. Crestani et al. (Eds.): Mobile and Ubiquitous Info. Access Ws 2003, LNCS 2954, pp. 42–53, 2004.

with services that are very specific for the context area these people are currently in and (iv) to support the networking of individuals to form *communities*.

We envision communities being built of humans, who come together in physical proximity, reside in equal or similar situations (e.g. people waiting on train- or tram stations), or do have the same interests. Spoken more general these communities will be built of humans, who come together in the same context area. Context areas are constrained by physical or logical boarders. Examples of context areas defined by physical boundaries are sport stadiums, railway stations, airports, or a university campus. Context areas that are constrained logically can be defined through activities or tasks people are engaged in, like waiting on a tram station, driving on the highway, waiting in front of a concert hall and the like. As opposed to the traditional one-to-one communication paradigm, the communication paradigm deployed here reaches from *one-to-many* to *many-to-many*, like it is known from chat-rooms and groupware systems [4].

Based on this motivation the problem to be addressed by this work is to create value-added mobile and ubiquitous applications through providing mobile people with services that are very specific for the context area these people are currently in, and, by supporting the networking of individuals to form communities. Further on we will refer to those context area specific services as *contextual services* and to those communities being built of humans, who come together in the same context area, as *context based communities (CBCs)*.

The rest of this document is structured as follows: Section 2 describes the vision of creating value-added contextual services for mobile people. Section 3 focuses on describing the idea of WCANs as enabler of CBC applications. Preliminary analysis results of the requirements of mobile people and the so far developed scenarios of contextual services and CBC applications are summarized within Section 4. A survey on related work is done in Section 5. The document closes with concluding remarks and an outlook to further work.

2 Wireless Context Area Networks (WCANs)

In this section we want to describe our vision for creating value-added contextual services and CBC applications.

In order to address the problem of creating added value through mobile and ubiquitous applications this work introduces the notion of *wireless context area networks (WCANs)*. Wireless context area networks are motivated by two facts:

1. Contextual information as a vital ingredient for a successful mobile and ubiquitous application
2. The boundary principle [5]

Contextual Information as Vital Ingredient

If we compare the execution context of a mobile application and the execution context of an application running on a fixed desktop computer we can observe great dynamics of mobile applications (e.g. context of movement linked with a changing location, ambient

conditions, available interfaces, bandwidth, user tasks and habits, personal interests, temporal and spatial situations etc.).

As a forerunner of future mobile applications one can observe three typical questions when calling someone on his mobile phone. These questions are:

1. "Where are you just now?" (time known, context unknown)
2. "What are you doing just now?" (time known, context unknown)
3. "Do I disturb you?" (time known, context unknown)

Looking at these questions we can see that elementary questions are not answered even in the most ubiquitous mobile application, namely mobile telephony. Thereof, we derive a great potential for future mobile and ubiquitous applications that utilize contextual information. Dey defines this usage of contextual information as *context awareness* ([6], p. 6):

> "A system is context-aware if it uses context to provide relevant information and/or services to the user, where relevancy depends on the user's task."

More than sitting in front of a desktop computer and interacting with an application or even more applications, where the execution context is mainly static, we believe that the usage of contextual information for providing relevant information and/or services to the user will be vital for a success of upcoming mobile and ubiquitous applications, where the execution context is fairly dynamic. Therefore, mobile applications have to be aware of their execution context [7], more than traditional applications.

The Boundary Principle

The second fact that lays the basis for wireless context area networks is the boundary principle of Kindberg and Fox [5], which says the following:

> "Ubicomp system designers should divide the ubicomp world into environments with *boundaries* that demarcate their content. A clear *system boundary criterion* - often, but not necessarily, related to a boundary in the physical world - should exist. A boundary should specify an environments scope but doesnt necessarily constrain interoperation."

In conjunction with Kindberg and Fox a WCAN has to be understood as an autonomous network (cf. 2.1) that is characterized by a specific context. The locality of this network should not constrain interoperability beyond the boundaries of this network. Therefore, the scope of a WCAN is not necessarily constrained to physical boarders, instead a *context area* can also be defined by logical boundaries. Examples of context areas defined by physical boundaries are sport stadiums, railway stations, airports, or a university campus. Context areas that are constrained logically can be defined through activities or tasks people are engaged in, like waiting on the tram station, driving on the highway, waiting in front of the concert hall and the like. The idea of WCANs is - according to the boundary principle - to subdivide the environment into areas that represent a specific context. "The real world consists of ubiquitous systems,

Fig. 1. Relevant Context Areas During a Workday

rather than 'the ubiquitous system'"[5]. A possible set of context areas a human might meet during a workday is illustrated in Figure 1.

Taking into consideration that the user is currently present in one of these context areas relevant information and/or services can proactively be provided to him, preferably depending on his preferences, habits and tasks (cf. 2: context awareness). This leads to the notion of WCANs as service environments, which is described in the next section.

2.1 WCANs as Service Environments

The idea of sub-dividing the environment into areas of specific context comes along with the consideration to provide services, applications and information that are very specific for that area. Closely related in this concern is the AROUND project described by José [8] and José et al. [9]. The work presented there is about supporting the association of services with location in such a way that mobile applications can select services relevant for their location. A service-based architecture that supports location-based service selection is presented. A central element of this architecture is a *scope model* that assumes for each service to have an associated scope that specifies the physical range in which it should be available. This is very similar to our notion of context areas. Nevertheless, the primary context information used within the AROUND project is location, while we also consider context areas that are constrained logically, e.g. through activities or tasks people are engaged in.

A car driver A for instance can relay information about a traffic jam he recently passed by. Drivers of oncoming traffic would come up with this traffic jam. For those drivers the information driver A provides will be valuable in order to avoid coming up with the traffic jam. In this scenario information arises in the context area road traffic and is also consumed in this context area. This is what is meant by context areas as *autonomous networks* mentioned above. What happens here is a form of context sensitive ad hoc communication, as described by Yau et al. [10]. Further on, the participants of the road traffic in this scenario can be viewed as being parts of a context based community.

Another example is an indoor tennis court, where visitors are interested in all topics concerned with tennis. For instance information about persons, who played on this court before, results of previous games, which events and tournaments are planned for the future etc.

Looking at these examples we can see that a service, which is of value in various context areas, is to *relay information to other persons*. This is just one service, a number of further services that may be of interest in various context areas can be considered. Nevertheless, there will also be services that are only of value in a very specific context. Therefore, single context areas can be understood as service environments.

Two examples of service environments will be explained in the following:

1. Interactive conference
2. Mobile passenger information and ticketing

Interactive Conference

The *interactive conference service environment* consists of services, which may be interesting for participants of a scientific conference. The conceptual model of a service space at the conference site can be imagined as shown in Figure 2.

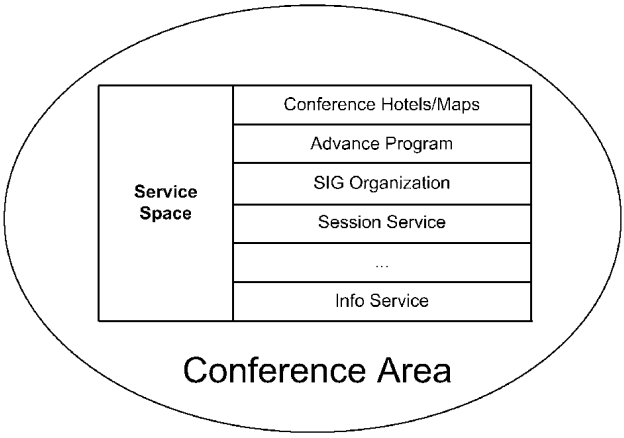


Fig. 2. Service Space at Conference Site

At the time a conference participant enters the conference area the start page of the interactive conference service environment appears on the display of his mobile device (Figure 3).

As every fixed (conference site infrastructure) and mobile device (participants mobile devices) can act both as service consumer and service provider, the available space of services at the conference site can grow, if participants also act as service provider. The result is a common shared service space. In the case of the interactive conference scenario the conference backbone infrastructure will primarily act as service provider.

Fig. 3. Interactive Conference: Available Services

Mobile Passenger Information and Ticketing

Another example of a specific service environment is the *mobile passenger information and ticketing service space* for public transport. Available services for passengers are shown in Figure 4.

Considering the following situation the *Query Route* service might be interesting for passengers:

Gerhard is sitting in the tram. Suddenly he bears in mind that he has promised his son to show him a photo of an airplane taking off when coming home. Thus, he decides not to drive home, but instead to drive to the airport to take the promised photo. The question for Gerhard now is, how to come to the airport. He uses his mobile companion to query the route to the airport as shown in Figure 5. Using the "Ticket Info" service he can check if his current ticket is valid for the trip to the airport. If not, he can use the "Get Ticket" service to order the required ticket.

2.2 Humans as Travellers between Context Areas

It is natural that humans are travelling between the two above-mentioned context areas or the context areas as shown in Figure 1. Furthermore, people are using the services provided within the single context areas. Through the usage of the mobile device - the user's personal information appliance - in distinct context areas the boundaries between these areas are softened. Interoperation among various context areas is thus possible (cf. 2: boundary principle).

Fig. 4. Mobile Passenger Information and Ticketing: Available Services

Additionally, the services on the mobile devices of humans, who are concurrently within the same context area, can interact, and thus enabling context sensitive ad hoc collaboration of those humans. This can be exchange of information, music sharing, or to get in contact with persons, who have similar interests etc. What happens then is a form of community building, which is described in the following section.

3 WCANs Enabling Context Based Communities

In order to address the current widespread problem of creating added-value through mobile and ubiquitous applications we suggest to support the networking of individuals to form communities. This objective of mobile and ubiquitous applications, together with the provision of contextual services, follows the goals of (i) supporting humans to stay-in-touch with their families, relatives, colleagues etc. and (ii) supporting a seamless access to required information like personal data or context dependent information.

We see added-value of mobile and ubiquitous applications in supporting community building of persons, who reside in the same context area. Now, where does this aim come from? Therefore, let's have a look at closely related work and into the history of social interaction among persons.

In his theory of proxemics Edward T. Hall [11–13] established the idea that there are distinct levels of proximity in interpersonal communication:

Fig. 5. Query Route Service

- *Intimate space* – the closest "bubble" of space surrounding a person. Entry into this space is acceptable only for the closest friends and intimates.
- *Personal space* – is used for conversations and among friends and family members.
- *Social space* – the space in which people feel comfortable conducting routine social interactions with acquaintances as well as strangers.
- *Public space* – the area of space beyond which people will perceive interactions as impersonal and relatively anonymous.

Kortuem [14] builds on Hall's concept of social space. The augmentation of social interactions and social space is the key mechanism of his alternative model of wearable computing, called social wearable computing. More than augmenting humans sensory capabilities by wearable computers during face-to-face social interactions as done by Kortuem, we believe that there is a need to support interaction of people, who reside within the same context area (cf. 2). Therefore, we suggest to extend Hall's spatial zones by the notion of *contextual space*, where from now on we use contextual space as a synonym for context area. We place contextual space between social space and public space, as interaction with people in the same context area is a form of social interaction, and, as this interaction is based on sharing the same interests for instance, it is not sensed that anonymous. In fact sharing the same interests or residing in the same situation as others lays the basis for humans feeling as part of a community - a context based community.

Examples for such communities can be people visiting a sports event like a Formula 1 race; people residing in concert area and waiting for the musicians starting their performance; people waiting on (different) train- and tramstations; people driving on a highway etc. A famous example for such a community application is the Hocman prototype [15], which supports social interactions among motorcyclists.

We see wireless context area networks as enabler for building applications supporting the interaction of humans residing in the same context area. However, one interesting challenge will be to determine the "proximity" - context proximity - between people, who meet in a logical context area.

4 Common Patterns

In this section we will briefly describe the requirements of mobile people as well as the characteristics that were identified when analyzing scenarios of context areas and the provided contextual services. This is just a preliminary result as the development and analysis of scenarios is still ongoing work and the above-mentioned examples of context areas represent just a sub-range of the already developed scenarios. Requirements of mobile people:

- Information exchange/dissemination with/to other parties (people or services)
- Information capturing
- Accessing context dependent, timely information
- Time independent usage and provision of information and services
- Supporting community building (i.e. to get acquainted with persons of same interests)
- Usage of services to shorten waiting- or idle times (e.g. games)
- Guide-me services
- Mobile access to discussion forums (but not that high priority)

Among the issues that can primarily be seen as common characteristics of contextual services are:

- Context discovery / information discovery / service discovery
- Service deployment (e.g. games, individual services)
- Frequently (and often unpredictable) context changes

These lists will be extended as analysis of scenarios continues.

5 Related Work

As presented in the paper the work of José [8] and José et al. [9] is closely related in terms of providing services that are relevant within a specific area. The work of Kortuem [14] is closely related to the proposed work in terms of supporting interactions among mobile people. Nevertheless, the concept proposed within this work aims at supporting interaction of people residing in the same context area, rather than augmenting face-to-face interactions as done by Kortuem. Other related (conceptual) work will be Gaia [16].

Platforms and frameworks that facilitate the application development for mobile environments are important for this work. Some existing ones that provide support for issues that are inherent for ubiquitous systems are the following: the Context Toolkit by Dey and Abowd [17] for instance focuses on the development of context aware applications. Proem by Kortuem et al. [18] and XMIDDLE by Mascolo et al. [19] provide computing platforms for mobile ad hoc applications, whereby the latter especially focuses on synchronization mechanisms of data replicated on several mobile devices. LIME by Murphy et al. [20] is targeted towards physical mobility of users and their mobile devices and logical mobility of code.

Dividing the environment into context areas (or spaces) to provide or to use information and services that are specific for that area can also lead to the notion to use space-based technologies for that reason. Therefore, the suitability of space-based technologies as platform for mobile context sensitive services has to be evaluated. The above-mentioned LIME is one of those space-based technologies. Further examples for that technology are CORSO [21], JavaSpaces [22], TSpaces [23], Limbo [24], and GigaSpaces [25].

Ongoing work in the service discovery domain incorporating the characteristics of mobile environments is also of interest for the described work. Konark [26] is a service discovery and delivery protocol designed specifically for ad hoc, peer to peer networks, and targeted towards device independent services in general and m-Commerce oriented software services in particular. Handorean and Roman [27] are describing a service model built on top of LIME for service provision in ad hoc networks. JDSP (JESA Service Discovery Protocol) [28] also aims at efficient service discovery in ad hoc networks. And Lee and Helal [29] describe the use of context attributes for dynamic service discovery. Questions that have to be answered in order to facilitate service location, provision, and access in mobile and ubiquitous environments include: How can a mobile device detect a remote service in mobile and ubiquitous environments? How can a mobile device access a remote service in mobile and ubiquitous environments? How can a device advertise its desire to provide services to the rest of the members residing in the same context area? Project JXTA [30] can be an answer to these questions.

6 Conclusion and Further Work

In this paper we have presented our vision of creating value-added mobile and ubiquitous applications through the provision of contextual services and the support of networking of individuals to form communities. Based on the concept of wireless context area networks areas representing a specific context act as service environments. These context areas or service environments can be constrained physically or logically. Humans, as travellers between context areas, use their personal information appliances to access services in the respective context area. Application scenarios of contextual services and CBC applications as well as some of the requirements of mobile people and for software supporting the development of contextual services and CBC applications were presented.

As future work we will continue developing scenarios of contextual services and CBC applications in various context areas and analyzing them in order to identify com-

mon aspects and patterns that act as requirements for software support to realize the scenarios. Based on these requirements we will perform detailed investigations of platforms and frameworks that seem to be promising for the realization of the scenarios. The goal is to create a framework for the development of WCANs as enabler of contextual services and CBC applications. Thereby, we will base on identified promising approaches. Through prototypical implementations of some of the developed scenarios the developed concept and WCAN framework shall be evaluated.

Acknowledgements

The author acknowledges support of the K*plus* Competence Center Program which is funded by the Austrian Government, the Province of Upper Austria, and the Johannes Kepler University Linz. The author would also like to thank Prof. Gabriele Kotsis and Dr. Wieland Schwinger for valuable comments on preliminary versions of this document.

References

1. MB-net: Network of excellence in mobile business applications and services. http://www.mbnet-forum.org/ (2003) Last visited: July 2003.
2. UMTS: UMTS Forum. http://www.umts-forum.org/ (2003) Last visited: July 2003.
3. WWRF: Wireless World Research Forum. The book of visions 2001 - visions of the wireless world (2001)
4. Johansen, R.: Groupware: Computer Support for Business Teams. Free Press, New York (1988)
5. Kindberg, T., Fox, A.: System software for ubiquitous computing. IEEE Pervasive Computing **1** (2002) 70–81
6. Dey, A.K.: Providing Architectural Support for Building Context-Aware Applications. PhD thesis, College of Computing, Georgia Institute of Technology (2000)
7. Capra, L., Emmerich, W., Mascolo, C.: Middleware for mobile computing. UCL Research Note RN/30/01 (2001)
8. José, R.: An Open Architecture for Location Based Services in Heterogeneous Mobile Environments. PhD thesis, Computing Department, Lancaster University, England (2001)
9. José, R., Moreira, A., Meneses, F.: An open architecture for developing mobile location-based applications over the Internet. In: Proc. of 6th IEEE Symposium on Computers and Communications, Hammamet, Tunisia (2001)
10. Yau, S., Karim, F., Wang, Y., Wang, B., Gupta, S.: Reconfigurable context-sensitive middleware for pervasive computing. IEEE Pervasive Computing **1** (2002) 33–40
11. Hall, E.T.: The Silent Language. Anchor Press/Doubleday, New York (1959)
12. Hall, E.T.: Proxemics: The Study of Man's Spatial Relations. International University Press, Connecticut (1962)
13. Hall, E.T.: The Hidden Dimension. Anchor Press/Doubleday, New York (1966)
14. Kortuem, G.: A Methodology and Software Platform for Building Wearable Communities. PhD thesis, Department of Computer and Information Science, University of Oregon (2002)
15. Esbjörnsson, M., Juhlin, O., Östergren, M.: The hocman prototype - fast motor bikers and ad hoc networking. In: Proc. of MUM 2002, Oulu, Finland (2002)
16. Román, M., Hess, C., Cerqueira, R., Ranganathan, A., Campbell, R., Nahrstedt, K.: A middleware infrastructure for active spaces. IEEE Pervasive Computing **1** (2002) 74–83

17. Dey, A.K., Abowd, G.D.: The context toolkit: Aiding the development of context-aware applications. In: Proc. of Workshop on Software Engineering for Wearable and Pervasive Computing, Limerick, Ireland (2000)
18. Kortuem, G., Schneider, J., Preuitt, D., Thompson, T., Fickas, S., Segall, Z.: When peer-to-peer comes face-to-face: Collaborative peer-to-peer computing in mobile ad-hoc networks. In: Proc. 2001 International Conference on Peer-to-Peer Computing (P2P2001), Linköpings, Sweden (2001) 27–29
19. Mascolo, C., Capra, L., Zachariadis, S., Emmerich, W.: XMIDDLE: A data-sharing middleware for mobile computing. Personal and Wireless Communications Journal **21** (2002)
20. Murphy, A., Picco, G., Roman, G.C.: Lime: A middleware for physical and logical mobility. In: Proc. of the 21st International Conference on Distributed Computing Systems (ICDCS-21), Phoenix, AZ, USA (2001) 524–233
21. CORSO: Corso Shared Object Space Technology. http://www.tecco.at/en/eTechnology.html (2003) Last visited: July 2003.
22. JavaSpaces: JavaSpaces technology. http://java.sun.com/products/javaspaces/ (2003) Last visited: July 2003.
23. TSpaces: Intelligent connectionware. http://www.almaden.ibm.com/cs/tspaces/ (2003) Last visited: July 2003.
24. Davies, N., Wade, S., Friday, A., Blair, G.: Limbo: A tuple space based platform for adaptive mobile applications. In: Proc. of the International Conference on Open Distributed Processing/Distributed Platforms (ICODP/ICDP'97), Toronto, Canada (1997)
25. GigaSpaces: http://www.j-spaces.com/ (2003) Last visited: July 2003.
26. Helal, A., Desai, N., Verma, V.: Konark - a service discovery and delivery protocol for ad-hoc networks. In: Proc. of the Third IEEE Conference on Wireless Communication Networks (WCNC), New Orleans (2003)
27. Handorean, R., Roman, G.C.: Service provision in ad hoc networks. In: Proc. of the 5th International Conference COORDINATION 2002. Number 2315 in Lecture Notes in Computer Science, Springer-Verlag (2002) 207–219
28. Preusz, S.: JESA service discovery protocol: Efficient service discovery in ad-hoc networks. University of Rostock, Dept. of Computer Science, Chair for Information and Communication Services (2001)
29. Lee, C., Helal, A.: Context attributes: An approach to enable context-awareness for service discovery. In: Proc. of the Third IEEE/IPSJ Symposium on Applications and the Internet, Orlando, Florida (2003)
30. JXTA: Project JXTA. http://www.jxta.org (2003) Last visited: July 2003.

Accessing Web Educational Resources from Mobile Wireless Devices: The Knowledge Sea Approach

Peter Brusilovsky[1] and Riccardo Rizzo[2]

[1] School of Information Sciences, University of Pittsburgh, Pittsburgh PA 15260, USA
peterb@sis.pitt.edu
[2] Istituto di Calcolo e Reti ad Alte Prestazioni, Italian National Research Council, 90146 Palermo, Italy
ricrizzo@pa.icar.cnr.it

Abstract. This paper addresses the issue of finding and accessing online educational resources from mobile wireless devices. Accomplishing this task with a regular Web search-and-browse interface demands good interface skills, a large screen, and fast Internet connection. Searching for the proper interface to access multiple resources from a mobile computer we have selected an approach based on self-organized hypertext maps. This paper presents our approach and its implementation in the Knowledge Sea system. It also discusses related research efforts and reports the evaluation of our approach in the context of a real classroom.

1 Introduction

The modern Web is the largest treasury of educational resource has ever been available. It's customary nowadays for college professors to recommend a set of useful Web resources for any lecture and to encourage the students to find more relevant resources themselves. It is currently anticipated that the students access these resources from computers at home or at the university labs. This model contradicts with the popular "anytime, anywhere" slogan of Web-enhanced education. While the Web is always "present" the students can't yet access it from anywhere. It is certainly a restriction to an educational flexibility - like a requirement to read a textbook always at home or in class, but not outside, in a café, or while riding a bus. The use of mobile wireless handheld devices potentially allows the students to access educational resources really "anywhere", however, a number of steps have to be preformed to make it really happen. The problem here is not simply technical. Supplying all students with wireless handheld computers and providing a wireless connection in some large area is an important step towards the solution, but is not the solution on itself. The problem is that almost all expository and objective Web-based educational resources have been designed for relatively large screens and relatively high bandwidth. Special research efforts have to be invested to develop educational resources that are suitable for use with handheld devices or to adapt existing resources for the new platform.

The goal of our group at the Department of Information Science and Telecommunication at the University of Pittsburgh is to explore different ways in which mobile wireless devices can be used for college education. Having both information science

F. Crestani et al. (Eds.): Mobile and Ubiquitous Info. Access Ws 2003, LNCS 2954, pp. 54–66, 2004.

and telecommunication faculty under the same roof, a school-wide wireless network, and dozens of wireless handheld devices, we have very nice settings for developing new systems and exploring them in the classroom. The focus of one of our research project is the access to multiple educational Web resources from mobile devices. As we have mentioned above, a variety of Web resources is available for any course. The resources often overlap and complement each other, so multiple resources have to be used for studying almost any topic. For example, in our "Programming and Data Structures" course based on C language, we recommend the students to use several free C language tutorials and other on-line resources (such as C language FAQ). Different tutorials cover different topics with different details and also do it using different styles. Altogether, they well complement the course textbook and enable students with different levels of knowledge or different learning styles to get a better comprehension of the subject.

Unfortunately, it is hard to expect a teacher to provide a list of relevant readings for a lecture from more than one source (that is usually a textbook). What a teacher usually can do is provide the links to the home pages of all these tutorials hoping that the students will be able to locate tutorial fragments that are relevant for each lecture. Unfortunately, as we have found in the course of our research, the students almost never do it. Even on a desktop computer finding relevant reading fragments buried deeply under the tutorial home pages and distributed over several tutorials is a challenging activity that requires good navigation skills, a large screen, and a fast Internet connection (Figure 1). Mobile computers with small screens and slower connection need another interface to accomplish the same task.

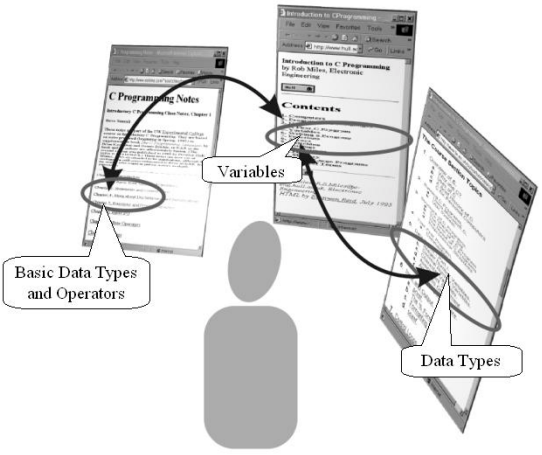


Fig. 1. Studying from multiple on-line resources

Searching for the proper interface to access multiple resources on a mobile computer we have considered several options and finally selected an approach based on self-organized hypertext maps. This paper presents our approach and its implementation, discusses related works, and reports the results of using our approach in the context of a real classroom.

2 Navigating Multiple Educational Resources with a Self-organized Map

The core of our approach to navigating educational resources is a self-organized hyperspace map. Hyperspace maps are generally regarded as one of the most important tools in hypertext navigation. A map can provide concise navigation and orientation support for a relatively large hyperspace. Traditionally hypertext maps are designed manually by hypertext authors. This manual approach is totally inappropriate for a heterogeneous distributed Web hyperspace that has no single author. However, there are a number of known approaches to automated or automatic building of hypertext maps. The approach that we have chosen is based on the Self-Organizing Map (SOM), an artificial neural network that builds a two dimensional representation of the inputs. SOM is a very attractive technology for developing compact maps for a large hyperspace since it builds a map representing only the neighborhood relationship between the objects. In these maps only the relative distance between objects is reported and any other information is lost.

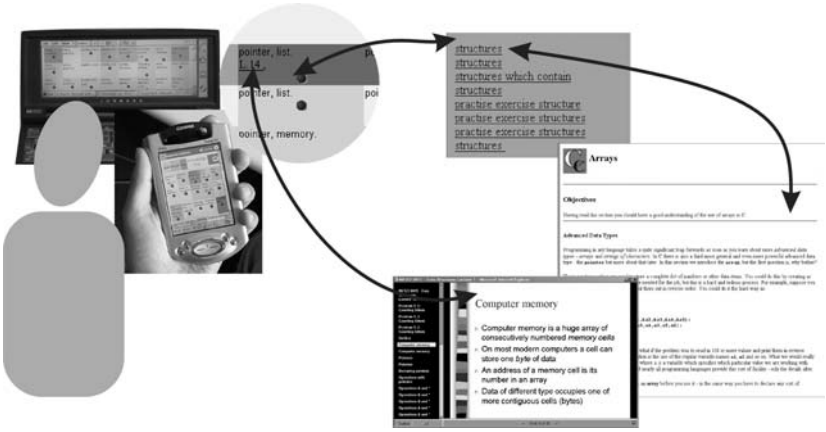


Fig. 2. A session of work with the Knowledge Sea system

A two-dimensional map of educational resources developed with SOM technology is the core of our Knowledge Sea system for map-based access to multiple educational resources (Figure 2). Knowledge Sea was designed to support a typical university class on C programming. In this context, the goal of the students is to find the most helpful material as a part of readings assigned for every lecture in the course. The most easily available Web educational resources are multiple hypertextual C tutorials. In this context, the goal of the Knowledge Sea system is to help the user navigate from lectures to relevant tutorial pages and between them.

The users see the Knowledge Sea map as an 8-by-8 table (Figure 2). Each cell of the map is used to group together a set of educational resources. The map is organized in such a way that resources (web pages) that are semantically related are close to

each other on the map. Resources located in the same cell are considered very similar, resources located in directly connected cells are reasonably similar, and so on.

Each cell displays a set of keywords that helps the user locate the relevant section on the map. It also displays links to “critical” resources located in the cell. By critical resources we mean resources that are known to the user and that can serve as origin points for map-based navigation. For example, for lecture-to-tutorial navigation the critical resources are lectures and lecture slides known to the users (see two map cells in the enlarged section on the upper left part of Figure 2). The cell color indicates the "depth of the information sea" – the number of resource pages lying "under the surface" of the cell. Following the "information sea" metaphor we use several shades of blue in the same way they are used on geographic maps to indicate depth. For example, light blue indicates "shallow" cells with just a few resources underneath while deep blue indicates "deep cells" that have the largest number of resources. The resources "under" the cell can be observed by "diving". A click on the red dot opens the cell content window (right on Figure 2) that provides a list of links to all tutorial pages assembled in the cell. A click on any of these links will open a resource-browsing window with the selected relevant page from one of the tutorials. This page is loaded "as is" from its original URL. A user can read this page and use it as a starting point to navigate an area of interest in the tutorial.

The map serves as a mediator to help the user navigate from critical resources to related resources. These links to critical resources work as landmarks on the map, and, together with the keywords, give an idea of the material organized by the map. If the user is interested in finding some additional information on the topic of lecture 14 (devoted to pointers), the first place to look is the cell where the material of this lecture is located (shown as L14 link on the enlarged section of Figure 2). If the user is looking for the material that can enhance the topic of the lecture in some particular direction, the cells that are close to the original cell provide several possible directions to deviate. For example the material related to memory usage in the context of pointers is located underneath of the cell with L14 mark. The links to other critical resources shown on the map can help selecting the right direction for deviation. For example, a good place to look for a material that can connect the content of lectures 14 and 15 is a cell between cells where L14 and L15 links are shown. The map helps the user to select the page related to the original in the “right” sense.

3 The Mechanism of the Self-organizing Map

The Knowledge Sea map is automatically built by an artificial neural network. Artificial neural networks are formed by a set of interconnected simple processing units that can “learn” to process the input data by using a supervised learning algorithm or using self-organization. The neural network used to build the document map is the Self-Organizing Map (SOM, sometimes referred as Kohonen map) [4]. In this neural network the units are organized in a sort of elastic lattice, usually two-dimensional, placed in the input space (in our case the hyperspace spanned by the set of documents). During the learning phase this lattice “moves” towards the input points. This “movement” becomes slower and at the end of the learning stage the network is “frozen” in the input space.

After the learning stage the units of the map can be labeled using the input vectors and the map can be visualized as a two-dimensional surface with the inputs vectors distributed on it. Input vectors that are near each other in the input space are near each other on the map (Figure 3).

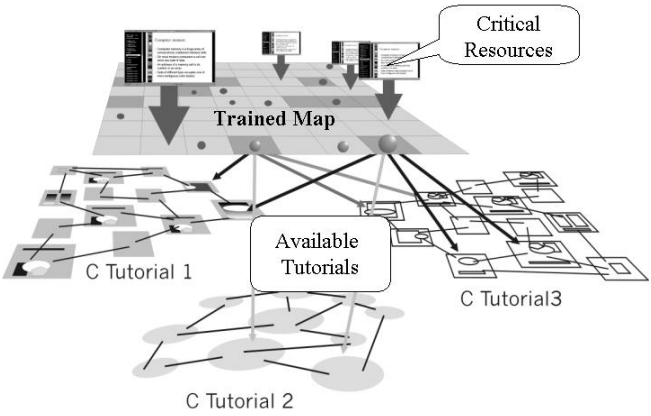


Fig. 3. The organzation of different input and the structure of the map

3.1 SOM Algorithm

The SOM algorithm is explained below referring to a N_1 x N_2 rectangular grid (the extension to a hexagonal grid that does not favor horizontal and vertical directions is straightforward).

Each unit $i = \{1,2,...,N_1 \times N_2\}$ has a weight vector:

$$w_i(t) \in \Re^n \tag{1}$$

where *i* defines the position of the unit inside the array. The SOM model also contains the *h(c,i,t)* function that defines the "stiffness" of the elastic surface to be fitted to the data points. This function depends on the relative position of the two units *c* and *i* on the network grid and contains some parameters that are updated during the learning stage.

Suppose we have a set of *m* training vectors $\mathbf{X} = \{\mathbf{x}_k, k=1,2,... m\}$, with $\mathbf{x}_n \in \Re^n$. During the learning stage these vectors are presented to the network. After a sufficient number of learning steps the weight of each neural unit will specify a codebook vector for the input distribution, these codebook vectors will sample the input space.

The unit weights (codebook vectors) will be organized such that topologically close units of the grid are sensitive to inputs that are similar. The learning algorithm is below:

1. Initialize the unit weights w_i, the discrete time *t=0*, and the parameters of the function *h(c,i,t)*;
2. Present the input vector $\mathbf{x} \in X$;
3. Select the best matching unit *c* (b.m.u.) as:

$$\|\mathbf{x}-\mathbf{w}_c\| = \min_{i=1,2,...N_1 x N_2} \{\|\mathbf{x}-\mathbf{w}_i\|\}$$

4. Update the network weights

$$\mathbf{w}_i(t+1) = \mathbf{w}_i(t) + h(c,i,t)[\mathbf{x}-\mathbf{w}_i(t)]$$
$$i = 1,2,..., N_1 \times N_2$$

5. Update the parameters of the function *h(c,i,t)*
6. Increment the discrete time *t*
7. If $t =< t_{max}$ then go to step 2.

The learning function is indicated in step 4. In this step the *b.m.u* and the nodes that are close to the *b.m.u* in the array will activate and update their weight vectors moving towards the input vector (Figure 4).

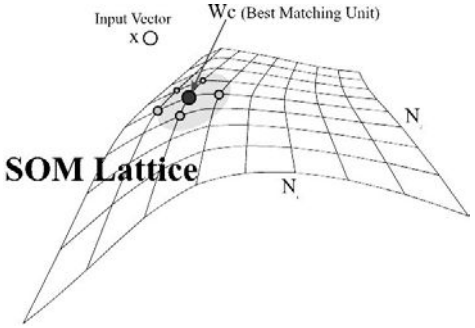


Fig. 4. A representation of the SOM learning algorithm. The gray area is the neighborhood of the best matching unit

The amount of movement is modulated by the *h(c,i,t)*, the so-called neighborhoods function, a smoothing kernel defined over the lattice points. For the convergence of the algorithm it is necessary that:

$$\lim_{t\to\infty} h(c,i,t) = 0 \tag{2}$$

The *h(c,i,t)* takes the max value on the *b.m.u* and decays on the units that are distant from it. In the literature two functions are often used for the *h(c,i,t)*: the simpler one refers to a square neighborhood set of array point around the *b.m.u*. as shown on Figure 5. If their indexes set is denoted $N_c(t)$ then the function is defined as:

$$h(c,i,t) = \begin{cases} \alpha(t) & if \quad i \in N_c \\ 0 & otherwise \end{cases} \tag{3}$$

Where:

- $N_c(t)$ is a function of time and is shrinking during the time
- $\alpha(t)$ is defined as learning rate and is monotonically decreasing during the time.

The other widely applied smoothing neighborhood kernel is written in terms of the Gaussian function.

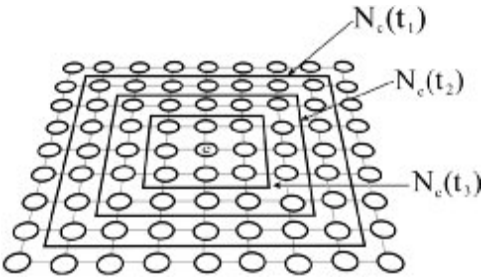


Fig. 5. *Nc(t)* gives the set of nodes that are considered the neighborhood of the node *c*. $t_1<t_2<t_3$

3.2 Parameter Values

If the SOM network is not very large (a few hundred nodes at most) the selection of parameter values is not very crucial. As a "rule of thumb", it is possible to start with a fairly wide $N_c(0)$, even more than half the diameter of the network, and letting it shrink with time. An accurate function of time is not very important for the learning rate $\alpha(t)$ - it can be linear, exponential or inversely proportional to *t*. The accuracy of the learning depends on the number of steps in the learning phase: it should be at least 500 times the number of the network units. There is no theoretical way to determine the amplitude of the parameters that have been chose by tentative. By empirical observation the learning stage is divided into two phases of very different length:

- **ordering phase:** in this phase the network organizes the weights of the units in order to roughly approximate the input distribution. The parameters should have the following initial values: α_0 near to the unit (e.g. 0.8) and the smoothing kernel should be large enough to take almost the whole network when the weights are changed.
- **convergence phase:** the convergence phase is the refining phase in which the vectors reach their final positions. It is 8 or 9 times longer than the ordering phase and during this phase there are no large variations of the unit weights. The parameter α_0 should be small (0.2 or less) and constant or slightly decreasing. The smoothing kernel initial value should be narrow enough to change just a few units or only the *b.m.u.*

A rough way to evaluate the quality of the result obtained after the learning stage is to calculate for each input vector $\mathbf{x}_k \in X$ the *b.m.u. c* and to evaluate the quantity *A* defined as:

$$A = \frac{1}{m}\sum_{k=1}^{m} \left\|\mathbf{x}_k - \mathbf{w}_c\right\| \tag{4}$$

It is convenient to calculate several maps with different initial values and to choose the best result.

4 The Implementation of the System

The neural network is just one part of the developed system. In order to prepare the learning set of the SOM map the HTML documents were preprocessed in order to

remove "noise" (copyright notes, author name, HTML tags, C code, and so on) and encoded using TF*IDF approach. With TF*IDF, each document is represented by a vector where each component corresponds to a different word. The value of the component is proportional to the occurrence of the word in the document and inversely proportional to its occurrence in the whole set of documents [8]. The calculation of the TF*IDF often includes a normalization factor to obtain a representation vector that is independent from the text length.

The document set used for the learning phase of the SOM network included 210 HTML files from three Web-based tutorials on C programming language. The whole set of pages contained 4249 different words. They were represented by the 500 most common words after the removal of stopwords. All document representations were collected in a file and submitted to the neural network simulator. At the end of the learning phase each cell of the map collected conceptually similar pages from various tutorials.

The output of the neural network simulator was used to build a set of HTML pages that the user accesses interacting with the system. All pages were designed to fit the screen of a handheld PC such as the HP Jornada. The home page of the system contains only the map visualized as an HTML table. Each cell of the table corresponds to a neural unit of the map and is labeled by representative keywords.

The system is also scalable: it is possible to add new resources to the system simply by building the TF*IDF representation and submitting the vectors to the Self-Organizing Map. The neural network will classify the new vectors into the right cells.

5 A Challenge of a Narrow Screen

In order to choose which map geometry will fit small computer devices, several different maps were trained using different approaches. Since our first mobile platform was the HP Jornada with a relatively wide screen, we have started with a popular 8x8 SOM map. This geometry and this size provided enough space to organize all documents. The learning stage in this case was not complicated and the standard value of parameters sufficient. The obtained 8x8 map was successfully used by our students for several month and it is this map that was used in a study presented below.

Table 1. Parameters value for the Self-Organizing Map Training

	Ordering phase	Convergence phase 1	Convergence phase 2
t_{max}	10000	30000	50000
α_0	0.2-0.1	0.05-0.02	0.01-0.005
$N_c(0)$	3	2	1

Later, when a wireless card become available for the Palm (Handspring) platform, we have started to experiment with Palm-based devices. The standard Palm screen is relatively narrow (160 pixels). With our current Web interface it can fit only 3-4 map cells in a row. To adapt the map approach to Palm-size screen, we have explored a non-traditional 4x15 geometry. The goal was to obtain visualization scrollable only in vertical dimension in order make it easy to navigate the map. For this geometry the

learning stage was more complicated. First, we had to use the hexagonal geometry for the map to have the cells more tighten. Second, it was necessary to split the learning phase in three sessions and to use non-standard values of the parameters. The parameter values are provided in the Table 1. A representation of the geometry of the two maps is shown in Figure 6.

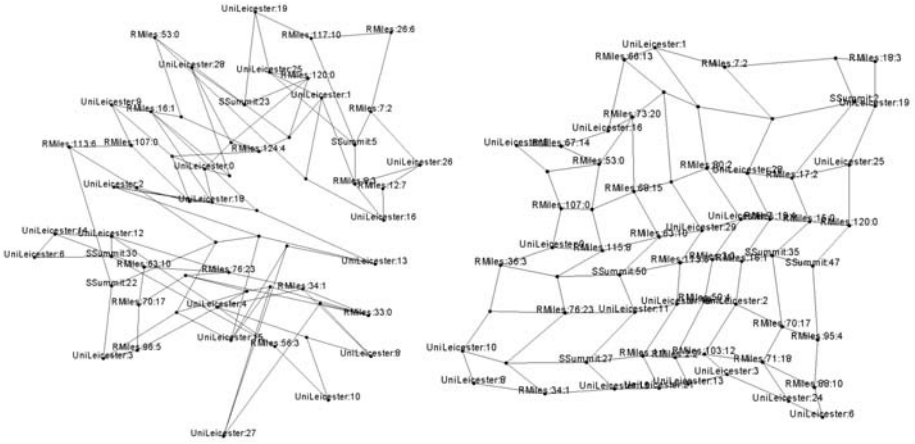


Fig. 6. The geometry of the 4x15 map (left) and the 8x8 map (right) after the learning phase

Despite the efforts we have put into developing 4x15 maps, we were not satisfied with the results. The resulting map did not look very natural (its geometry on Fig, 6 shows it clearly) and contained too many cells with no information. We concluded that this map could be more confusing than helpful for the students and ceased our work with narrow screens. Fortunately, the introduction of newest wireless Palm devices with 320x320 screens allows us to continue our work with Palm-based devices.

6 Similar Work

There are a number of known attempts to use SOM for developing various "information maps" - two-dimensional graphical representations in which all the documents in a document set are depicted. The documents on a SOM are grouped in clusters. Clusters that group documents on similar topics are near each other on the map. The effectiveness of the SOM as a tool to cluster information and to develop information maps was discussed in many research works. Some studies indicate that the clustering results obtained with SOM maps have meaning for the users. In particular, the proximity hypothesis (related topics are clustered closely on the map) was validated in [6].

In the WEBSOM system a SOM document map was used as a Web interface to classify Usenet newsgroup articles. The paper [3] reports the application of SOM network to organize 4600 documents. The documents were messages from the "comp.ai.neural-nets" newsgroup. In [5], a document map capable of organizing 131500 newsgroup messages was built using a parallel SIMD computer.

The computational complexity of a SOM neural network is particularly emphasized using TF*IDF representation because of the high dimensionality of the resulting vector space. The paper [7] argued that it is difficult to generate a map for large document collections (i.e. Gigabytes of data). This paper proposed a method for improving the speed of learning by exploiting the fact that the representing vectors are sparse vectors with many zeros.

Our approach combines the ideas of "information mapping" using SOM with the ideas of dynamic navigation in an open corpus hyperspace. Our goal is not simply to "map" the information, but to help the user navigate from a set of critical items (for example, lectures) to similar items. The use of a map distinguishes our approach from traditional "intelligent" hypertext that explores automatic and dynamic linking. Traditional automatic and dynamic linking ignores the user's intelligence in finding relevant hyperspace paths substituting it by "machine intelligence" that can offer ready to be used one-click links to relevant items. Our map-based approach relies on both "machine intelligence" in organizing a hyperspace map and the user's own intelligence in selecting a proper link on the map. It is similar to providing a city visitor with a map developed by an intelligent professional guide.

7 The Evaluation

The functionality and the usefulness of our map-based information access approach was evaluated it in the context of two programming-related courses at the University of Pittsburgh. Unfortunately, due to the insufficient number of Jornada organizers we were not able to run a large-scale evaluation of our approach on mobile devices. Instead, we have performed a formative questionnaire-based evaluation of 8x8 Knowledge Sea map used on a desktop computer. We have made the system available to the students of our courses, logged the student interaction with the system, and administered a non-mandatory questionnaire at the end of each course. The analysis of the student answers to some of the questions was partially reported in [2]. It has demonstrated that students regarded Knowledge Sea as a powerful tool for accessing external educational resources. Most impressed the students were with the system ability to place similar resource pages close to each other.

Only one question in the evaluation questionnaire was directly related to the issues of mobile access. The students were asked in which context they would expect to use the Knowledge Sea system from a Jornada-like device if it could be accessible from anywhere. The format of the question was "multiple selection"; the students were able to check any subset of the four offered options that ranged from "in the classroom" to "anywhere". Figure 7 summarizes the answers of 72 students who used the system in the context of an introductory programming during one of the three consecutive semesters (Spring 2002 to Spring 2003). It was a surprise for us to see that the locations selected most often (by about 60% of students) were home and library. Less than 40% of the respondents considered using the system in class and less than 35% "from anywhere". It shows that students are not quite ready for "anytime, anywhere" access. They consider a mobile device more as a different kind of computer and tend to use it in the context where they traditionally use computers (home, lab, and library).

Fortunately, the student attitude to the use of mobile technology in education is changing as rapidly as the mobile devices are becoming common in everyday life.

Figure 8 that splits the data presented on Figure 7 into three consecutive semesters shows that the percentage of students who are ready to access our system "from anywhere" has grown steadily over the 1.5 years of our study. At the same time, the percentage of student considering the use of mobile devices in a context where regular computers were more appropriate has declined.

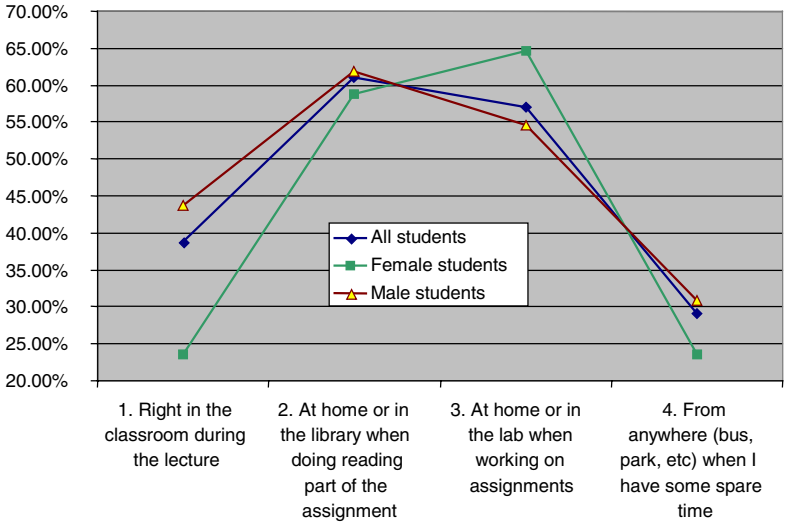


Fig. 7. Percentage of students considering the use of Knowledge Sea in different contexts

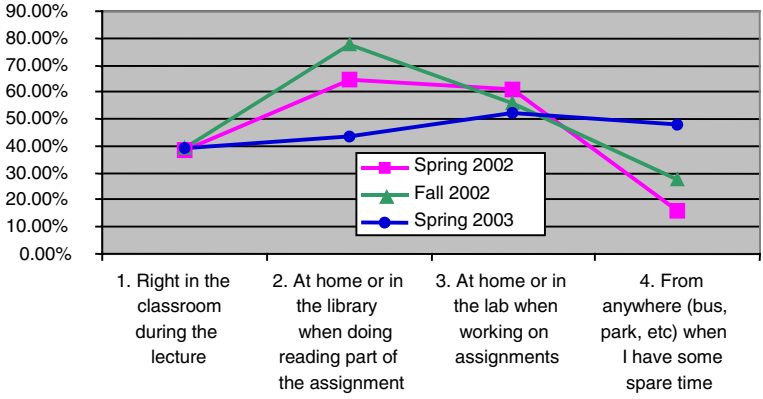


Fig. 8. The change of the percentage of students considering the use of Knowledge Sea in different contexts over three consecutive semesters

Another observation brought by our study is the difference between the attitude of male and female students to mobile technology. As a cohort, female students who have filled the questionnaire (17 out of 72) were slightly behind their male classmates

in being ready to use the Knowledge Sea system outside of traditional context. As shown by Figure 7, female students are more eager to use the system in the currently most traditional "desktop" context - at home when working on an assignment. They are less eager to use the technology in non-traditional places - like a lecture theatre or a bus. Another evidence is that females have checked generally fewer options among the offered four than male students. None of the female students selected all four options (checking all would mean that they are ready to access our system really from any context) while 10% of male students did so. Also, more than 47% of female students checked just one of the four contexts while only about 40% of male students did so.

Summarizing the results we can conclude that many students are not "mentally ready" to use mobile devices for educational needs "anytime, anywhere" as the proponents of the technology hope. Moreover, female students are slightly behind their male classmates in embracing the technology. At the same time, the prospects of educational use of mobile devices look quite bright since the students' attitude to this technology changes rapidly in the desired direction.

8 Lessons Learned and Future Works

Overall, we can conclude that SOM-based access to multiple information resources is a very useful technology. The 8x8 map that we have explored has worked well for the students. This map is large enough to provide a reasonable split of diverse content, yet is small enough to fit a Jornada-like handheld. We are now investigating the same map and the same interface in the context of a larger hyperspace of educational material (6 and more external tutorials instead of 3). We are also developing an improved interface for the system and working on integrating the map-based information access approach with our earlier work on adaptive hypermedia [1] and adaptive Web-based systems to develop an adaptive version of Knowledge Sea.

References

1. Brusilovsky, P.: Adaptive hypermedia. User Modeling and User Adapted Interaction **11**, 1/2 (2001) 87-110, available online at http://www.wkap.nl/oasis.htm/270983.
2. Brusilovsky, P. and Rizzo, R.: Using maps and landmarks for navigation between closed and open corpus hyperspace in Web-based education. The New Review of Hypermedia and Multimedia **9** (2002) 59-82.
3. Kaski, S., Lagus, K., Honkela, T., and Kohonen, T.: Statistical Aspect of the WEBSOM System in Organizing Document Collections. In: Scott, D. W. (ed.) Computing Science and Statistics 29. Interface Foundation of NorthAmerica Inc., Fairfax Station, VA (1998) 281-290.
4. Kohonen, T.: *Self-Organizing Maps*, Springer Verlag, Berlin, 1995.
5. Lagus, K., Honkela, T., Kaski, S., and Kohonen, T.: Self-organizing maps of document collections: A new approach to interactive exploration. In: Sinoudis, E., Han, J. and Fayad, U. (eds.) Proc. of Second International Conference on Knowledge Discovery and Data Mining, Menlo Park, CA, AAAI Press (1996) 238-243.

6. Lin, C., Chen, H., and Nunamaker, J. F.: Verifying the proximity hypothesis for self-organizing maps. In: Proc. of The 32nd Hawaii International Conference on System Sciences (1999).
7. Roussinov, D. and Chen, H.: A Scalable Self-organizing Map Algorithm for Textual Classification: A Neural Network Approach to Thesaurus Generation, Communication and Cognition. Artificial Intelligence **15**, 1-2 (1998) 81-112.
8. Salton, G., Allan, J., and Buckel, C.: Automatic Structuring and Retrieval of Large Text Files. Communications of the ACM **37**, 2 (1994) 97-108.

Spoken versus Written Queries for Mobile Information Access

Heather Du and Fabio Crestani

Dept. of Computer and Information Sciences
University of Strathclyde
UK G1 1XH
{heather,fabioc}@cis.strath.ac.uk

Abstract. Ease of browsing and searching for information on mobile devices has been an area of increasing interest in the information retrieval (IR) research community. While some work has been done to enhance the usability of handwriting recognition to input queries, the characteristics of speech as an input mechanism have not been extensively studied. It is intuitive to think that users would speak more words when issuing their queries due to the ease of speech when they are enabled to form queries via voice to an information retrieval system than forming queries in written form. Is this in fact the case in reality? This paper presents some new findings derived from an experimental study to test this intuition, and assesses the feasibility of the spoken queries for the search purposes.

1 Introduction

Today, the phone is the most widely adopted communications device anywhere in the world. Mobile phone subscriptions are increasing faster than Internet connection rates. A new market study indicates that nearly 700,000 people around the world are signing up every day for mobile phone subscriptions, even though mobile phone calls cost about three times as much as calls made with fixed or "wired" telephones. There were 23 million mobile phone subscriptions which surpassed the total population in Taiwan by the end of March in 2002. In UK, 70% of adults said they owned or used a mobile phone and almost 4 in 5 (78%) UK homes claimed to have at least one mobile according to a survey in May 2001. The development of wireless technology enables this huge mobile user community to take advantage of the large amount of information stored in digital repositories and access the information anywhere and anytime they want such as stock trading, e-commerce, travel reservations, order placements and tracking, and much more. Currently, the means of input user's information needs available are very much limited in keypad capability by either keying in or using a stylus on the mobile phone screen. Text-entry rates for the multi-tap method on older mobile phones are commonly 7-15 wpm; with predictive-text facilities this rate roughly doubles [3]. Key-tapping would therefore allow the entry of a typical 10-word question in 20-40 seconds, with continuous visual attention. Hand-writing with a stylus can be doubled at comparable speeds [4]. This would suffice to satisfy some information needs. However, such input style does not work well for those users in

F. Crestani et al. (Eds.): Mobile and Ubiquitous Info. Access Ws 2003, LNCS 2954, pp. 67–78, 2004.

many situations such as when users are moving around, using their hands or eyes for something else, or interacting with another person. In addition, the availability of screens and keyboards are not useful to those with visual impairment such as blindness or difficulty in seeing words in ordinary newsprint, not to mention those with limited literacy skills. In all those cases, given the ubiquity of mobile phone access, speech enabled interface has come to the lime light of today's IR research community which lets users access information solely via voice.

The transformation of user's information needs into a search expression, or query is known as query formulation. It is widely regarded as one of the most challenging activities in information seeking [1]. Research on query formulation with speech is denoted as spoken query processing (SQP), which is the use of spoken queries to retrieve textual or spoken documents. From 1997 (TREC-6) to 2000 (TREC-9), TREC (Text REtrieval Conference) evaluation workshop included a track on spoken document retrieval (SDR) to explore the impact of automatic speech recognition (ASR) errors on document retrieval. The conclusion draw from this three years of SDR track is that SDR is a "solved problem" [13]. SQP has very much been focusing on studying the level of degradation of retrieval performance due to errors in the query terms introduced by the automatic speech recognition system. The effect of the corrupted spoken query transcription has a heavy impact on the retrieval ranking [15]. Because IR engines try to find documents that contain words that match those in the query, therefore any errors in the query have the potential for derailing the retrieval of relevant documents. Two groups of researchers have investigated this problem by carrying out experimental studies. One group [5] considered two experiments on the effectiveness of SQP. In their first experiment, they recorded 35 TREC queries (topics 101-135) with query length ranging from 50 to 60 words with word error rate at three different percentage levels: 25, 33 and 50. The second experiment adopted substantially shorter queries of three lengths: 2-4, 5-8, and 10-15 content words which showed that as the query got slightly longer, the drop in effectiveness of system performance became less. Further analysis of the long queries by another group showed that [6] the longer "long" queries are consistently more accurate than the shorter "long" queries. In general, these experiments concluded that the effectiveness of IR systems degrades faster in the presence of automatic speech recognition errors when the queries are recognized than when the documents are recognized. Further, once queries are less than 30 words, the degradation in effectiveness becomes even more noticeable [7]. Therefore, it can be claimed that despite the current limitations of the accuracy of speech recognition software, it is feasible to use speech as a means of posing questions to an information retrieval system which will be able to maintain considerable effectiveness in performance. However, the query sets created in these experiments were dictated from existing queries in textual forms. Will people use same words, phrases or sentences when formulating their information needs via voice as typing onto a screen? If not, how different their queries in written form are from spoken form? Dictated speech is considerably different from spontaneous speech and easier to recognise [8]. It would be expected that spontaneous spoken queries to have higher levels of word error rate (WER) and different kinds of errors. Thus, the claim will not be valid until further empirical work to clarify the ways in which spontaneous queries differ in length and nature from dictated ones.

In this paper we present the results of an experimental study on the differences between written queries and their counterpart in spoken forms. The paper is structured as follows. Section 2 discusses the usefulness of speech as a means of query input.

Section 3 describes our experimental environment of the study: the test collection and the experimental procedure. The results of this study are reported in section 4. Conclusion with some remarks on the potential significance of the study and the future directions are presented in section 5.

2 The Question of Spoken Queries

The advantages of speech as a medium are obvious. It is natural just as people communicate as they normally do. It is rapid: commonly 150-250 wpm [9]. It requires no visual attention. It requires no use of hands. All mobile phones and many PDAs are equipped with microphones.

However, ASR systems are imperfect, which means that there is bound to be recognition mistakes at different levels depending on the quality of the ASR systems. Queries are generally much shorter than documents in the form of both text and speech. The shorter duration of spoken queries provides less context and redundancy, and ASR errors will have a greater impact on effectiveness of IR systems [7]. In contrast with spoken documents which can be processed and indexed offline, spoken queries need to be processed online and "almost" in real time. This intensifies the already computational expensive recognition process and demands the time for speech process to be kept short as it has been observed that user satisfaction with an IR system is dependent also upon the time the user spends waiting for the system to process the query and display the results [18]. Furthermore, input with speech is not always perfect in all situations. Speech is public, potentially disruptive to people nearby and potentially compromising of confidentiality. Speech becomes less useful in noisy environment. The cognitive load imposed by speaking must not be ignored. Generally when formulating spoken queries, users are not simply transcribing information but are composing it. For such tasks, the real limiting factor may be how quickly one can generate and formulate ideas. In this sense, it is no different from an accomplished typist who may be able to copy information quickly, but is slowed considerably when having to compose original text.

However, despite the unavoidable ASR errors, research shows that the classical IR techniques are quite robust to considerably high level of WER (about up to 40%), in particular for longer queries [12]. Voice is more expressive. It has more cues including voice inflection, pitch, and tone. Research shows that there exists a direct relationship between acoustic stress and information content identified by an IR index in spoken sentences since speakers stress the word that can help to convey their messages as expected [16]. People also express themselves more naturally and less formally when speaking compared to writing and are generally more personal. It has long been proved that voice is a richer media than written text [10]. Thus, we would expect, as a result, that spoken queries would be longer in length than written queries. Furthermore, the translation of thoughts to speech is faster than the transition of thoughts to writing. To test these two hypotheses, we constructed an experiment as described in the following section.

3 Experimental Study

Our view is that the best way to assess the differentiations in query formulation between spoken form and written form is to conduct an experimental analysis with a group of potential users in a setting as close as possible to a real world application [14]. We used a within-subjects experimental design [19] and in total, 12 subjects participated.

3.1 Subjects

As retrieving information via voice is still relatively in its infancy, it would be difficult to identify participants for our study. We therefore decided to recruit from an accessible group of potential participants who is not new to the subject of Information Retrieval. 7 of our participants were from the IR research group who have knowledge of Information Retrieval to some degree and 5 participants were research students who all have good experience of using search engines within the department of computer and information sciences, but few have prior experience with Vocal Information Retrieval. Our subjects participated the experiment voluntarily. It is worth to mention that all participants were native English speakers.

3.2 Text Collection

The topics we used for this experimental study was a subset of 10 topics extracted from TREC topic collection. Each topic consists of four parts: id, title, description and narrative. An example of such topic is shown in Table 1.

Table 1. An example of a TREC topic

<id> 1 <title> Topic: Coping with overcrowded prisons <desc> Description: The document will provide information on jail and prison overcrowding and how inmates are forced to cope with those conditions; or it will reveal plans to relieve the overcrowded condition. <narr> Narrative: A relevant document will describe scenes of overcrowding that have become all too common in jails and prisons around the country. The document will identify how inmates are forced to cope with those overcrowded conditions, and/or what the Correctional System is doing, or planning to do, to alleviate the crowded condition.

3.3 Experimental Procedure

The experiment consisted of two sessions. Each session involved 12 participants, one participant at a time. The 12 participants who took part in the first session also took part in the second session. An experimenter was present throughout each session to answer any questions concerning the process at all times. The experimenter briefed

the participants about the experimental procedure and handed out instructions before each session. Each participant was given the same descriptions of 10 TREC topics in text form. The 10 topics were in a predetermined order and each had a unique ID. The tasks were that each participant was asked to form his/her own version for each topic in either written form or spoken form as instructed via a graphic user interface (GUI) on a desktop screen (written in Java). For session 1, each participant was asked to form his/her queries in written form for the first 5 queries and in spoken form for the second 5 queries via the GUI.

For session 2, the order was reversed, that was each participant presented his/her queries in spoken form for the first half topics and in written form for the second half topics via the GUI. Each session lasted approximately 3 hours, which gave each participant to finish the tasks within 30 minutes and a maximum of 5 minutes time constraint was also imposed on each topic. Session 2 was carried out one week after session 1, this was because after the participants had taken part in session 1, they had familiarised themselves with the 10 topics to some degree, which would definitely pose a threat to the validity of our data if they worked with the same topics in session 2 immediately. By running session 2 some time after session 1, we hoped this threat would be minimised. At the end of the experiment, each participant was interviewed for about 10 minutes and a questionnaire was administered to each participant in order to obtain additional information about the process by which a participant formed the queries.

3.4 Data Capture

We utilised three different methods of collecting data for post-experimental analysis: background system loggings, interviews and questionnaires. Through these means we could collect data that would allow us to analyse and test the experimental hypotheses.

During the course of the experiment, the written queries were collected and saved in text format along with the duration of the formulation for each query after the participates typed their queries into the query field in the GUI and clicked "submit" button. The duration of each written query was counted as the total time a participant spent to comprehend a topic and formulate his/her query in the query field and submit it. The spoken ones were recorded and saved in audio format in a wav file for each participant automatically along with the duration for each query. After reading a topic, to record a query, the participant could click "starting speaking" button and speak his/her query into a microphone and then click "stop speaking" to terminate the recording. Similarly, the duration of each spoken query was calculated as the total time a participant needed to comprehend a topic and record his/her query.

The interviews sought to solicit participants' comments on the GUI design and explanations of his/her occurrence of some exceptional behaviour the experimenter observed during the course of experiment. They were also asked to point out the easiest and most difficult topics in written and spoken form and the reasons for their judgments.

The same questionnaires would be handed out after the completion of both sessions to gather participants' assessment on the complexity of the tasks. By comparing their answers, we could see how their ratings on the difficulty of the tasks would vary from session 1 to session 2.

Table 2. Characteristics of WRITTEN queries

Data set	q1-q120
Number of queries	120
Unique terms in queries	328
Average query length (with stopwords)	9.54
Average query length (without stopwords)	7.48
Median query length (without stopwords)	7
Average duration	02:13

Table 3. Characteristics of SPOKEN queries

Data set	q1-q120
Number of queries	120
Unique terms in queries	459
Average query length (with stopwords)	23.07
Average query length (without stopwords)	14.33
Median query length (without stopwords)	11
Average duration	01:58

4 Experimental Results and Analysis

From this experiment, we have collected 120 written queries and 120 spoken queries. Some of the characteristics of written and spoken queries are reported in Table 2 and Table 3 respectively.

These two tables pictured clearly that the average length of spoken queries is longer than written queries with a ratio rounded at 2.48 as we have hypothesised. After stopwords removal, the average length of spoken queries reduced from 23.07 to 14.33 with a 38% reduction rate and the average length of written queries reduced from 9.54 to 7.48 with a reduction rate at 22%. These figures indicated that spoken queries contained more stopwords than written ones. This indication can also be seen from differentials between the average length and median length for both spoken and written queries. There had no significant differences on durations for formulating queries in spoken and written forms.

The number of unique terms occurred in the written query set and spoken query set was very small. This was because each participant worked on the same 10 topics and generated a written query and a spoken query for each topic. Therefore, there were 12 versions of written queries and 12 versions of spoken queries in relation to one topic.

4.1 Length of Queries Across Topics

The average length of spoken and written queries for each topic across all 12 participants was calculated and presented in Fig. 1.

In Fig.1, the line for spoken queries is always above the line for written queries, which suggests the spoken queries were lengthier than the written ones. This was the

case for every topic persistently. This was exactly what we expected to see. We know from previous studies that the textual queries untrained users posed to information retrieval systems are short: most queries are three words or less. With some knowledge of information retrieval and high usage of web search engines, our participants formulated longer textual queries. When formulating queries verbally, the ease of speech encouraged participates to speak more words. A typical user spoken query looks like the following:

"I want to find document about Grass Roots Campaign by Right Wing Christian Fundamentalist to enter the political process to further their religious agenda in the U.S. I'm especially interested in threats to civil liberties, government stability and the U.S. Constitution, and I'd like to find feature articles, editorial comments, news items and letters to the editor."

Whereas its textual counterpart is much shorter:

"Right wing Christian fundamentalism, grass roots, civil liberties, US Constitution."

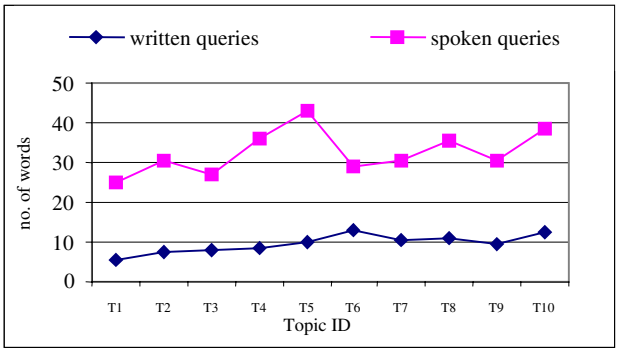


Fig. 1. Average length queries per topic

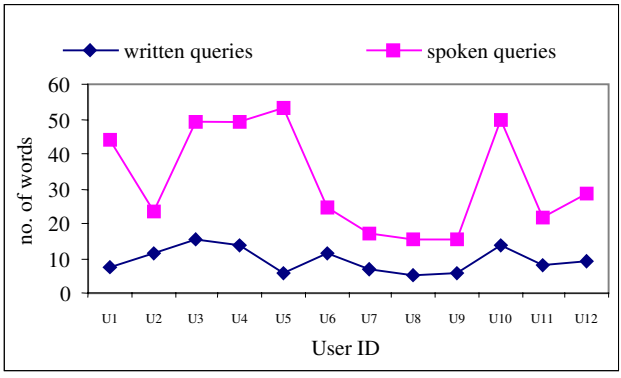


Fig. 2. Average length of queries per user

4.2 Length of Queries Across Participants

We also summarised the length of queries for all 10 topics across all participants. The average length of queries per user is presented in Fig. 2.

We could observe from Fig. 2 that it was the same case for every participant that his/her spoken queries were longer than written ones consistently. However, the variations of the length between spoken and written queries for some participants were very timid. In fact, after we studied the transcriptions of spoken queries, we observed that the spoken queries generated by a small portion of participants were very much identical to their written ones. The discrepancies of length within written queries were very insignificant and relatively stable. All participants used similar approach to formulate their written queries by specifying only keywords. The experience of using textual search engines influenced the participants' process of query formulations. For most popular textual search engines, the stopwords would be removed from a query before creating the query representation. Conversely, the length fluctuated rapidly within spoken queries among participants.

We didn't run a practice session prior to the experiment such as to give an example of how to formulate a written query and a spoken query for a topic, because we felt this would set up a template for participants to mimic later on during the course of experiment and we wouldn't be able to find out how participants would go about formulating their queries. In this experiment, we observed that 8 out of 12 participants adopted natural language to formulate their queries which were very much like conversational talk and 4 participants stuck to the traditional approach by only speaking keywords and/or broken phrases. They said they didn't "talk" to the computer was because they felt strange and uncomfortable to speak to a machine.

4.3 Duration of Queries Across Topics

The time spent to formulate each query was measured. A maximum of 5 minutes was imposed on each topic and participants were not allowed to work past this. All participants felt that the time given was sufficient. There was only one occasion a participant didn't formulate a written query within the time limit.

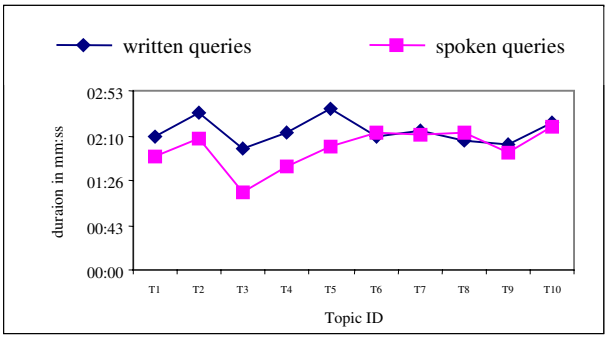


Fig. 3. Average duration of queries per topic

The average time participants spent on each topic is shown in Fig. 3. For the first half topics, more time was needed to form the written queries than spoken ones but the discrepancy was not as great as we expected. Participants spent almost same time to formulate query in written and spoken forms for each of the second half topics. From this figure, we were able to establish that no significant difference existed between the two query forms in terms of the duration. This appears to reduce a little weight to our claim that perhaps the participants would require less time to form spoken queries since that is the way people communicate to each other. However, we couldn't neglect the fact that the cognitive load of participant to speak out their thoughts was also high. Some of them commented that they had to well-formulate their queries in head before speaking aloud with no mistakes. One could revise one's textual queries easily in a query field, but it would be difficult for the computer to understand if one corrected one's words while speaking. Information retrieval via voice is a relatively new research area and there aren't many working systems available currently. Lacking of experience also pressurised the spoken query formulation process.

4.4 Duration of Queries Across Participants

The duration of queries per participant is shown in Fig. 4. Some participants spent less time on spoken queries than written ones, whereas it was a reverse case for some other participants. The variations of durations across all participants were very irregular and there were no significant differences among the durations for the two forms, therefore, we were unable to establish any strong claims. Nevertheless, the figure did show that two thirds of the participants spent less time on spoken queries than written ones whereas only one third of the participants required more time for spoken queries than written ones.

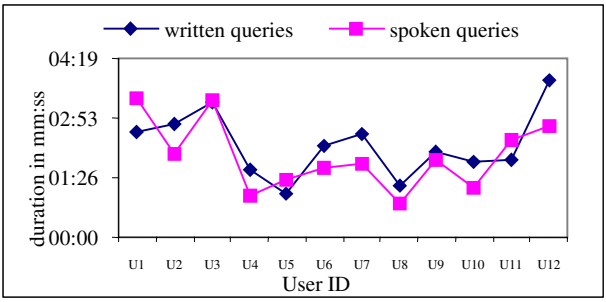


Fig. 4. Average duration of queries per user

4.5 Length of Spoken and Written Queries without Stopwords Across Topics

From the previous analysis, we know that spoken queries as a whole were definitely lengthier than written queries. One would argue that people with natural tendency

would speak more conversationally which results in lengthy sentences containing a great deal of function words such as prepositions, conjunctions or articles, that have little semantic contents of their own and chiefly indicate grammatical relationships, which have been referred as stopwords in information retrieval community, whereas the written queries are much terser but mainly contain content words such as nouns, adjectives and verbs, therefore, spoken queries would not contribute much than written queries semantically. However, after we removed the stopwords within both the spoken and written queries and plotted the average length of spoken and written queries against their original length in one graph, as shown in Fig. 5, which depicts a very different picture.

As we can see from the above figure, the line for spoken queries is consistently on top of the one for the written queries; after stopword removal, each of them is also undoubtedly becoming shorter. Moreover, the line for spoken queries without stopwords stays above the one for written queries without stopwords consistently across every topic. Statistically, the average spoken query length without stopwords is 14.33 and for written query, that is 7.48, which shows the spoken queries have almost doubled the length of the written ones. This significant improvement in length indicates that the ease of speaking encourages people to express not only more conversationally, but also more semantically. From the information retrieval point of view, more search words would improve the retrieval results. Ironically, for mobile information access, the bane is the very tool that makes it possible: the speech recognition. There are wide range of speech recognition softwares available both for commercial and research purposes. High quality speech recordings might have a recognition error rate of under 10%. The average word error rates (WER) for large-vocabulary speech recognisers are between 20 to 30 percent [2]. Conversational speech, particularly on a telephone, will have error rates in the 30-40% ranges, probably on the high end of that in general. In the case in our experiment where spoken queries are twofold lengthier than written queries, even if at the WER at 50%, it would not cause greater degradations on the meanings for spoken queries than written queries, in other word, the spoken information clearly has the potential to be at least as valuable as written material.

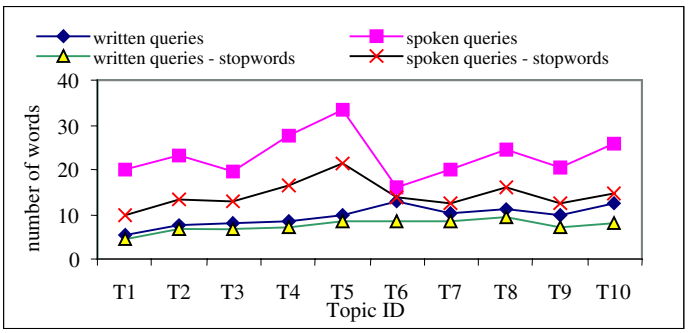


Fig. 5. Average length of queries across topics

4.6 Length of Spoken and Written Queries without Stopwords Across Participants

The average length of spoken and written queries with and without stopwords across all 12 participants is shown in Fig. 6. This graph shows a consistency with the result of the previous analysis that people tend to use more function words and content words in speaking than writing. This is a very case for every participant in our experiment.

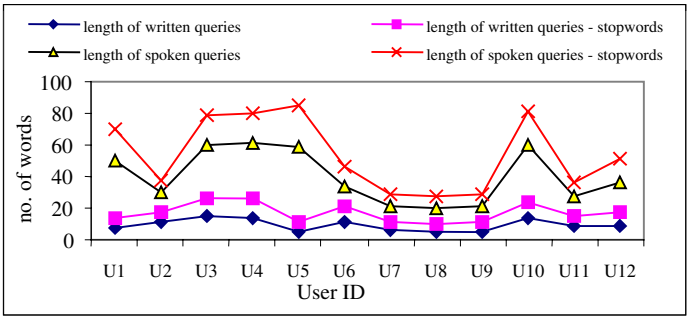


Fig. 6. Average length of queries per user

5 Conclusions and Future Work

This paper reports on an experimental study on the differentiations between spoken and written queries in terms of length and durations of the query formulation process, which also serves as the basis for the preliminary speech user interface design in the near future. The results show that using speech to formulate one's information needs not only provides a way to express naturally, but also encourages one to speak more semantically. This means that we can reach the conclusion that spoken queries as a means of formulating and inputting information needs are utterly feasible. Nevertheless, this empirical study was carried out with small number of participants, further studies are required with larger user population to underpin these results.

Information retrieval systems are much more sensitive to recognition errors when the queries are spoken than when the documents are speech recognition output [11]. We are fully aware of this potential threat, therefore for future work, we'd like to transcribe the recordings of the spoken queries using automatic speech recognition software and identify an information retrieval system which can be used to evaluate the effect of word error rate of spoken queries against written queries on the effectiveness of the retrieval performance.

In the mean time, we are carrying out a similar experiment on Mandarin which has a completely different semantic structure from English. The topics being used for this experimental study are a subset extracted from the TREC-5 Mandarin Track and the participants are all native Mandarin speakers with good experience in using search engines. The results obtained from this study will be compared to the ones reported in this paper.

Acknowledgments

The authors would like to thank all the participants who were from the Department of Computer and Information Sciences at the University of Strathclyde for their efforts and willingness in taking part in this experiment voluntarily.

References

1. Cool, C., Park, S., Belkin, N.J., Koenemann, J. and Ng, K.B. Information seeking behaviour in new searching environment. *CoLIS 2*. Copenhagen. (1996)403-416.
2. Eedro J. Moreno J-M. Van Thong, Beth Logan. From Multimedia Retrieval to knowledge management. *Computer,* pages 58-66, 2002.
3. M. Silfverberg, S. MacKenzie, and P. Korhonen. Predicting text entry speed on mobile phones. In *Proceedings of the ACM CHI 2000 Conference on Human Factors in Computing Systems*, pages 9–16, The Hague, 2000.
4. W. Soukoreff and I.S. MacKenzie. Theoretical upper and lower bounds on typing speeds using a stylus and keyboard. *Behaviour and Information Technology*, 14:379–379, 1995.
5. J. Barnett, S. Anderson, J. Broglio, M. Singh, R. Hudson, and S.W. Kuo. Experiments in spoken queries for document retrieval. In *Proceedings of Eurospeech*, volume 3, pages 1323-1326, 1997.
6. F.Crestani. Spoken Query Processing for Interactive Information Retrieval. *Data and Knowledge Engineering,* 41(1): 105-124, 2002.
7. J. Allan: Perspectives on Information Retrieval and Speech. *SIGIR Workshop: Information Retrieval Techniques for Speech Applications* 2001: 1-10.
8. E. Keller (Ed.), Fundamentals of Speech Synthesis and Speech Recognition, *John Wiley and Sons*, Chichester, UK, 1994.
9. D. R. Aaronson and E. Colet. Reading paradigms: From lab to cyberspace? Behavior Research Methods, *Instruments and Computers*, 29(2):250–255, 1997.
10. Barbara L. Chalfonte, Robert S. Fish, Robert E. Kraut. Expressive richness: a comparison of speech and text as media for revision. In *proceeding of the SIGCHI conference on Human factors in computing systems: Reaching through technology*. Pages: 21 – 26, 1991.
11. J. Allan. Knowledge Management and Speech recognition. *Computer*. April 2002, pages 46-47.
12. F. Crestani. Effects of word recognition errors in spoken query processing. In *Proceedings of the IEEE ADL 2000 Conference,* pages 39-47, Washington DC, USA, May 2000.
13. J. S. Garofolo, C.G.P. Auzanne, and E. M. Voorhees. The TREC spoken document retrieval track: a success story. In *Proceedings of the TREC Conference,* pages 107-130, Gaithersburg, MD, USA, November 1999.
14. S. Miller. *Experimental design and statistics.* Routledge, London, UK, second edition, 1984.
15. E. Mittendorf and P. Schauble. Measuring the effects of data corruption on Information Retrieval. In *Proceedings of the Workshop on Speech and Natural Language,* pages 14-27, Pacific Grove, CA, USA, February 1991.
16. A. Tombros and F. Crestani. User's perception of relevance of spoken documents. *Journal of the American Society of Information Science,* 51(9):929-939, 2000.
17. C. Cleverdon, J. Mills, and M. Keen. ASLIB Cranfield Research Project: factors determining the performance of indexing systems. ASLIB, 1966

Focussed Palmtop Information Access Combining Starfield Displays with Profile-Based Recommendations

Mark Dunlop[1], Alison Morrison[2], Stephen McCallum[1],
Piotr Ptaskinski[1], Chris Risbey[2], and Fraser Stewart[1]

[1] Computer and Information Sciences University of Strathclyde, Glasgow G1 1XH, Scotland
Mark.Dunlop@cis.strath.ac.uk
[2] Scottish Hotel School, University of Strathclyde, Glasgow G1 1XH, Scotland
Alison.J.Morrison@strath.ac.uk

Abstract. This paper presents two palmtop applications: Taeneb CityGuide and Taeneb ConferenceGuide. Both applications are centred around Starfield displays on palmtop computers – this provides fast, dynamic access to information on a small platform. The paper describes the applications focussing on this novel palmtop information access method and on the user-profiling aspect of the CityGuide, where restaurants are recommended to users based on both the match of restaurant type to the users' observed previous interactions and the rating given by reviewers with similar observed preferences.

1 Introduction

Starfield display technology has been proven to provide quick, dynamic and easy access to large amounts of complex data through use of scatter-plot displays and dynamic queries [1]. These techniques have been shown to be of great benefit in searching in many domains, e.g. house purchases [2], movies [1] and musical pieces guide[3]. However, starfield technology has traditionally been used only for large colour screens and is thus not widely considered suitable for small mobile devices.

Collaborative filtering has proven to be a very successful tool in many domains for helping users select appropriate items from large collections [4] and has been used in tourism, for example, to calculate guided tours [5]. Malone et al [6] describe three forms of information filtering: cognitive (often known as content-based), social (or collaborative) and economic. Balabanović and Shoham [7] discuss the relative merits of content-based and social recommendation approaches, primarily that content-based approaches are less prone to individuals with unusual tastes or to a small number of ratings, while social recommendations naturally take more account of significant non-content information that is likely to be missed by content-only recommendations.

This paper discusses our development of two starfield displays on palmtops – a city guide and a conference guide. The paper goes on to propose a combination of starfield displays with recommendation systems as a natural extension to starfield displays and describes this in the context of our city guide. In line with Balabanović's and Shoham's work on recommendation systems for internet pages, we take a view that combination of content-based and social recommendations are likely to be most effective for a tourism applications. Section 2 of the paper discusses starfield displays on palm-

F. Crestani et al. (Eds.): Mobile and Ubiquitous Info. Access Ws 2003, LNCS 2954, pp. 79–89, 2004.

tops, section 3 our collaborative filtering approaches with section 4 concluding the paper.

2 Palmtop Starfield Displays

In a previous project we showed that a palm-top computer based starfield display was a successful access method for a movie database despite being used on very small, monochrome, low resolution screens [8]. In that work, we compared using traditional palmtop-style access and starfield access to a collection of movies using two zoomable axes – year of release (x) and popularity of movie (y) together with direct on screen filters for movie genre (e.g. comedy, thriller…) and film classification certificate (e.g. U, PG…). The results confirmed our belief that starfield displays could be used on such small screens. Figure 1 shows an example search for all non-18 certificate movies excluding comedies. As well as providing fast searching, Starfield displays provide two main benefits over traditional data access methods: dynamic feedback and intuitive transitions from data overviews to focussed searching.

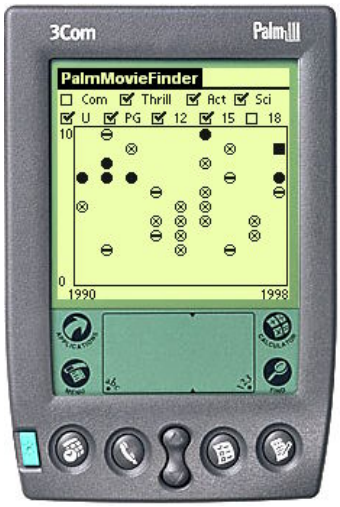


Fig. 1. PalmMovieFinder

Dynamic feedback is supported through controls and filters over the dataset. These controls support users searching the database rapidly and provide easy correction for many traditional database problems, e.g. when no data matches a query. In traditional database searching, null queries are notoriously difficult for users to correct – it is very difficult to slightly weaken a complex database query. In contrast, with starfield displays the query is built up in stages and the user knows precisely what (s)he did to cause the null query, thus (s)he can quickly undo that operation. For example, in the PalmMovieFinder a user looking for the lowest certificate thriller would deselect all genres bar thriller and then deselect 18, 15, 12,… in decreasing order. Once there are no matches turning that certificate back on shows the lowest certificate thrillers.

As highlighted in the HomeFinder [2] users can use the searching method to get an overview of the data and rapidly focus in on areas of interest. In the HomeFinder, users are given filters including type of property and house price on a geographic map of an area. When lowering the maximum price filter for a selected type of house, users rapidly learn the expensive areas of a city because they are the first to disappear. This kind of clustering or data overview is very hard to achieve with non-visual interfaces. If a user identifies a suitably priced area near his/her office (s)he can then zoom into that area and naturally restrict further queries to that geographic area.

We have developed two starfield displays on palms: the CityGuide tourist information application based around a geographic map using starfield display to show tourist attractions around a city and ConferenceGuide based around a timetable visualisation of a conference. Both applications have been developed for high resolution (320x320) PalmOS devices (the CityGuide in colour, the ConferenceGuide in greyscale) and were developed using a combination of PalmOS C and Sybase database storage.

2.1 CityGuide

The CityGuide application is designed around a map-based starfield (c.f. HomeFinder [2]) display to help tourists find attractions around a city. Our current implementation is based on a guide to Glasgow and contains an extensive restaurant guide with some information on cinemas, theatres and pubs. Brown and Chalmers [9] state that "tourists deliberately make plans that are not highly structured and specific, so that they can take advantage of changing circumstances". Our aim in developing this application of mobile starfield technology is to support tourists' unstructured searching of a city centre.

Fig. 2 shows a map of Glasgow city centre with an overlay of all restaurants, represented as squares. The map interface offers typical electronic map features such as zooming and panning: a user tapping on the display in Fig. 2 over, say, Central Station would zoom the map into that location, a further tap zooms further into that location and then the user can tap on a square attraction icon to see the name and then again for more details on that attraction. Users can pan the display by dragging with their stylus and zoom out by clicking on the '-' zoom icon.

On the top right of the display are a set of dynamic filters for controlling what points are shown on the starfield display, here showing type of attractions as restaurant (), the restaurant-type/menu filter () and the restaurant price filter (). Single choice filters are controlled by a pop-up menu, for example price in Fig. 3B. Due to limitations in PalmOS, multiple choice filters are controlled via a pop-up window, for example restaurants food type filter is shown in Fig. 3A. The results of a query are displayed directly on the starfield display as a revised set of attraction icons that match the current set of filters.

Brown and Chalmers [9] state that "when choosing where to go to, it is often safer to pick an area with more than one potential facility". Providing this kind of clustering information is one of the traditional strengths of starfield displays. Fig. 4 shows a brief interaction after applying the filters shown in Fig. 3 – here the user zooms into a promising looking area of the map, clicks on one restaurant then clicks on the restaurant name to get full details.

Fig. 2. Restaurant guide to Glasgow[1]

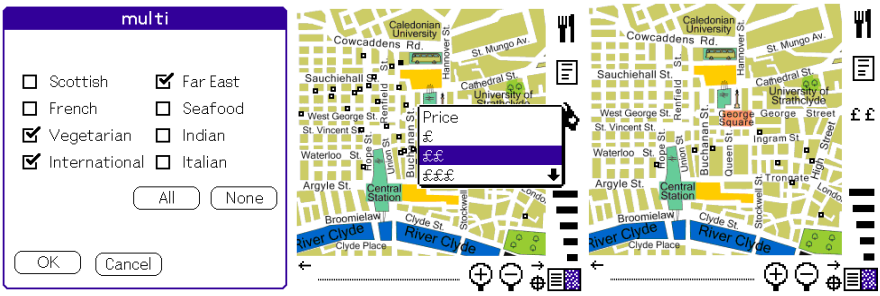


Fig. 3. Restaurant query filters (A: restaurant filter, B: price band filter, C: result of filters)

Users can mark an attraction as being on "My List" (similar to favourites/bookmarks in web browsers, see last image in Fig. 4) and later filter to show only attractions that have been added to this list. Users can also write their own reviews (see last image in Fig. 4), and have these published for others to read. On the bottom right of the map display is a scale bar for relevance and a list view – both of which are discussed later.

[1] Colour images are available at www.taeneb.com

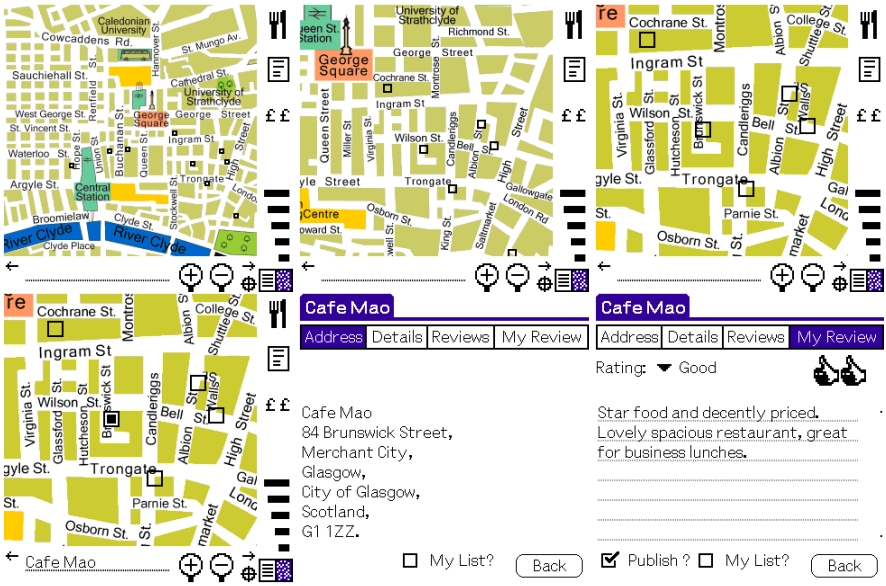


Fig. 4. Zooming into full details on Café Mao after applying fig. 2 filters

2.2 ConferenceGuide

Based around a timetable starfield display the ConferenceGuide initially shows users an overview of a day at a conference (see Fig. 5 for a sample day from our trial conference – EMAC 2003). Here parallel streams are shown as vertical columns with plenary sessions (in case of Fig 5, only breaks) being horizontal bars across all columns – closely reflecting the standard PalmOS Date Book application. Clicking on a session shows its name in the info box at the bottom of the screen with a further click giving full session details (e.g. session name and theme together with a list of talk titles, speakers and abstracts).

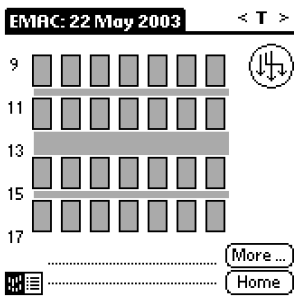


Fig. 5. EMAC conference timetable overview

Filters are provided on session theme and expected audience (not shown in fig 5). The session theme filter is initially set for all themes and users can limit the filter to

show only themes they are interested in (e.g. “Consumer Behaviour” or “Social Responsibility”). As with the CityGuide, users can add a session to their “My Sessions” list for later filtering to show only these sessions (a helpful tool for planning time at a multi-parallel conference) and view a textual version of a day’s events. The ConferenceGuide also supports inter-delegate communication through messaging and discussion forum services associated with each conference session.

2.3 Implementation Details

Both applications were developed on PalmOS devices (primarily Sony Cliés) using Metrowerks C. Extensive use was made of Sybase databases to hold the majority of the data and to manage synchronisation between palm, desktop and internet versions of the databases. Synchronisation was primarily via a physical connection to a networked PC but the Conference Guide was tested for "virtually continuous" wireless synchronisation using a wi-fi enabled Palm.

2.4 User Trial

A prototype of the CityGuide and ConferencePlanner were distributed to delegates at the EMAC 2003 conference on “Marketing: Responsible And Relevant?”. Twenty delegates were selected by the conference organisers for the trial and were given a Sony PalmOS Clié greyscale device for the duration of the conference. The city guide was populated in advance with a restaurant, pub and cinema guide (including “what’s on” information). Some restaurants were populated with reviews but both trial users and other delegates (through combined paper review forms and prize draw entries) were encouraged to write new reviews. At the conference venue the participants were provided with access to synchronise the software and were encouraged to do so at least daily. The feedback was gathered after two days of trial in a form of informal interviews.

A lot of interest in the application was shown by people who had experience of palmtops and by those interested in high-tech applications. People who had other than PalmOS devices, mostly PocketPC-based, expressed considerable disappointment that they could not use their own palmtop. One user found our PDA device in general too small and difficult to read and gave up using the system after the first day of trial. However, in general the users found the interface easy to use and intuitive – even those who had never used a palmtop before. They used the CityGuide mostly for searching for restaurants, they found the starfield interface an easy way of finding a restaurant and the review system very helpful during their selection of a restaurant (in particular delegates found it helpful to be able to read the opinion of other delegates attending the conference and most users added their own reviews).

The overall feedback was positive with many suggestions of extending the data content (such as adding museums, galleries, train timetables etc.) and connectivity of the device to give live update features.

3 Palmtop Collaborative Filtering

In this project we investigated combining a recommendation system with starfield displays to provide a filter on "relevance" in addition to the more traditional database style filters. Our recommendation system is based partly on content-based matching between user profiles and attraction profiles and partly on a social element from similar users' ratings of, say, restaurants. To build each user's profile we use a combination of implicit and explicit ratings. Nichols [10] highlights the problems of achieving a satisfactory number and quality of explicit ratings, where users are requested to explicitly *score* each, in his case, document: "the act of rating alters a user's behaviour from their normal pattern of reading" and that "unless the user perceives some benefit for participating in the system then they have an incentive for leaving". Within the tourism domain, local residents have a clear incentive for writing reviews of restaurants that their friends/colleagues may benefit from and thus they will benefit from their colleague's and friend's reviews. However, for visitors to a city there is little incentive for a user to write reviews as there is now little direct link between gain and effort. In contrast, implicit ratings are developed by simply monitoring a user's behaviour with the system. Nichols identifies the following potential types of implicit information: purchase, assess, repeated use, save/print, delete, refer/cite, reply, bookmark, examine/read, consider, glimpse, associate, and query. Most of these categories are used to build the user's profile in the CityGuide, as discussed below.

This section reviews our model for combining explicit information with implicit monitoring of the user's interaction and discusses how these are used to drive a relevance filter on a starfield display.

3.1 User Profile Building and Direct Matching

For each filter on the city guide we keep an individual user weight for each filter-value (e.g. Italian for the food-type filter value). When our prototype system is started users are asked to fill in a brief questionnaire for each food type (e.g. how much they like Italian food on a 5-point likert scale from "hate" to "love"). These initial scores are given a weight of 0 (hate) to 50 (love) and are then adjusted based on implicit ratings.

Following a scheme similar to Nichols and inspired by relevance feedback techniques in information retrieval [e.g. 11], scores for each matching criteria (e.g. Far-eastern and sea-food) are adjusted for many user actions in the interface. Currently the weights are adjusted as follows (Nichols's categories shown in parenthesis); when a user:

- writes a very good review of restaurant in that type: score +5 (assess)
- writes a good review of restaurant in that type: +3 (assess)
- writes a medium review of restaurant in that type: +1 (assess)
- writes a very bad review of restaurant in that type: -1 (assess)
- filters with this type turned on: +2 (c.f. query)
- add to "my attractions": +3 (bookmark)
- gets more details of a restaurant in this type: +1 (examine)
- read reviews of a restaurant in this type: +1 (examine)

In Nichols scheme reply, examine and glimpse were considered to be time based – the longer a user spent examining, the more important the document. For a mobile setting we felt this to be unreliable as levels of interruption were likely to be much higher, thus more frequently giving misleading measures of, say, how long a user spent reading a restaurant review. As such a fixed increment was used instead of a time-based measure, future experimentation is needed to investigate this decision.

Fig. 6 shows a sample rolling user profile after using the system for some time. Users would not normally see this information but it highlights how the system has developed a simple model of the user's tastes. Here the user has viewed/reviewed more on *International* and *Far Eastern* food than other categories thus we assume (s)he has a preference for these categories and has a relative dislike of sea-food.

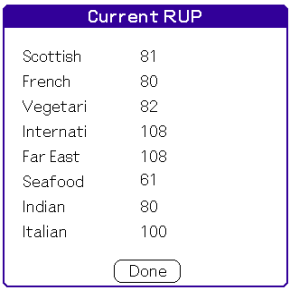


Fig. 6. Sample Rolling User Profile

3.2 End User Reviews

As shown earlier (Fig. 4), one of the core elements of the CityGuide is community reviewing where all users can read and write reviews. When a review is submitted the author's current Rolling User Profile is submitted with that review. This allows reviews to be measured for closeness to the current user's views (for example someone who hates expensive Indian food will have different view on an Indian restaurant to someone who loves all Indian food)[2].

Inspired by free text information retrieval techniques [e.g. 12] we calculate each user a personalised rating, $PARR_i'$, for restaurant R_i as follows:

$$PARR_i' = \frac{\sum \cos(P_{ai}, P_u) * R_{ai}}{\sum \cos(P_{ai}, P_u)}$$

where

P_u = user's current profile for whom we are personalising
P_{ai} = author's rolling user profile at time of submitting review i
R_{ai} = author's rating for restaurant R_i (scaled to between 0 and 1)
cos = cosine function for matching document vectors (see, e.g. [12])
$PARR_i'$ is a value between 0 and 1
summations are carried out over all reviews for restaurant R_i

[2] The current implementation only supports food-type.

The final personalised review value is given as follows to reduce the effect of only one review and manage zero reviews:

$PARR_i$ =

- 0.5 if there are no reviews
- ($PARR_i$' + 0.5) / 2 if there is only one review
- $PARR_i$' if there are two or more reviews.

Given the user ratings shown in Fig. 6, Fig. 7 shows Glasgow Restaurants rated by PARR given the current database of community reviews. This textual view is a useful complement to the starfield display for when location is not a prime issue in the user's selection – all filters work identically on the list view and map view and the user can rapidly flip between the two views.

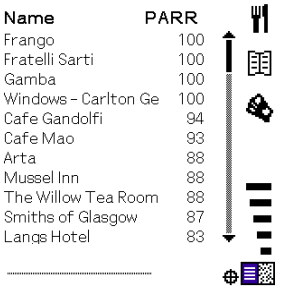


Fig. 7. Glasgow Restaurants rated by PARR

3.3 Combining Review Ratings and User Profile

The Personalised Rating (PARR) uses explicit review ratings of restaurants which are biased towards reviews from people with similar profiles to the user. However, the user's rolling profile does not directly impact these scores (it simply impacts the belief given to others' reviews). In contrast, the Rolling User Profile (RUP) does not take into account restaurant reviews.

These two scores are simply combined into the Combined Attraction Score (CAS): $CAS_i = RUP_i * PARR_i$.

While this scheme appears to work well, longitudinal studies are required to collect substantial amounts of data in order to formally experiment with different recommendation approaches and refinements of our current approach.

3.4 Combination Filtering

Bringing together collaborative community reviews and starfield filtering, we have added a "relevance filter" to the starfield map display. On the bottom right of the display a set of bars represent the openness of the relevance filter (from very open to very restrictive). Fig. 8 shows all restaurants in Glasgow on the left, reasonably tightly relevance filtered in the centre and only the best match on the right.

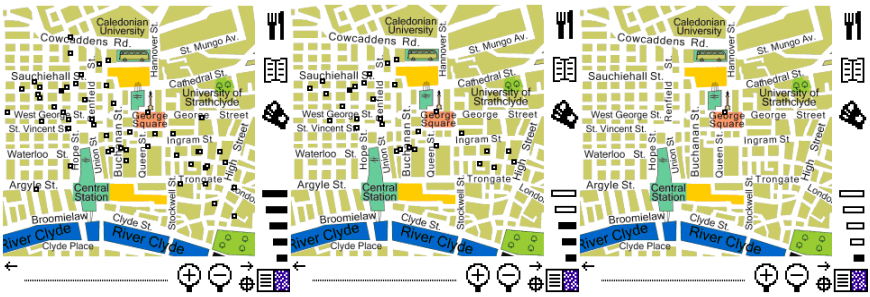


Fig. 8. Relevance Filter on wide, medium and tight settings

The relevance filter is driven by the Combined Attraction Score (CAS) to recommend restaurants based on both their review ratings and their match to the user's rolling profile. This filter works in combination with other filters so, for example, the tightest relevance filter will show the highest CAS ranked attraction that matches the current restaurant-type and price filter settings.

While not currently implemented we envisage a similar system for conference guides that would help to identify both sessions and individual relevant papers in conference with many parallel sessions.

4 Discussion

In this paper we have presented two novel starfield display implementations on palmtop computers: the CityGuide, based around a tourist attraction guide on a geographic plan of Glasgow, and the ConferenceGuide, based around a timetable for a multiple parallel session conference. Both interfaces have proven easy to use in a user trial of conference delegates visiting Glasgow.

The paper has also proposed a combined content and social recommender system for restaurant reviews based on a hybrid explicit/implicit rating system. This rating system is then used to drive a novel interaction tool, the relevance filter, on a starfield display so that users can directly control how much the system recommendations are taken into account when looking for items on the starfield display.

We are currently planning more formal user and technical evaluation of the algorithms and interface. Other context such as weather, user's current context, c.f. [13], and distance of attraction from current location, c.f. [14], are also being investigated as possible inputs to the recommendation system for general tourism attractions (e.g. walks in beautiful but distant botanic gardens tend to lose their appeal in heavy rain). Distance is likely to be more useful in textual lists, e.g. Fig. 7, as starfield displays naturally support zooming into a sub-area of the map, the interaction between these two views is also being investigated. We are also investigating possible improvements to the interface through using recommendations to guide the application of labels to some attractions (c.f. [15]).

In conclusion starfield displays on small devices have been shown to be successful on small devices and combining these with a recommendation system provides a powerful information access interface for small handheld devices.

Acknowledgements

The work presented here was mostly funded through The Taeneb grant from The Scottish Enterprise Proof of Concept Fund – to whom we are most grateful. We also extend our thanks to our trial users and to the organisers of EMAC03 conference for allowing us to experiment on their attendees!

References

1. C. Ahlberg and B. Shneiderman, "Visual Information Seeking: Tight Coupling of dynamic query filters with Starfield displays", *Proceedings of CHI '94*, 313-317 & 479-480, 1994.
2. C. Williamson and B. Shneiderman, "The Dynamic HomeFinder: evaluating dynamic queries in a real estate information exploration system", *Proceedings of ACM SIGIR 92*, 339-346, 1992.
3. H. Hochheiser, "Browsers with changing parts: a catalog explorer for Philip Glass' website", *Proceedings of the ACM conference on Designing interactive systems*, 105-115, New York, 2000.
4. P. Resnick , H.R. Varian, "Recommender systems", *Communications of the ACM*, 40(3), 56-58, March 1997
5. J. Fink and A. Kobsa: "User Modeling for Personalized City Tours". *Artificial Intelligence Review*, 18(1): 33-74, 2002.
6. T.W. Malone, K.R. Grant, F.A. Turbak, S.A. Brobst, and M.D. Cohen, (1987), Intelligent information sharing systems, *Communications of the ACM*, 30(5), 390-402.
7. M. Balabanović and Y. Shoham, "Fab: Context-based, collaborative recommendation", *Communication of the ACM,* 40(3), 66-72, March 1997.
8. M. D. Dunlop and N. Davidson, "Visual information seeking on palmtop devices", *Proceedings of HCI2000*, vol2, 19-20, 2000.
9. B. Brown and M. Chalmers, "Tourism and mobile technology". In: K. Kuutti, E. H. Karsten et al (Eds.), *ECSCW 2003: Proceedings of the eigth european conference on computer supported cooperative work*, Helsinki, Finland, p335-355, Dordrecht: Klewer Academic Press, 2003.
10. D.M. Nichols, "Implicit Rating and Filtering", *Proceedings of 5th DELOS Workshop on Filtering and Collaborative Filtering*, 31-36, Budapest, Hungary, 1998.
11. D. Harman, Relevance feedback and other query modification techniques. In Frakes and Baeza-Yates *Information Retrieval: Data Structures and Algorithms*, Ch 11, 241-263, 1992.
12. C.J. Van Rijsbergen, *Information Retrieval (second edition).* Butterworths, 1979.
13. K. Cheverst, N. Davies, K. Mitchell, A. Friday and C. Efstratiou, "Developing Context-Aware Electronic Tourist Guide: Some Issues and Experiences", *Proceedings of CHI'2000*, Netherlands, 17-24, 2000.
14. M. Brunato and R. Battiti, "PILGRIM: A location broker and mobility aware recommendation system", in *Proceedings of IEEE PerCom2003*, 2003.
15. J. Tatemura, "Dynamic Label Sampling on Fisheye Maps for Information Exploration", *Proceedings of the ACM Working Conference on Advanced Visual Interfaces*, 238-241, Palermo, Italy, 2000.

Designing Models and Services for Learning Management Systems in Mobile Settings

Alfio Andronico[1], Antonella Carbonaro[2], Luigi Colazzo[3], Andrea Molinari[3], Marco Ronchetti[4], and Anna Trifonova[4]

[1] Dipartimento di Ingegneria Dell'Informazione, Università degli Studi di Siena, Italy
andronico@unisi.it
[2] Department of Computer Science, University of Bologna, Italy
carbonar@csr.unibo.it
[3] Department of Computer and Management Sciences, University of Trento, Italy
{colazzo,amolinar}@cs.unitn.it
[4] Department of Information and Communication Technology. University of Trento, Italy
marco.ronchetti@unitn.it, anna.trifonova@dit.unitn.it

Abstract. The paper presents the guidelines of a project of three Italian Universities (Bologna, Siena, Trento) which aim is to investigate the use of mobile computing technologies to support the learning processes in a University context. The project covers three main areas. The first area is concerned with finding effective models for mobile learning. The second regards the evaluation of learning processes in mobile learning environments. The third focuses on the technological aspects of mobile learning, and on their integration with e-Learning systems, and more generally, with the information systems of the academic institutions. The project has its foundations in the availability of significant experience on e-learning real processes, and on the availability of the source code of an e-learning system developed in previous projects and currently used by different faculties, and of the newer platform that gathers the experience obtained in the past.

1 Introduction

Mobile learning is a field which combines two very promising areas – mobile computing and e-learning. Mobile learning could be considered any form of learning (studying) and teaching that occurs in a mobile environment or through a mobile device, like cellular phones, Personal Digital Assistants (PDA), smartphones, tablet PC etc. On the other side of mobile learning, we have e-learning, i.e., every educational process assisted by computers through the networks, and Internet in particular. M-learning has been considered as the future of learning or as an integral part of any other form of educational process in the future.

As m-learning is quite a new domain, there is a lot of work and research that is presently going on. Specifically, people are trying to understand:

- which learning models can help obtaining better learning processes when communication is mediated by mobile devices, and how the student mobility affects her/his learning process.

F. Crestani et al. (Eds.): Mobile and Ubiquitous Info. Access Ws 2003, LNCS 2954, pp. 90–106, 2004.

- how it is possible to evaluate efficiency and effectiveness of learning processes based upon mobile technologies, given the physical limitation of mobile devices.
- which services are useful for mobile devices, which is the enabling technology that can affect the wide diffusion of mobile learning.

A mobile learning educational process can be considered as any learning and teaching activity that is possible through mobile tools, or in settings where mobile equipment is available. National and international researches in the m-learning field are geared towards some lines that we shall here overview. Different devices that exist and all the devices that are coming up on the market, with their limitations and advancements, provoke different ideas for applying them on learning, thus any device can mean different m-learning. Among the open problems, some are relative to the pedagogical use of mobile devices. Since the m-learning term appeared for the first time, some research has been done to investigate the cognitive and pedagogical aspects.

Investigation had been done also on how useful mobile computing devices could be for reading or for workplace activities [1], on the basis of studying activity theory. Some authors [2] try to give directions to application designers for the areas, where the mobile devices should be most useful. Others [3] are trying to achieve conclusions by analyzing the theories of adult informal learning. In a few papers some interesting positive sides of using new technologies are underlined i.e. the participants are excited and want to try "new" things.

Some findings show that introducing new forms of teaching (even if this means just using a standard tool for drawing on a PDA) make students spend more time in working on that subject, comparing to the other subjects.[4] The currently evaluations and analyses of m-learning projects show many positive results. On the other hand there are some doubts if this excitement is, or is not, a temporary side effect. Most of the researchers think ([5][6]) that PDAs and other mobile devices should be seen more like extension, rather than replace the existing learning tools. Moreover not all kinds of learning content and/or learning activities are appropriate for mobile devices [7].

The paper will present our view regarding the topic on mobile computing. In particular, we'll present a project of our three Universities in which we want to use an existing Learning Management System and adapt it to the needs of mobility, having the source code of the system available. This mobile platform will be used to test principally new models for learning in mobile settings and tools for assessment of learning process through the use of mobile technologies. These objectives will be pursued through:

- Adoption of a well tested e-learning platform adapted to the usage of mobile devices
- Implementation mobile computing services in a University setting
- Study of learning models linked to mobile technologies
- Study of learning evaluation models based in an m-learning environment
- Design and development of Learning Objects suited to mobile learning, together with services for evaluating their effectiveness
- Experimentation of prototypes built in real learning processes

The paper is organized as follows: first we will present the state of the art in mobile learning, then we will briefly present the three elements that in our opinion help to build a mobile learning environment, i.e., models, evaluation systems, back-office

tools. Next we will present the guidelines for studying new models for learning processes in mobile settings, and one approach for the evaluation of these processes. Finally, the problems faced and choices made regarding the adaptation of a Learning Management System to mobile needs will be outlined.

2 State of the Art in Mobile Learning

The state-of-the-art in mobile learning research is heavily conditioned by the features of the devices available on the market. Different user interfaces, capabilities and connectivity may generate different ideas for possible learning applications: each single device can mean a different way to "m-learn". We shall here review the main trends and indicate some of the relevant papers in the field, with special attention to the themes that are more closely related with the aim of the present paper. A more extensive analysis of the state of the art can be found in [8][9].

Among the open problems, some are relative to the pedagogical use of mobile devices. Some research has investigated the cognitive and pedagogical aspects. Investigation had been done also on how useful mobile computing devices could be for reading or for workplace activities [1], on the basis of studying activity theory. Some authors [2] try to give directions to application designers for the areas, where the mobile devices should be most useful. Others [3] are trying to achieve conclusions by analyzing the theories of adult informal learning. In a few papers some interesting positive sides of using new technologies are underlined i.e. the participants are excited and want to try "new" things. Some findings show that introducing new forms of teaching (even if this means just using a standard tool for drawing on a PDA) make students spend more time in working on that subject, comparing to the other subjects.[4] The current evolution and the analyses of m-learning projects show many positive results. On the other hand there are some doubts if this excitement is, or is not, a temporary side effect. Most of the researchers think ([5][6]) that PDAs and other mobile devices should not be seen as a replacement of existing learning tools, but rather as a new and different opportunity. Moreover not all kinds of learning content and/or learning activities are appropriate for mobile devices [7].

People are experimenting with the application of m-learning to different fields: a promising one is language learning. At Stanford Learning Lab [10] an exploration of mobile learning has been done by developing prototypes that integrate practicing new words, taking a quiz, accessing word and phrase translations, working with a live coach, and saving vocabulary to a notebook. They envisioned that a good approach would be to fill the gaps of time by short (from 30 seconds to 10 minutes) learning modules in order to use the highly fragmented attention of the user while on the move. The research indicates some very useful directions, like the length of the learning materials, the personalization of interaction and the frustration of the user and the decreasing of the perception of the learning materials because of the poor technological implementation. In the same field an ongoing project [11] aims at porting to mobile systems an ad-hoc language-learning system developed for the special needs of an Italian bilingual region, where every public officer is supposed to be fluent in Italian and German. One problem investigated in this context is the one of anticipating user's need and pre-caching the needed content when a cheap and fast connection (such as a direct connection via cradle) is available, since the whole material is too large to fit in a small palmtop device.

Many authors approach m-learning in the context of life-long learning. One of the biggest initiatives in such domain is the *HandLeR* project [12] (University of Birmingham). The project attempts to understand in depth the process of learning in different contexts and to explore the lifelong learning. The stress is on communication and on human-centred systems design. Similar in some concepts to *HandLeR* is the project undertaken at the Tampere University of Technology (Finland) [13], where PDAs are used for mathematical education of children. The study-content is presented in the form of a game where the pupils can communicate and help each others and the electronic device is used to measure the average students' knowledge level and to adapt the speed of presenting new material to the learners' ability.

One of the most straightforward application of the usage of mobile devices as educational supporting tool is messaging. At Kingston University (UK) an experiment was undertaken to research the effectiveness of a two-way SMS campaign in the university environment [14,15]. The team has developed a system that sends SMS to students, registered to the service, about their schedule, changes in it, examinations dates and places, student's marks and etc. The conclusions of the experiment were that the students in certain scenarios where a certain type of response is required preferred SMS as a medium to e-mail or web-based announces. SMS could be efficiently used in education (m-learning) as a complementary media. As the technology improves (i.e. EMS and MMS, potential more user-friendly interface) the potential increases too. For this reason, as explained in the next sections, we decided to include in our experimentation the management of SMS from teachers / administrative staff to students as one of the approaches to info-mobility. Also at the University of Helsinki the *LIVE* (*Learning In Virtual Environment*) experiments, made with SMS system and with WAP phones, were very positive [16]. The project went on by introducing digital imaging and sharing photos between the participants (teachers). The conclusions were that it is very possible that the introduction of MMS and the other 3G services in the large scene will lead to more and more possibilities for m-learning. Another project [17] on evaluation of a Short Messaging System (SMS) to support undergraduate students was done at Sheffield Hallam University. The implemented system was again not for learning, but for managing learning activities (to guide, prompt and support the students in their learning). The findings were overwhelmingly positive, with students perceiving the system to be 'immediate, convenient and personal'. Positive results were underlined and after the outcomes from a survey in Norway - almost 100% of the students in that University have cell phones and SMS system would be widely accepted [18]. Once again an SMS system was considered to be used to spread information about lectures and classes, corrections in the schedule and etc. In certain cases students find it more convenient than e-mail or WWW as the information always comes on time. These projects open some very important issues to be considered in doing further research in the mobile learning domain. One is that the current technology gives enough powerful instruments to support some new forms of auxiliary learning tools. They also show the enthusiasm of the students to accept such new technologies.

Several m-learning projects focus on of how to apply e-learning techniques and content on mobile platforms. The UniWap project ([19][20][21][22]) concentrated on testing the use of WAP technology in higher education, by exploring the process of creating an operating environment for studying and teaching through smart-phones and WAP phones. One phase of the project was to create some working prototypes

(courses modules) and to investigate the problems and the value of such courses. The positive results they encountered (easy to develop, willingly accepted and widely used modules) encourage them to continue investigating the new coming technologies – digital imaging with mobile devices, 3G, etc. At Ultralab *M-Learning* project the team is producing m-learning materials for people with literacy and numeracy problems [23],[24]. A great potential is encountered from the cognitive and pedagogical point of view, even by using simple development tools (Macromedia Flash).

"From E-learning to M-Learning" [7] is a long-time project that aims at creating a learning environment for wireless technologies by developing course materials for range of mobile devices. The authors discuss the devices characteristics that are proper for learning and highlight analogies and differentiation between e-learning, d-learning (distance learning) and m-learning. They also try to predict which methods and technologies should be used for successful m-learning.

Tourist and museum guides are often considered to be applications in mobile learning domain. They usually refer to newest technologies as location-discovery via GPRS, radio frequency or etc. However we rather consider them as a separate applicative field and therefore we will not discuss them in this context. Also, due to space constraints we cannot discuss the very interesting approach of using mobile devices in the framework of collaborative and problem-based learning. The interested reader can find indications and a short discussion of this topics in [9].

In conclusion, the overall view on the existing research work and projects in the m-learning domain shows that it most probably applies best to processes, where specific knowledge should be retrieved/accessed in a certain moment, where discussions in distributed groups (i.e. brainstorming) appear, where data is collected or utilized "on the field", and where context-information is strongly related to the learning content. The nature of mobile devices, with their small screens and poor input capabilities leads to the assumption that they can not replace the standard desktop computers or laptops. However, the same properties can make them efficient in learning domain, if certain constraints are kept ([7][17][25][26]):

- Short modules (max 5-10 minutes). Users should be able to use their small fragments of waiting time (i.e. waiting for a meeting or while travelling in a train) for learning, like reading small pieces of data, doing quizzes or using forums or chat for finding answers to "on field" questions.
- Simple, funny and added value functionality. The limited computational power and the other properties of mobile devices (as they are today) make it difficult to use complex and multimedia content. One should find it more interesting or necessary and useful (or at least equally) to study using this m-learning system in his/her 5 min. break than playing a game on the same device.
- Area/Domain specific content, delivered just in time/place. The mobility should bring the ability to guideline and support students and teachers in new learning situations when and where it is necessary. The dependency of the content can be relative to *location* context (i.e. the system knows the location where the learner resides and adjusts to it), *temporal* context (i.e. the system is aware of time dependent data), *behavioural* context (i.e. the system monitors the activities performed by the learner and responds to them adjusting its behaviour) and interest specific context (i.e. the system modifies its behaviour according to the user's preferences). Of course a mix of the contextual dependencies is possible and likely.

3 The Three Elements of Building a Mobile Learning Environment

As said in the introduction, the aim of the project has three key elements. Firstly, we are interested into analyzing and viewing the system as whole and thus researching, whenever it would be possible, models that would allow us to individuate the relationships that connect those elements, as well as their knowledge value and reach. Therefore, the concept of model becomes the basis to connect the learning process with the languages, the methods, and the tools that are employed to implement and experiment the Virtual-Real Learning Communities. Such communities should deliver evaluations of the result of learning process and objective measurements parameters, which are (possibly) independent from the teaching contents.

A second but not secondary issue is concerned with how to evaluate the m-learning tools and their model as a function of the induced quality in the learning processes. Talking about good quality in distance learning is undoubtedly a not easy task. Not easy for various reasons, first among everybody because has not closed the debate on what he understands, in more general sense, for quality of a formative intervention, with all what which this involves yet: didactic effectiveness, social and professional impact, investment, etc. We would like to assume for quality not as much the excellence as rather the management of a continuous process to approach the most possible the wished effect (for instance, what one wishes is learned) to real effect (what which has been learned). We call such systems closed ring, key element of this kind of systematic realignment is a constant monitoring aiming to the evaluation both of the users and of the whole process. The system of new generation which we intend to develop is based on the interaction of all the parts of the process, to give way to the distributor of the formative action, to monitor the process and to regulate it, when necessary, wished to redirect it adequately toward the effect.

A key element for this is a constant monitoring, whose aim is to both evaluate users and the whole process. The new generation tool that we intend to develop is based on the interaction of all process components, so as to allow tutors to monitor and steer the process. In such way it will be possible to achieve a better coherence with the stated objectives, making therefore easier to reach the desired goals. More in detail, the evaluation of the proposed system is expressed in functionalities which refer to various kinds of Assessment. The first and simpler functionality is the self-evaluation which must be understood as complement of an educational process. The self-evaluation is not sufficient to guarantee the success of an educational process, in fact, not all the students are able to self-manage it in an effective way. So, we would like to consider some other assessment strategies. The evaluation process assumes as a good evaluation is not reduced to the administering of a final test and to the production of a judgment, or more simply of a vote. The assessment must to precede, to follow and to direct all the formative process. That means that the system obtains information about the students before beginning a course (using previous relationships with the same student or a diagnostic test), during the development (through the analysis of link and documents chosen by the same students, explicit preferences and formative tests) and in conclusion of unitary subject sections.

A big complexity resides in the difficulty for the electronic computers to semantically interpret sentences in natural language. A first approach to the problem has been performed trying to isolate the verifiable difficulties in traditional testing systems

(refer in particular to the North American model, which uses questions with answers to multiple choice). These have been summarized in the following six points, concerning multiple choice tests:

- they are concerning the results of the learning process, not to the processes
- they underline the knowledge level not the potential of learning
- they are far from the working contexts
- the memory can sometimes be more useful than the comprehension
- the so-called tests-taking skills can affect the result.

Possible answers to these problems are presented in [27]. In the context of the present project we would like to highlight two particulars. First of all, the personalization of the tests is possible only in presence of a student model that memorizes a description of his expertise and brings up to date. Besides, the enlargement of the field of action of the evaluation, from the results to all the educational process, makes it possible the use a graph structure.

As a third key element of the project, in order to support the experimentation of any tool or technique of m-learning, a rather complex information system is necessary. Its role includes distributing didactic material, users identification and authorization, gathering of data relative to the user-system interaction, provisioning of mobile services, supplying statistics on level of usage and satisfaction etc. From this point of view, the project attempts to interconnect m-learning technologies with e-learning, and e-learning is in turn always more integrated in the information systems of academic institutions.

E-learning systems, and Learning Management Systems (LMS) in particular, are nowadays a key element in the learning processes that take place at Universities, and they are widely investigated in literature [28], [29], [30], [31]. Several implementations are available on the market, like for instance LearningSpace™, WebCT™, Blackboard etc. [32]. They are in the middle of a transformation from simple support of on-line learning (like in the case of LMSs) into real information systems (Learning Information Systems -LIS). As such, they integrate many components of the wide spectrum of a formative action [33]. Our project needs to integrate such systems with our project's specific mobile-computing requirements. This means that we have to focus mainly on two points: on the one hand we have all the administrative and back-office processes of a Faculty (e.g. exam registration, didactic design, theses management, bookkeeping of teacher's activity, University marketing etc.).

On the other hand, research attempts to focus on the technological evolution that brought to people mobility and mobile terminals (PDAs, pocketPCs, cellular phones, smart-phones, tabletPCs etc.) that are now present in every day's life. These tools are an interesting for a LIS, since they allow the various actors (such as students, teachers, administrative personnel etc.) to have a mobile platform that keeps them in touch with the LIS wherever they are. The possible applications are therefore very many: we can for instance think at the possibility for a secretary to communicate with mobile-technology enabled students, or at possible mobile collaboration among teacher and students within a course framework (our research will explore this aspect).

Some work has been done on Learning Management Systems, but the idea of a University Information System having a mobile component that belongs to the skeleton of the Information System is still in its infancy. It is therefore clear that it is not possible to be concerned with single classes of actors without considering the whole picture, since LIS aggregates users with different roles. The focus therefore moves

from a system dealing with "courses" to a system that deals with "virtual communities", i.e. with a generalized communication space that allows using a variety of tools to support collaboration needs that may arise in various situations. A virtual community can be supported at various levels by mobile technologies. LIS, in our definition, become computerized tools that give various kinds of services to virtual communities. Such services can be adapted to the special needs of a given community. One research aspect of the present project is therefore linked to virtual communities and info-mobility related to learning: we intend study and experiment how activities of an e-learning portal can be integrated with the emerging mobile technologies. The research group will use an already existing community-oriented e-learning portal that has been in use for some time to integrate and test mobile technology and related methodologies.

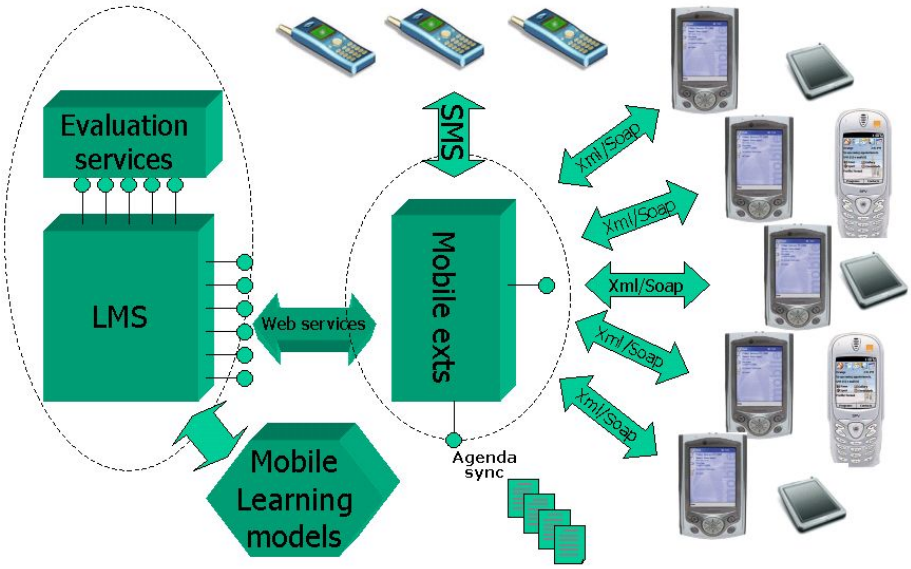


Fig. 1. A general schema of the prototype

4 Evaluating Mobile Learning Settings

The experience from years of development and use, the advance of technology, and the development of authoring tools for questions and tests has resulted in a sophisticated, computer based assessment system. However, there is still a lot of room for further development. Some of the current ideas for development are discussed in the remainder of this paragraph. In line with many writers in the field of assessment, we distinguish three types of assessment:

- diagnostic assessment; it provides an indicator of a learner's aptitude for a programme of study and identifies possible learning problems;
- formative assessment; it is designed to provide learners with feedback on progress and informs development but does not contribute to the overall assessment;
- summative assessment; it provides a measure of achievement or failure made in respect of a learner's performance in relation to the intended learning outcomes of the programme of study.

The most common distinction in the literature is that made between formative assessment and summative assessment. A formative computer-based test is described as one where the results of the test do not contribute to a student's final grades. Instead, the student's scores are used to assist in improving the student's learning, often by identifying weaknesses in the student's knowledge and understanding of a given area or by helping them to identify and correct misconceptions. In a similar way, lecturers can also make use of the results obtained to help them improve their teaching by identifying areas that students have found difficult to understand. Nonetheless, in many assessment activities the difference is not so evident.

A primary aim of assessment is provide the necessary information to improve future educational experiences because it provides feedback on whether the course and learning objectives have been achieved to satisfactory level. Yet, it is important that the assessment data be accurate and relevant to effectively make informed decisions about the curriculum [34]. As discussed above, formative assessment can also be used to help bridge the gap between assessment and learning. This may be achieved particularly where assessment strategies are combined with useful feedback, and integrated within the learning process [35].

This feedback need not be limited to correct/incorrect responses, but can include detailed textual feedback about answers and the topic area of the question. Formative assessment can assist in consolidation of learning, and in identifying weaknesses in assumed understanding. We think that it would be helpful to be able to deliver the same questions in a number of modes. For example, help mode, exercise and exam, with the test author being able to configure this to their own requirements. The help mode supports students when they start out on their learning; accordingly, the questions are delivered with maximum feedback including hints, visible marking on screen and the chance to reveal a correct answer. Exercise mode restricts the help to just visible ticks and crosses on screen for right and wrong responses. Finally, exam mode presents questions with no option for revealing answers and no ticks/crosses appearing.

Our summative strategy consists of two phases: the former to find the approximate student level, the latter to give the student the right mark using a set of questions customized on his capabilities. The preliminary examination contains for every subject two or more questions for each difficulty level. The score obtained by the student in the first test is used to choose questions to propose in the second test. Using this technique we can build a test which is not redundant (due to the adaptivity) and the same firs test set for every student, so we can get data on the quality of the items. Diagnostic assessment is quite similar. In particular, the two-session strategy is the same. The main difference is that it is taken before starting a course, to decide what kind of resources will be used. In this case, the system knows nothing about the student's knowledge; it also records the scores of every answer, so the system can use them when it needs to explain a topic already scored.

When an exam session is completed, we will have a score for every candidate and for every question. To obtain a human-understandable mark we used a function depending on two parameters α and p. We used this function in a large number of real cases and the experimental data showed that the choice of α is important to obtain well-distributed marks. This value can be adjusted after the test correction, in response to the candidate's answers. Moreover, useless items may be discovered. The value p is used to give full marks.

To compose tests easily from a set of items and correct them, the system uses normalized questions and manages the item weighting: when an author creates a course, he sets weights that will influence the automatic item selection and the scoring algorithms. Some of the available forms of assessment strategies included in the proposed system are:

- true/false,
- multiple-response question; it is defined as a question in which the candidate is required to select two or more correct answers from a list of options. Both the number of correct answers and the number of options may vary. We consider the following three principle modes: i) constrained selection: the student is forced to make a prescribed number of selections, usually the same as the number of correct answers; ii) partially constrained selection: the student may make any number of selections up to the number of correct answers; iii) unconstrained selection: the candidate may make any number of selections up to the maximum number of options,
- extended matching item and drag and drop question types share the same process of selection. In either case the student is required to select a number of items from a list then enter or move them to their correct positions. Thus the candidate must make two selections - which item and where to put it. The scoring simplest form considers a positive score allocated for each item correctly positioned,
- image hot spot,
- code writing.

The process of assessment involves gathering information from a variety of sources to develop a rich and meaningful understanding of student learning. Modern computer assisted assessment packages are capable of storing and analysing vast amounts of information on student learning. With appropriate analysis this data can be used to identify the strengths and weaknesses of individual students and match them to learning resources that meet their needs.

Finding appropriate, high quality resources has now become a significant challenge. Furthermore, based on user's requirements and interests, filtering and retrieval tools should be developed, improving their usage. Information filtering systems can help learners by eliminating the irrelevant information, operating like mediators between the sources of information and the learners. Personalized filtering should be also a process of filtering based on not only the long-term interests but also the short-term requirements. For these purposes, we consider relevant the integration of an hybrid recommender system that combine content analysis and the development of virtual clusters of students and of didactical sources. This information management system provides facilities to use the huge amount of digital information according to the student's personal requirements and interests, with special focus on the development of new algorithms and intelligent applications for personalized information

classification and filtering. In this way data can be obtained about which material is proving to be most effective in raising student achievement. Taken together with the profiles of student strengths and weaknesses, this may prove an effective tool for identifying which resources are most suitable for each student, giving them an individual program of study, tailored to their needs.

The assessment process could be organized in the following phases:

a) Creation of the architecture for the management of the evaluation moments for the whole formative process: that is, the teaching interface building (through mobile devices and through fixed Web positions), the student interface building (through mobile devices as cellular telephones, PDA, Smart-phone etc.) and the administrative interface building, for example for the creation of authorized teachers and students.

b) Creation of the test databases organized in atomic sets of different kind of requests (multiple choices, open, closed, fill in gap, building of sentences, problem-solver …). Please notice as the sentences building is applicable to also very different contexts among them, what, to example, the program writing (building of code) and the slang contexts of hypothetical deductive disciplines: in these cases, in fact, we should use words extracted from a predefined vocabulary and verify the respect of detailed set of rules.

c) effectiveness and consistency analysis of the databases produced to the previous point through the application of "item analysis specifications" (on real cases)

d) Management of the various assessment processes. The distinction of the evaluation moment affects the management, for example, the choice of questions to be submitted to each student.

e) system evaluation which allows to make experimentations on the principal platforms which at present the more diffuse PDA computers equip on the market. We intend to experiment the project using different student groups, for example in "Programming" course (Laurea Triennale in Scienze dell'Informazione, Cesena) and "Artificial Intelligence and E-learning" (Laurea Specialistica in Informatica, Bologna).

5 Adapting a Learning Management System to Infomobility

As already mentioned, a rather complex information system is needed in order to support the experimentation of any tool or technique of m-learning,. The role of such system includes distributing didactic material, user identification and authorization, gathering of data relative to the user-system interaction, provisioning of mobile services etc. The objective of the project is to obtain an unified platform where the various actors can use different communication services, both mobile and not. In this regard, e-learning systems in general, and more specifically Learning Management System, are by now a vital component in the distance educational field. We have to integrate LMS with two different classes of processes:

- on one hand, processes connected with the administrative (back-office) activity of a faculty (like registering exams, programming the teaching activity, theses management, bookkeeping of the lecture hours, faculty marketing etc.: all such processes have important overlaps with processes managed by an LMS.

– on the other hand, technology evolution has pushed toward a strong mobility of all the actors, and has furnished mobile devices (PDA, pocketPC, cell-phones, smartphones, tablet-pc) that accompany the user in every day's life. Such tools can become additional terminals for a LIS, because they allow all actors (students, teachers, secretaries, dean, tutors, administrative personnel etc.) to stay in touch with the LIS wherever they are.

The number of possible applications is huge: for instance, the possibility for the administration to communicate in real time with students equipped with such devices, new forms of collaboration among students and teachers within an University course, the chance for the students to interact among them regarding the courses etc. The focus moves therefore from a system that is based on "offering courses" into a system based on the idea of "virtual community". A virtual community is a highly generalized collaboration space. In such way, a course given by a teacher, a seminar, the group of students preparing their thesis with the same teacher, students working together on a project, etc. are all instances of virtual communities. A LIS becomes a computer-based tool that gives services to virtual communities, and must be adapted to the specific needs of each particular community. We already built, over several years, a community-oriented learning portal. Starting from this existing background, we intend to experiment various ways to support collaboration among users interconnected by mobile technologies through the already active portal based on our LIS.

The adaptation of the Learning Information System to info-mobility will need different steps:

a) Extension of the traditional functions of a Learning management system to the mobile-computing needs required by the project. This will imply the creation of teacher-system-student interaction tools mainly based on SMS messages concerning the activities of these actors in the system. Moreover, the portal will provide an access point to the system's actors, in order to download the educational material and the self-evaluation tests produced according to the objectives of the project. Besides, different structures will be created to support the research activities, like forums usable via mobile technologies, mailing lists for the various users, management of some virtual communities (students enrolled in a course, participants to laboratories etc.).
b) Distribution of the educational material specifically created for the fruition on mobile equipment. This will regard both the educational materials and the self-evaluation tests created in point c)
c) Integration of the self-evaluation system into the LIS. This system will allow conducting tests on the main platforms that currently equip the most widespread PDAs on the market. The choice of producing self-evaluation applications for both the PDAs environments is because we want to extend as much as possible the experiment, and most of all we want to create a self-assessment mechanism that must be generalized as much as possible with respect to technological platforms, due to the extreme volatility of the market.

As regard as the development of the systems, we decided on which devices to concentrate our development. This is a very important issue, as the market is continuously changing with new products emerging everyday. So, it is practically impossible to have a general mechanism for involving all possible devices currently available. We found the following devices useful for our experimentations:

- GSM/GPRS cellular phones
- PDA
- Smart-phones
- UMTS telephones
- Tablet PCs

The platforms have been already found in their main components. These platforms will be the ones based with Symbian OS on one side (this means to involve the whole cellular phones market with the biggest world producers), and on the other side the platforms equipped with Windows CE, i.e. the PDAs that present points of contact with the Windows desktop environment in terms of applications and working environment. We will also experiment with the Palm OS, so that our experiment will cover a very large share of the market. In the first step of the project, however, the choice made on some Microsoft™-dependent PDAs is related mainly on the consideration that most of the educational material is currently published in Microsoft™ software tools, especially PowerPoint and Word. In this sense, a device equipped with Microsoft™ operating systems will facilitate the interchange of educational material already available. However, the modular structure of the approach followed in the building of web services based on XML and SOAP will provide a sufficient grade of extensibility of our mobile platform to other PDAs, like those that are equipped with Symbian OS.

The test of the system will consist in some lessons conducted using Learning objects distributed using the LMS and used by students and teachers using PDAs, traditional viewers (like PowerPoint and Acrobat Reader) and other available mobile devices. Part of these educational materials will be available only through mobile devices: students will have to learn studying only on PDAs. In this way, different groups that have studied on different devices with different approaches will be available for our research: those who followed face-to-face lessons, those who studied on learning objects without following the lessons and those who studied on mobile devices. By creating a specific and calibrated set of tests, we want to verify the level of learning of the single groups, by analyzing the differences and the relative motivations. The results of these tests will be matched with the results of the self-evaluation tests distributed to the students, in order to verify thoroughly the level of learning reached by the students. The reactions of the students will be also analyzed, especially those related with problems in studying with a new but limited tool like a portable device. For this purpose, a forum on the web will be specifically activated, and some tutors will be available in order to help students with practical or technical problems.

As regarding the use of specific tools available with mobile technology, the most evident problem we faced in the design phase was the choice of the technology by which building the tools provided to the client in order to use our services. The current project provides ten different classes of services to mobile users, but in order to simplify the choice, we decided to concentrate initially on two different services for mobile devices:

- The management of SMSs sent by teachers to students or by administrative staff to teachers and students when particular events happen (meetings, reminder for expiration dates etc.)

- The consultation of a common agenda (we call it organizer) that will be available on the mobile device and will keep all the important dates for the actor (mainly students and teachers)

The first service is quite simple to build but not so easy to manage, if the LMS that operates behind the scenes does not have all the information needed. The main problem has been found in allowing the right person to send and receive SMSs, and in granting this permission inside correct boundaries, in terms of number of SMSs sendable by the user. The second service is under development and is more complicated, as it involves one of the most difficult task to manage inside a LMS, i.e., time management. We are currently building a system that allows students and teachers to connect with their mobile device and consult their agenda, dynamically built with all the events that could happen during a normal university activity. This implies a great effort of abstraction and integration between the LMS platform and the mobile devices. We have evaluated five different alternatives to build the interaction between the PDA (the platform chosen for the experimentation) and the central database. The problem is related to the way the client (the PDA) interrogates the remote server module requesting the update of the events since last connection. These are the alternatives we evaluated and tested, from the simplest to the most complicated:

- Using the embedded browser of the PDA to navigate through the web pages that web users will see using the traditional browser available for desktop PCs. This is the simplest solution, both for the users and for the development team. Only a particular attention to screen adaptation is necessary, in order to concentrate the most important information on the left-uppermost part of the screen and to avoid the necessity of frequent scrolling. The web page will be created using device-specific tags and languages, like the .NET™ mobile toolkit, in order to navigate through the data available on the server. However, we decided not to follow this solution as the primary one, because of the necessity for the user to be constantly connected to the Internet to navigate through the organizer, thus requiring permanent connections (like WI-FI settings) or a significant expense for the students and the teachers when connected to the net using GPRS technology. In Italy this solution is very costly at the moment, and WI-FI technology with wireless LAN is still in its infancy. Other short-range connection solutions have been abandoned, as we want this service to be used outside the campus.
- Using a client database application, built specifically for mobile devices, that interrogates the server DB through the internet, synchronizing the data on the mobile device. This is a proprietary solution bounded to the back-end DB used and the availability of a Internet connection on the PDA, that requires also quite complicated settings from a end-user perspective. However, from our tests, this solution has the advantage of dramatically boosting performance, thus reducing connection times.
- Synchronizing the PDA with the central database and the agenda of the user by using cradles and database synchronization: this solution will solve a lot of issues, but creates a problem in terms of cradle availability around the campus, and especially the problem of supporting different cradles for different models of PDA.
- Building a client/server application in which the client (on the PDA) uses traditional RPC/RMI mechanisms to invoke server methods in order to receive data. This has the advantage of requiring short-time connection to the central system,

and could be personalized to the PDA device. The disadvantage of this solution is the proprietary mechanism of communication between server and client, and also the necessity of using particular TCP/IP – UDP ports that could complicate the management of security on the server side due to firewalls.

- Building a web application that request a web service through the use of XML/SOAP messages to the server. This is the best solution we found, as it provides the access in short time to the central database through the use of open technology like XML/SOAP, will use a port that is already opened for web access, and finally will guarantee the extension of the client part to other PDAs simply by creating the new client interface to the web service. We will therefore provide the agenda synchronization through a web service that will recognize the user, verify the state of his/her agenda, and will send an XML-formatted packet of data regarding last events in the system. The client side of the application, specific for the device, will format this data for the display: after that, the connection with the server will be closed and the navigation on the agenda will be completely off-line.

References

1. Waycott J., *An Investigation into the Use of Mobile Computing Devices as Tools for Supporting Learning and Workplace Activities* , 5th Human Centred Technology Postgraduate Workshop (HCT-2001), Brighton, UK, September 2001, available online at http://www.cogs.susx.ac.uk/lab/hct/hctw2001/papers/waycott.pdf
2. Roibás A.C., Sánchez I.A., *Design scenarios for m-learning*, Proceedings of the European Workshop on Mobile and Contextual Learning, pp. 53-56, Birmingham, UK, June 2002
3. Rogers T., *Mobile Technologies for Informal Learning – a Theoretical Review of the Literature*, Proceedings of the European Workshop on Mobile and Contextual Learning, pp. 19-20, Birmingham, UK, June 2002
4. Dvorak J. D., Burchanan K., *Using Technology to Create and Enhance Collaborative Learning*, Proc. of 14th World Conference on Educational Multimedia, Hypermedia and Telecommunications (ED-MEDIA 2002) , Denver, CO, USA, June 2002
5. Kukulska-Hulme A., *Cognitive, Ergonomic and Affective Aspects of PDA Use for Learning*, Proceedings of the European Workshop on Mobile and Contextual Learning, pp. 32-33, Birmingham, UK, June 2002
6. Waycott J., Scanlon E., Jones A., *Evaluating the Use of PDAs as Learning and Workplace Tools: An Activity Theory Perspective*, Proceedings of the European Workshop on Mobile and Contextual Learning, pp. 34-35, Birmingham, UK, June 2002
7. Keegan D., *The future of learning: From eLearning to mLearning*, available online at http://learning.ericsson.net/leonardo/thebook/book.html
8. Trifonova, A and Ronchetti M,. *Where is m-learning going*?, in Proceedings of E-Learn 2003, Phoenix, Arizona, USA November 7-11, 2003.
9. Trifonova, A., *Mobile Learning - Review of the Literature*, DIT Technical Report DIT-03-009, available online at http://eprints.biblio.unitn.it/archive/00000359/01/009.pdf
10. *Mobile Learning Explorations at the Stanford Learning Lab: A newsletter for Stanford academic community*, Speaking of Computers, Issue 55, January 8, 2001, available on line at http://acomp.stanford.edu/acpubs/SOC/Back_Issues/SOC55/#3
11. Trifonova, A, Knapp, J., Gamper, J. Ronchetti M., *Mobile ELDIT: challenges in the transition from an a-learning to a m-learning system,* in printing, University of Trento
12. HandLeR project web site: http://www.eee.bham.ac.uk/handler/default.asp

13. Ketamo H., *mLearning for kindergarten's mathematics teaching*, Proc. of IEEE International Workshop on Wireless and Mobile Technologies in Education (WMTE 2002) , pp. 167-170, Växjö, Sweden, August 2002
14. Stone A., Briggs J., Smith C., *SMS and Interactivity – Some Results from the Field, and its Implications on Effective Uses of Mobile Technologies in Education*, Proc. of IEEE International Workshop on Wireless and Mobile Technologies in Education (WMTE 2002) , pp. 147-151, Växjö, Sweden, August 2002
15. Stone A., Briggs J., *ITZ GD 2 TXT – How to Use SMS Effectively in M-Learning*, Proceedings of the European Workshop on Mobile and Contextual Learning, pp. 11-14, Birmingham, UK, June 2002
16. Seppälä P., *Mobile learning and Mobility in Teacher Training,* Proc. of IEEE International Workshop on Wireless and Mobile Technologies in Education (WMTE 2002) , pp. 130-135, Växjö, Sweden, August 2002
17. Garner I., Francis J., Wales K., *An Evaluation of the Implementation of a Short Messaging System (SMS) to Support Undergraduate Students*, Proceedings of the European Workshop on Mobile and Contextual Learning, p. 15-18, Birmingham, UK, June 2002
18. Divitini M., Haugalokken O. K., Norevik P., *Improving communication through mobile technologies: Which possibilities?*, Proc. of IEEE International Workshop on Wireless and Mobile Technologies in Education (WMTE 2002) , pp. 86-90, Växjö, Sweden, August 2002
19. Sariola J., Sampson J. P., Vuorinen R., Kynäslahti H., *Promoting mLearning by the Uni-Wap Project Within Higher Education*, Proc. of International Conference on Technology and Education (ICTE 2001), available online at http://www.icte.org/T01_Library/T01_254.pdf
20. Sariola J., *What are the limits of academic teaching? - In search of the opportunities of mobile learning*, TeleLearning 2001 Conference, Vancouver, Canada, available online at http://ok.helsinki.fi/tekstit/Article.rtf
21. Sariola J., *What Are the Limits of Academic Teaching? – In Search of the Opportunities of Mobile Learning*, available online at http://ok.helsinki.fi/pdf_tiedostot/mobileEN.pdf
22. Seppälä P., Sariola J., Kynäslahti H., *Mobile Learning in Personnel Training of University Teachers*, Proc. of IEEE International Workshop on Wireless and Mobile Technologies in Education (WMTE 2002) , pp. 23-30, Växjö, Sweden, August 2002
23. Traxler J., *Evaluating m-learning*, Proceedings of the European Workshop on Mobile and Contextual Learning, pp. 63-64, Birmingham, UK, June 2002
24. Collett M., Stead G., *Meeting the Challenge: Producing M-Learning Materials for Young Adults with Numeracy and Literacy Needs*, Proceedings of the European Workshop on Mobile and Contextual Learning, pp. 61-62, Birmingham, UK, June 2002
25. Steinberger C., *Wireless meets Wireline e-Learning*, Proc. of 14th World Conference on Educational Multimedia, Hypermedia and Telecommunications (ED-MEDIA 2002) , Denver, CO, USA, June 2002
26. Figg C., Burston J., *PDA Strategies for Preservice Teacher Technology Training*, Proc. of 14th World Conference on Educational Multimedia, Hypermedia and Telecommunications (ED-MEDIA 2002) , Denver, CO, USA, June 2002
27. Casadei G., Magnani M., *Assessment strategies of an intelligent learning management system*, accepted for publication in "International Conference on Simulation and Multimedia in Engineering Education, 2003" conference proceedings
28. A'herran A., *Integrating a course delivery platform with information, student management and administrative systems*, in Proc. EDMedia 2001, Tampere, Finland, June 25-30 2001
29. Hall B, *Learning Management Systems. How to Choose the Right System for your Organisation*, Brandon Hall, 2001

30. McMahon M., Luca J, *Courseware Management Tools and Customised Web Pages: Rationale, Comparisons and Evaluation*, Proc. EDMedia 2001, Tampere, Finland, June 25-30 2001
31. Hanna, D. E., Glowacki-Dudka, M. & Conceicao-Runlee, C., *147 Practical tips for teaching online groups: Essentials of Web-based education*, Madison, WI: Atwood Publishing.
32. Aggarwal, A. *Web-based learning and teaching technologies: Opportunities and challenges*, Hershey, PA: Idea Group Publishing 2000.
33. Colazzo L., Molinari A. *From Learning Management Systems To Learning Information Systems: One Possible Evolution Of E-Learning*, in Proc. Communications, Internet and Information Technology (CIIT) Conference, St. Thomas, USA – November 18-20, 2002
34. Huba, M.E. & Freed, J. E., *Learner-centered assessment on college campuses. Shifting the focus from teaching to learning*, Needham Heights, MA: Allyn & Bacon.
35. Dalziel, J. R., & Gazzard, S., *Beyond Traditional Use of Multiple Choice Questions: Teaching and Learning with WebMCQ Interactive Questions and Workgroups. Open, Flexible and Distance Learning: Challenges of the New Millennium,* Collected papers from the 14th Biennial Forum of the Open and Distance Learning Association of Australia, pp.93-96. Geelong: Deakin University.

E-Mail on the Move: Categorization, Filtering, and Alerting on Mobile Devices with the ifMail Prototype

Marco Cignini, Stefano Mizzaro, Carlo Tasso, and Andrea Virgili

Department of Mathematics and Computer Science,
University of Udine
Via delle Scienze, 206 – Loc. Rizzi – Udine – 33100 Italy
{cignini,mizzaro,tasso,virgili}@dimi.uniud.it

Abstract. We propose an integrated approach to email categorization, filtering, and alerting on mobile devices. After a general introduction to the problem, we present the ifMail prototype, capable of: categorize incoming email messages into pre-defined categories; filter and rank the categorized messages according to their importance; and alert the user on mobile devices when important messages are waiting to be read. The second part of the paper describes an extended evaluation of the ifMail prototype, whose results show the high effectiveness levels reached by the system.

1 Introduction

Information overload is the main problem for information access users: we are overwhelmed by too much information when we browse the Web, when we analyze the results of a search engine, when we use a directory, when we read the messages in a forum or in a newsgroup, and when we use electronic mail. Electronic mail, historically one of the first services made available by the Internet to the large public audience, is today one of the major activities of Internet users. All of us rely on email as one of the primary communication methods, both at work and at home: email has, at least partially, supplanted paper mail, messages, and telephone conversations.

Email overload is an important facet of information overload: the average user receives dozens of messages per day, and the trend is not slowing down at all [32]; some of us are lucky and receive a manageable number of email messages per day, whereas others are completely overwhelmed; unsolicited email, usually called spam or junk-mail, is constantly and worryingly increasing.

Usage of email is a highly personalized activity, and people use email in amazingly different ways [14]. People read emails with different strategies: *archivers* choose strategies that allow them to read everything and not miss anything important, and *prioritizers* want to limit the time spent on email reading to switch to "real work" [11]. Accordingly to Whittaker and Sidner [33], people can be divided into *no filers* (that keep all the messages in their inbox), *frequent filers* (that constantly clean up their inbox), and *spring cleaners* (that clean up their inbox once every few months).

Also, email software tools (Eudora, Outlook, Mozilla, to name just a few) are used not only in the standard ways foreseen by email tools designers, i.e., for reading and answering messages, but also in more "perverted" ways. We refer here to archiving,

F. Crestani et al. (Eds.): Mobile and Ubiquitous Info. Access Ws 2003, LNCS 2954, pp. 107–123, 2004.

managing a personal agenda or serving as a reminder tool: people send mail to themselves as a reminder; people use the inbox message list as an agenda; people use email for task management and delegation; people hit reply for avoiding to type in a long list of addresses; people archive a whole message when the attachment is an important document; people use email as a file transfer mean; and so on. This creative use of email has generated another meaning for the "email overload" expression [32], i.e., the overloading of uses of this tool, and because of this phenomenon, email has been named a serial-killer application [10].

In this scenario, advanced tools for email processing are desperately needed: threading, categorizing, archiving, filtering, alerting, and perhaps more. Today's email clients provide these functions in a rather limited way. Mail tools allow to view the messages sorted by date, by thread, by sender, etc. Users can manually categorize the messages, usually by drag-and-drop in one of a hierarchy of folders. A priority flag can be manually attached to a message by the sender, and shown to the receiver by the mail client. Filters based on pattern matching rules on (mainly) the structured part of messages (i.e., subject, sender, date, priority, size, etc.) can be manually defined by the user to automatically move the received messages in the appropriate folder (and to execute other operations on the message). Automatic anti-spam filters, to filter out spam exploiting some learning techniques, are common in many mail tools. All email tools can notify the user sitting in front of his/her desktop that new mail has arrived by visual and/or sound messages.

These activities are both time consuming and rather ineffective: manually defining a filter and managing a set of several filters puts a higher cognitive load to a user engaged in other activities and, often, the decision whether a message is interesting, junk, belonging to a certain topical category, and so on cannot be taken only on the basis of the structured part of the message but it has to be taken also on the basis of message body, attachment, meaning, and even context (i.e., the thread to which the message belongs, the current situation in which the user is, and so on). Also, alerting is rather neglected: having only a visual and/or acoustical "You have new mail" notification on our desktop is a rather poor way of communication, that ignores both the cognitive situation of the user, like his/her current task or degree of attention, and features of the message like its urgency, the sender, the topic, and so on.

The coming of portable devices (cell phones, PDAs, pagers, and so on), that are enabled to various network connection modes (GSM, GPRS, UMTS, Wi-Fi, Bluetooth, etc.), is a new and important variable to add in the above sketched scenario. There are several issues that need to be addressed. The new environment implies both limitations to be taken into account and opportunities to be exploited; therefore, simply replicating the non-mobile approach in the mobile world would lead to far from ideal solutions. For instance, using a mobile device to access one own email inbox via standard protocols like POP or IMAP is an unsatisfying solution that neglects both the always-on modality of a user empowered with a mobile device, and the cost usually implied by data transmission on a wireless connection. The usually complex user interfaces of mail tools cannot be replicated on small-screen devices, so it is much more difficult to have ease of reading and user's feedback (e.g., explicit feedback of relevance, categorization, importance, and urgency of a message is likely to be replaced by more implicit kinds of feedback, perhaps exploiting the time that a message waits in the "unread" status). The interaction modes requiring continuous attention (e.g., drag-and-drop), that are common for desktop-based tools, are not adequate for devices used out there in the real world, with several sources of distraction.

Notifications could and should be delivered on the nowadays widely available smaller and portable devices with the most appropriate modality (WAP-push, SMS, etc.). Notifications should be done depending on features of the received messages like their number, their importance, the category they pertain to, and so on. The well known limitations on bandwidth, screen size, and user cognitive load (time, distraction level, and so on) make extremely important to have a *selective* alerting functionality, capable of notifying the user only when really important messages arrive: not only the notification of a spam message would be very unpleasant for the user, but also the notification of a "normal" message when the user is in a particular context (e.g., while driving, or engaged in a meeting, or in an important phone conversation) can be unpleasant as well. The mobile world requires an integrated solution, exploiting categorization, filtering and alerting.

Moreover, in the mobile world, categorizing, filtering, and alerting will have an increased importance, since accessing email by a mobile device is more critical in many respects. People carry with themselves their mobile devices, that are therefore much more intrusive than a standard desktop: the "new mail" sound that might be an acceptable interruption when sitting in front of a desktop computer, is likely to be very annoying while engaged in real-world critical activities.

Turning our attention to more technical issues, we notice that new mail tools and protocols might be designed to allow the user (both as a sender and as a receiver) to specify (manually, semi-automatically, or automatically) the alerting modalities of certain message categories. Complex engineering solutions are needed because the limited storage and computational power available today on the mobile devices, and the bandwidth limitations, suggest a server side based solution, in which most of the computation takes place on the server and the data transmission on the mobile device is limited.

Also, the integration of all the devices that one can use to read his/her own email messages (desktop PC, mobile devices, internet points, etc.) is another interesting, and difficult problem, and reinforces the requirement for server side based solutions. A further kind of integration is that among all the different kinds of messages that the user of a mobile device can receive: besides email-like messages, we have SMS, EMS, and MMS (and perhaps more in the future). The integration of all these message services is a difficult problem as well.

Finally, the increased email access by mobile devices will change the people usage of email: nobody can predict all the range of new "perverted" or "creative" uses that mobile device users could imagine and adopt when mobile email tools will be broadly available (e.g., the sending of email to oneself as a remainder is likely to become much more frequent).

All these issues constitute a research agenda for the years to come, and need to be tackled from an interdisciplinary standpoint: user modeling, information retrieval and filtering, human computer interaction, software engineering, are all disciplines that can contribute to the development of more effective email tools for the mobile and wireless world. In this paper we do not present a final and general solution. Rather, our aim is twofold: (i) to show how to improve and make (at least partially) automatic the tasks of email categorization, filtering, and alerting; and (ii) to show how to integrate these new and more effective tools in the mobile scenario, where people access email while on the move. The paper is structured as follows. In Section 2 we highlight the main issues related to email categorization and filtering. We also survey the literature, briefly describing the relevant work that has been proposed so far. In Section 3

we describe the ifMail prototype, from both conceptual and technical perspectives. In Section 4 an extended experimental evaluation of the effectiveness of our approach is presented. Section 5 closes the papers and sketches future developments.

2 Categorization and Filtering of Email Messages

Text categorization (or classification) is the grouping of documents into predefined categories [28]. State-of-the-art classifiers automatically built by means of machine learning techniques show an effectiveness comparable to manually built classifiers.

Email messages are very heterogeneous. Examples of variables that can range over rather wide set of values are: length, language(s) used, importance of the contained information, presence/absence of attachments of various kinds, formal/informal tone, emoticons, jargon. Also structured data contained in the header like date, sender, subject, number of recipients, are bound to wide variations. Given the peculiar nature of email messages, email categorization is a very particular case of general text categorization.

Various approaches, mainly derived from the experiments on generic text categorization, have been applied to email categorization [9]: Cohen [7] uses the RIPPER algorithm; Payne and Edwards [24] compare CN2 (a rule induction algorithm) with IBPL1 (a modified version of K-nearest Neighbor algorithm using memory based reasoning); Rennie [25] exploits naïve Bayes classifiers; Segal and Kephart [29] develop a system for semi-automatic categorization (i.e., the system proposes to the user three alternative folders for each message) based on TF-IFD; Brutlag and Meek [4] compare Linear Support Vector Machine, TF-IDF, and Unigram Language Model, and obtain that no method outperforms the others; McCreath and Kay [14] show how the combination of hand crafted and learnt rules is more effective than either approach working alone. All these approaches show rather similar results, with accuracy (percentage of messages classified in a correct way) around 70%-80%. An even more difficult problem, the clustering of email messages (i.e., given a set of email messages, extract the categories and classify the messages in the found categories), is tackled in [13].

Spam (or junk) email filtering has seen an increasing interest in last years, due to the increasing amount of unsolicited emails: Pantel and Lin [19] and Sahami et al. [27] exploit naïve Bayes classifiers; Adroutsopoulos et al. [1] use a memory-based (or instance-based) approach, implemented as a variant of the K-nearest neighbor (K-nn) algorithm; Carreras et al. [5] rely on the boosting algorithm AdaBoost to find a highly accurate classification rule by combining many weak rules.

Anti-spam filtering has been approached as a separate problem from email categorization, even if, at first glance, it seems just a 2-categories categorization problem. However, anti-spam is an easier problem than categorization not only because it handles just two categories, but also because the two categories are rather well defined (it is rather easy to define spam), clear-cut (it is rather easy to sort out spam from non-spam), and objective (usually, what is spam for one user is spam for everybody). In turn, email categorization is highly subjective: each user can choose rather different criteria for creating the categories (e.g., some users divide messages on the basis of the sender, others on the basis of the topic, others on the basis of their a-priori categorization of their job activity, and so on); the number of categories can vary a lot

among users; the categories are sometimes not well defined (users can be very well organized or completely chaotic); and so on. Therefore, it is quite likely that a single fit-for-all email categorizer is not feasible, and that hybrid approaches are needed. Indeed, even if it is difficult to have a definitive comparison between the effectiveness of anti-spam filters and of email categorizers because of the high differences in the collections used, in the number and features of categories, and so on, it is evident that anti-spam filters effectiveness is rather higher (95% precision) than the more general email categorization problem.

The alerting problem is much less studied than email categorization and filtering: further research in terms of notification modalities, prototype implementation and evaluation, and user studies is needed. It seems anyway obvious that only important messages should be notified on mobile devises, to avoid high cognitive loads and distraction on the user. Therefore, an integrated solution, comprising categorizing, filtering and alerting is required.

The evaluation of the effectiveness of an email tool is not simple at all. The most naïve approaches show several limitations. Relying on general test collection like TREC (http://trec.nist.gov/) is not adequate, since the peculiar nature of email makes an email message different from a generic document. Usenet news seem more similar, but again differences do exist: for instance, an email message body usually starts with the name of the recipient, whereas this is obviously less frequent for Usenet messages.

Privacy is also an important issue: since email messages contain private data, few people are willing to make public their messages; perhaps those people will anyway clean some of the more compromising and confidential messages, thus making available only a portion of their message archive, that is not a good sample at all; anyway, people willing to make public their email archives are not a good sample for sure, since people that are more reserved are completely left out; and relying on messages archives of mail lists leads again to a biased sample.

3 The ifMail Prototype

At the Udine University we have started to study some of the above described issues and, on the basis of our work in the last 10 years, we have developed the ifMail prototype. ifMail handles, with a content based approach, categorization, filtering of email messages, and alerting on mobile devices. ifMail overall operation is shown in Fig. 1. The messages in the incoming stream are processed to extract the internal representations used in subsequent steps. The internal representation contains term/weight (weight representing the importance of each term) pairs, corresponding to both the structured part and the body of the email message. Categorization is obtained on the basis of a profile attached to each user-defined folder and dynamically updated by means of user's feedback. The profile contains two parts: a frame for the information included in the structured part of email messages, and a semantic network for the conceptual content of the body of messages [16]. The profile is matched with the internal representation of the incoming messages and the message is classified accordingly to its content. The matching takes into account both the structured and unstructured parts of email messages. Filtering, performed by re-using the evaluation made in the categorization phase, singles out the most relevant messages in each folder and alerting takes charge of notifying these messages to the user's mobile device. Our notion of filtering is therefore more general than just anti-spam filtering: ifMail tries

to associate to each message a numeric figure representing the importance that the message has for the user.

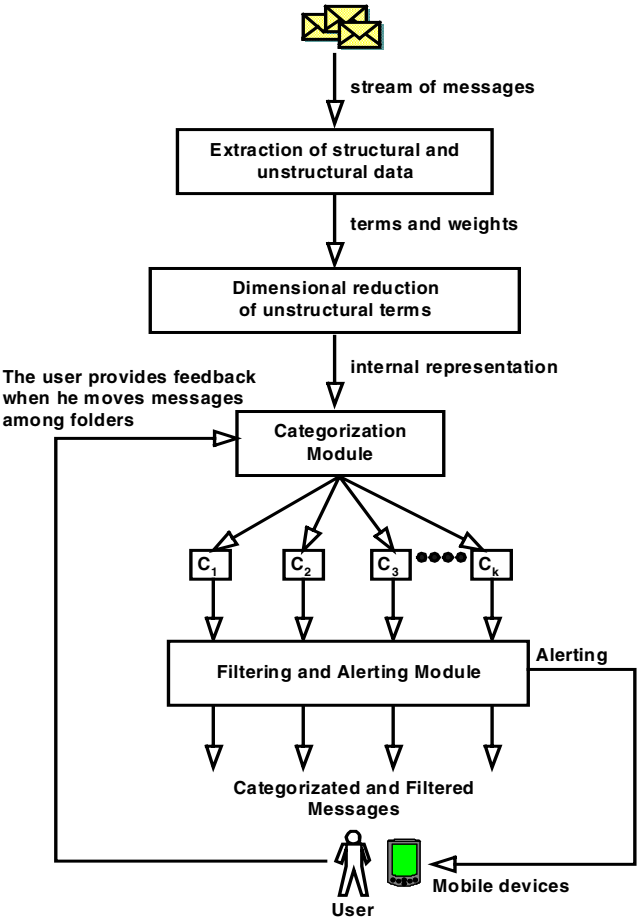


Fig. 1. Conceptual model of ifMail operation.

ifMail categorization and filtering are based on the IFT (Information Filtering Tool) system [14,16], capable of profile building, storing, and matching. IFT has been developed on the basis of the UMT (User Modeling Tool) shell [0] and has been applied to a variety of systems and domains, e.g., Web filtering [2], filtering of enterprise documents [30], and filtering of scholarly publications [17]. IFT matches the profile associated to each category with the internal representation of each message and returns a result made up of three values:

1. *Coverage*: the percentage of the most relevant concepts of the profile which are also present in the documents, computed taking into account also their weights.
2. *Match*: a measure of how much the concepts of the profile are present in the document (i.e., they are more or less numerous in the document).
3. *Rank*: a synthetic value (ranging from 0 to 5), which is obtained as a combination of the previous two values.

Categorization is performed on the basis of all three values; filtering is based on Rank score only.

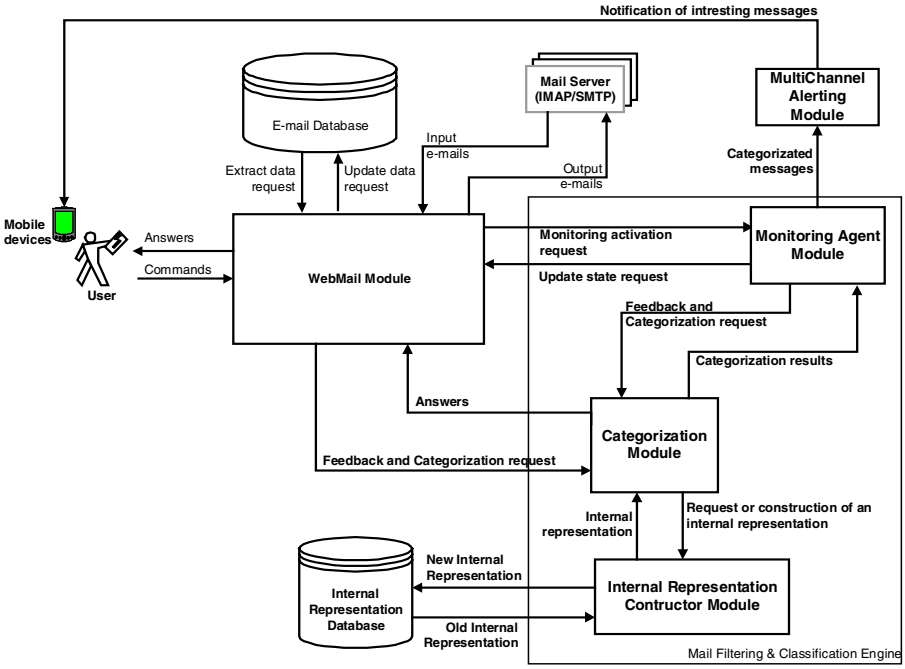


Fig. 2. ifMail overall architecture.

Fig. 2 shows the overall architecture of the ifMail system. The main modules are:

- WebMail, that allows the user to access email functionalities via a Web browser. It has been developed specifically for this project in order to connect and integrate categorizing, filtering, and alerting. More specifically, the WebMail module implements the only user interface of the system and it allows the configuration of the innovative services.
- Mail Filtering and Classification Engine, made up by the following three submodules:
    a) Monitoring Agent, that monitors the arrival of new messages and calls the categorization and filtering operations. ifMail supports POP and IMAP servers, and any number of email accounts.
    b) Internal Representation Builder, that parses the text of message subject and body, removes stop words, extracts the stem of the terms, and builds the internal representation of the message, stored in the Internal Representation Database.
    c) Categorization, that executes categorization and handles feedback data. This module contains, and relies on, the IFT submodule: IFT compares the internal representation of the incoming message with each category profile, and modifies the category profile according to user's relevance feedback.

- Multi Channel Alerting, that, on the basis of the categorization results and of user's personalized settings, notifies immediately to the user the most relevant messages via a mobile device.

Fig. 3 shows a snapshot of ifMail Web user interface: a quite standard email interface that allows standard mail management and that provides the commands and visualization items relevant to the new categorization and filtering features. The number of stars associated to each message is given by the Rank score associated to the message.

The PDA screenshots in Fig. 4 show the multi-channel alerting of ifMail: in the screenshot on the top, the notification of the arrival of a new relevant message for the "myWork" category is shown. The user can detect (by the number of stars) the message relevance computed by the system, he can archive the message, read message data like sender and subject, or read the whole message body (screenshot below).

The system has been developed with an XML-based technology to allow a higher flexibility for the presentation layer: multiple interfaces are generated by means of XSLT transformations, that produce the output in the markup language suitable for the requesting device by applying the corresponding style sheet to a common set of XML data. In such a way, the service is accessible by a wide range of devices such as PDAs, smartphones, and cell phones, provided that they comply to the WAP 1.2.1 or 2.0 standards [19, 21, 22, 25].

The interface design has been developed according to some guidelines for information access with mobile devices [5, 7, 11, 20]. The navigation through the pages of the service has been designed considering the physical interface used for the interaction with the device. Moreover, the complexity and extension of every page of the service are adapted to the dimension and capabilities of the display of the mobile devices. From a functional point of view, the interfaces are down-scaled (going from the PC version to the WAP version) to reduce the complexity of the service, considering the limited resources of the devices and the mobile context of use of the service.

4 Experimental Evaluation

We have discussed in Section 2 the intrinsic limitations in the evaluation of advanced email tools, and some of the issues that make the evaluation of these tools a difficult task. In order to overcome these limitations, we have designed and carried out an extensive evaluation of the ifMail prototype, taking also into account previous experimental work carried out in recent years in our laboratory. The goal of the experimental activity has been the evaluation of categorization, filtering, and alerting capabilities of ifMail. We have run various simulations on 6 collections of email and newsgroups messages (Table 1). We have used the term "simulation" since the experiments have been performed in a simulated environment in which the typical actions that a user could perform on ifMail can be repeated at will, without engaging (and overloading) real users.

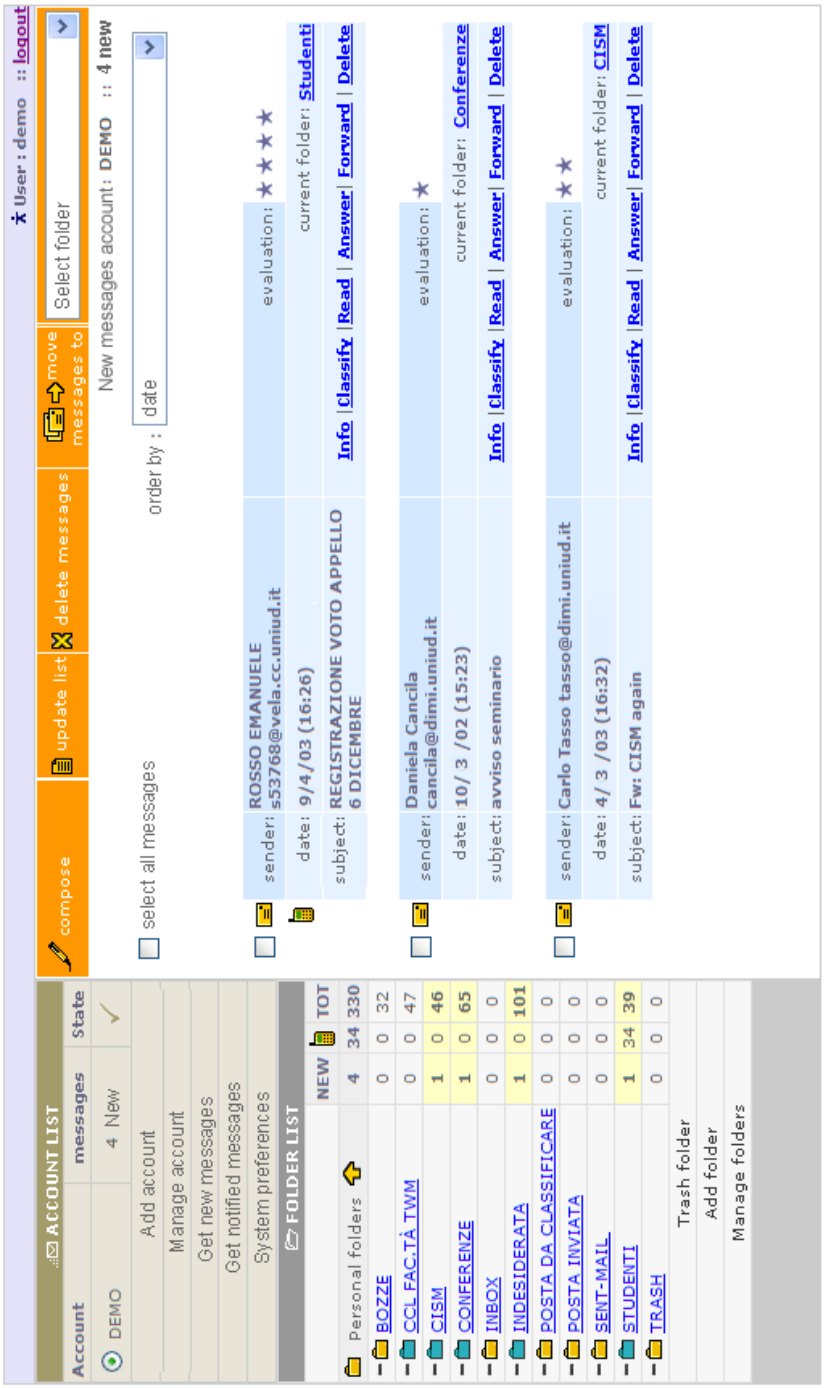


Fig. 3. ifMail user interface for Web mail.

Fig. 4. ifMail user interface: email reading on a PDA (left) and folders of new categorized messages on an Openwave WAP phone simulator (right).

Obviously, with this approach, we have intentionally not evaluated the usability of the user interface, nor we wanted to claim the effectiveness of our system in absolute terms. On the other hand, given the early development stage of the ifMail prototype, we were interested in evaluating some design decisions and in harvesting an experimental set of real data with a quick, light, and formative evaluation, capable of giving us hints on how to proceed with the development of the system.

Table 1 provides basic data on the six collections of email messages we have exploited: two of them come from real users, and include all the messages received over a period of about 30-40 days. All the messages received over that period were included, and none was eliminated. Both users (one of them is the third author of this paper) defined a set of categories (folders), to be used for evaluating the classification capabilities.

The collections extracted from newsgroups concern a similar number of messages and categories, with the exception of collection F, which is significantly larger and was considered for evaluating whether the results obtained with similar collections (A through E) were maintained in a much heavier situation.

Table 1. Email message collections used in the experiments.

Message kind	Collection	Number of categories	Total number of messages
Personal messages	**A**	9	540
	B	7	645
Newsgroups messages	**C**	7	525
	D	6	450
	E	7	540
	F	16	1309

We have defined two different modes of operation of ifMail usage:

- Mode *One-by-one*, in which ifMail provides only an advice: the user reading a message is shown a hint on which category(ies) are likely to be the correct destination of that message. By confirming or not confirming on each single message the (automatically) proposed categorization, the user provides relevance feedback, exploited by the system to update the relevant category profiles.
- Mode *Session*, in which ifMail automatically categorizes all the messages received during the current day (we have assumed daily batches of fixed size including 15 messages per day). The user provides relevance feedback only after all these categorizations have been done.

A first set of experiments concerned the comparison of these two modes of operation. The profiles associated to each folder were initially empty, and were incrementally built only through relevance feedback. Table 2 illustrates the average (over all the available collections) of precision, recall, and F1 measure [31, 34], where the results obtained for each category are combined using the micro-average indicator [28].

Table 2. Comparison between session mode and one-by-one mode.

	Session mode	One-by-one mode
Average Precision	75%	79%
Average Recall	72%	76%
Average F1	74%	78%

First of all, we notice that the values obtained are in the range from 70% to 80%. Other experiments reported in the literature [18, 28] concern the categorization of the Reuters-22713 collection (constituted by 21.450 articles, subdivided into 135 categories) or the Reuters-21578 collection (constituted by 12.902 articles, subdivided into 90 categories): the values obtained for the F1 measure are in the same range between 70% and 80%. We have considered this result as a confirmation of the adequacy of the baseline performance of ifMail. Furthermore, it should be highlighted that the values reported in Table 2 are average values, which include also the initial phases, where errors are most likely to happen: this implies that saturation ('steady state') values can be significantly higher.

Secondly, it can be noticed that precision reaches higher levels than recall. We can interpret this phenomenon in the following way: the number of messages considered (i) is capable of reducing the number of categorization errors, but, on the other hand,

(ii) is not sufficient for building profiles that cover all the concepts included in a category (and some message are not categorized, i.e. not assigned to any category). Finally, one-by-one mode outperforms session mode, reaching almost 80% in all the three considered indicators.

With reference to the same experiment, Fig. 5 shows the evolution (over the sequence of daily sessions and only for collection E) of the F1 measure. Both modes of operation reach values above 80%. The 70% level (conventionally taken as the value indicating the termination of the initial learning phase), is reached earlier in the one-by-one mode. In the long run the two mode of operation reach the same level of performance.

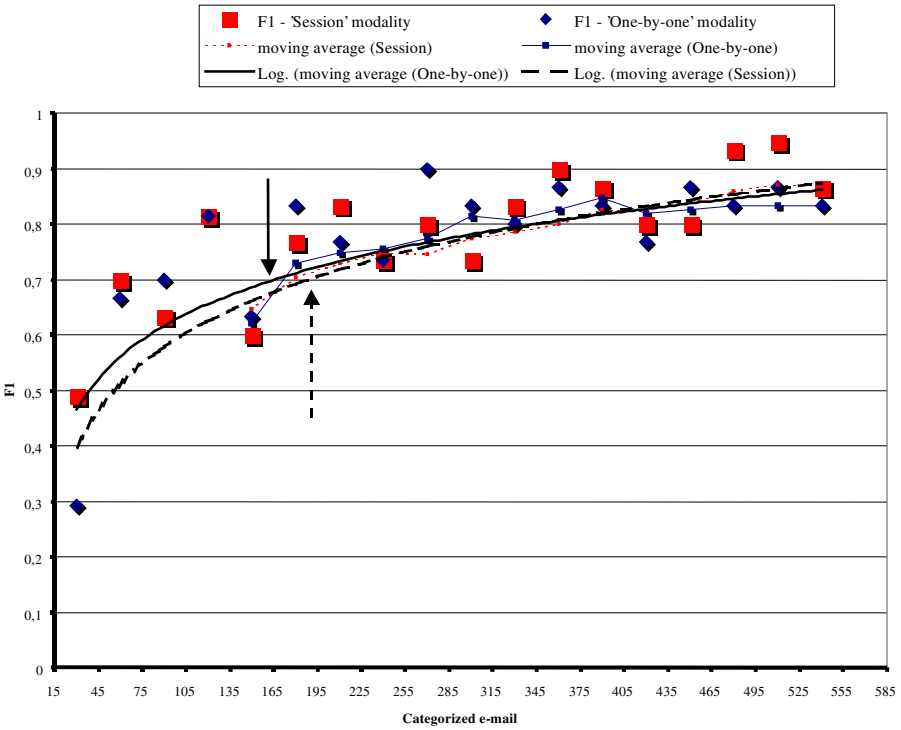


Fig. 5. Microaverage F1 in both operation modes for collection E.

Collections A and B, provided by real users, contained a Spam category, defined by the two users in order to collect all the 'not desired' messages (typically unsolicited advertising). In Fig. 6, we report the evolution over time of both precision and recall for the Spam folder of collection B. Precision reaches more than 95% and recall the range 70%-80%: this can be explained by the fact that when a Spam message is received, all the subsequent massages concerning the same topic will be detected, while new Spam topics are not known since never seen before, so they are left in the inbox, i.e., not categorized. This highlights a significant advantage of our content-based approach to Spam detection, in comparisons with standard anti-Spam systems based on an archive of spam messages: our system can detect any new Spam message

which concerns topics that previously have been already classified as Spam, independently from other facts (sender or subject already encountered or not).

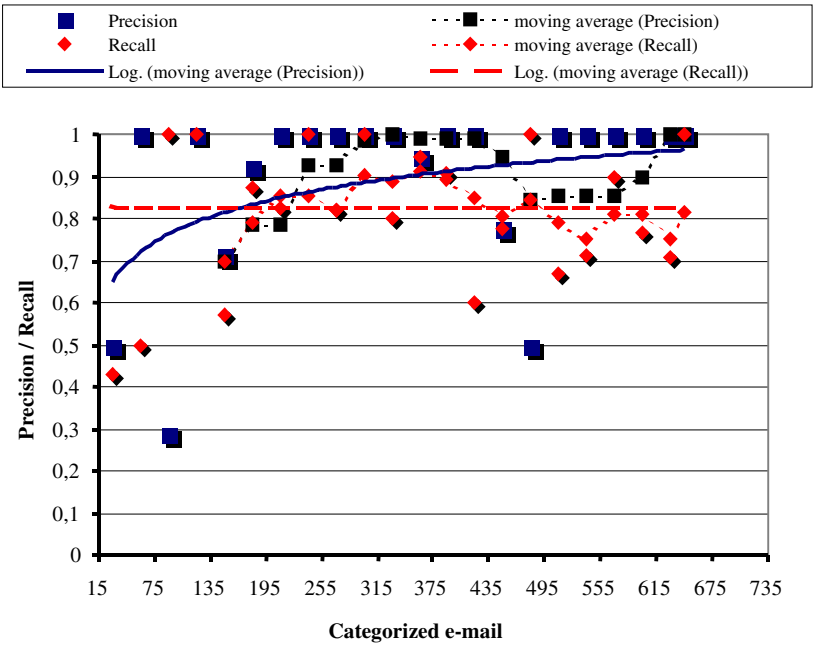


Fig. 6. Precision and Recall for the Spam category of collection B.

Another (expected) phenomenon observed in the experimentation concerns the relationship between performance and level of specificity of a category: whenever a category includes a well defined and limited topic, performance in terms of precision and recall is higher, reaching for both indicators the level of 85%. Analogously, for such categories, the learning phase is shorter.

Table 3 illustrates such a situation for some categories with this characteristics.

Other experiments have been focused on the identification of the best threshold to be employed for alerting. We have seen that using only the Rank value (an integer ranging from 1 to 5), precision was maximized (over 80%) and that, by increasing the specific value considered for the threshold, precision was further improved. Fig. 7 shows that the higher the threshold (4 or 5), the steeper is the learning curve, and higher are the precision values obtained (several values saturate at 100%).

Finally, we have computed a measure of the effort required to the user of ifMail, in terms of the number of 'move operations' of a mail message towards its correct folder (category). More specifically, we have considered successive groups of 60 messages (i.e., four days), and we have counted:

- the number of correct system categorization operations (green line on the top part of Fig. 8);
- the number of user moves, i.e., the explicit indication done by the user on a single message, since the system was not able to categorize the message correctly (red line in Fig. 8).

It is interesting to see that, as the user 'teaches' to the system how to categorize, the system 'learns'. After about 70 messages received, the user needs to move about 50% of the messages to their correct folder. After about 300 messages, the system 'has learned', and it is able to categorize correctly more than 50 messages out of the incoming 60, with a missed-categorization rate of less than 16%.

Table 3. Results for categories with well defined topic.

Collection	Folder	Precision	Recall	F1
A	*News*	0,91	0,83	0,87
B	*Students and courses*	0,94	0,93	0,93
	Department news	0,85	0,91	0,88
	Seminars	0,86	0,91	0,88
C	*ADSL*	0,92	0,92	0,92

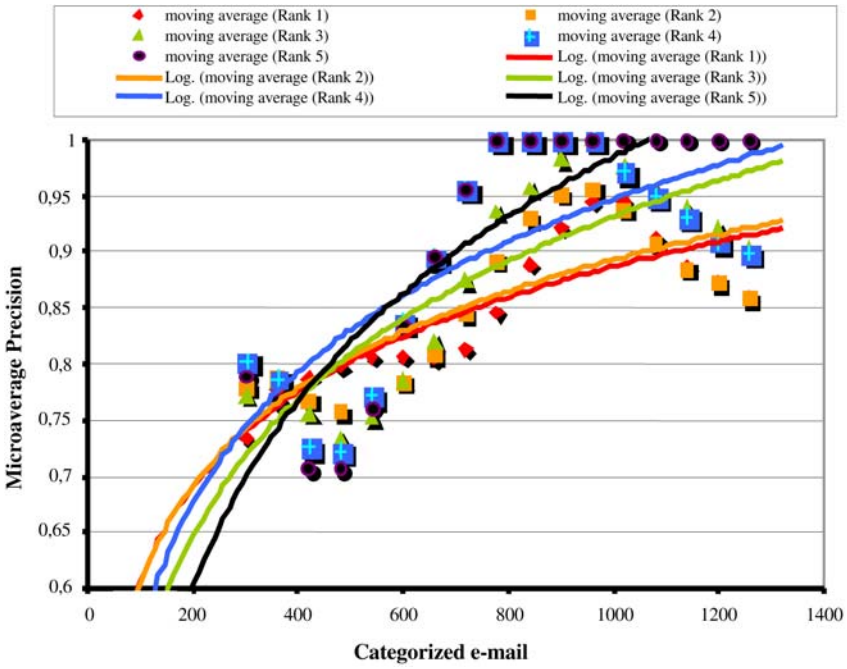


Fig. 7. Precision with different values as alerting threshold for collection F.

5 Conclusions and Future Work

We have discussed the issues of email categorization, filtering, and alerting. After a general introduction to the problem and a brief literature survey, we have presented the ifMail prototype, capable of: categorize incoming email messages into pre-defined categories; filter and rank the categorized messages according to their importance; and alert the user on mobile devices when important messages are waiting to be read.

We have also performed an extended evaluation of the ifMail prototype. The results show the high effectiveness levels reached by the system.

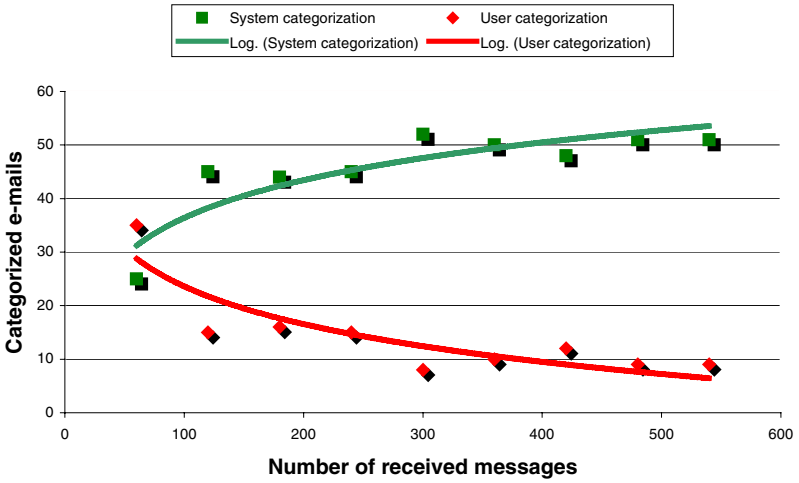


Fig. 8. Comparison of the number of user and system categorization actions (session mode).

We will continue this research in various ways. We are currently working at improving the ifMail prototype and we plan a more complete evaluation after these improvements. We intend to deal with privacy issues with a novel approach, by implementing a software capable of analyzing the email archives of users by running on their computers and simulating the behavior of a categorization algorithm. The categorization algorithm results should then be compared with the hand-made categorization and only the comparison results are made public. This software should be open source (to guarantee the privacy) and could be designed as a framework capable of hosting any categorization algorithm conforming to some well defined specifications. To take into account the time characteristics of messages (how long a message has been staying in the inbox, how long it has been in the unread status, for how long the user has not been checking his/her email, how much time the user spent in reading it, or in answering it, and so on) the software should also be capable of monitoring user's activity for a period of time.

Acknowledgements

We would like to thank Luca Chittaro and Paolo Dal Cin for their help in the design of the user interface on PDAs.

References

1. I. Androutsopoulos, J. Koutsias, K.V. Chandrinos, G. Paliouras, C.D. Spyropoulos, An Evaluation of Naive Bayesian Anti-Spam Filtering**.** *Proceedings of the Workshop on Machine Learning in the New Information Age, 11th European Conference on Machine Learning (ECML),* pp. 9-17, Barcelona, Spain, 2000.

2. F. A. Asnicar., M. Di Fant, C. Tasso User Model-Based Information Filtering. In M. Lenzerini (Ed.) AI*IA 97: Advances in Artificial Intelligence - *Proceeding of the 5th Congress of the Italian Association for Artificial Intelligence*, Rome, I, September 17-19, 1997, Springer Verlag, Berlin, LNAI 1321, 1997, pp. 242-253.
3. G. Brajnik, C. Tasso, A shell for Developing Non-Monotonic User Modeling System*, International Journal of Human-Computer Studies*, vol.40, pp.31-62, 1994.
4. C. Brutlag, J. Meek, Challenges of the email domain for text classification, In *Proceedings of the Seventeenth International Conference on Machine Learning*, 2000, pp. 103-110.
5. G. Buchanan, M. Jones, H. Thimbleby, S. Farrant, M. Pazzani, *Improving mobile internet usability,* Proceedings 10th WWW Conf., ACM Press, New York, 2001, pp. 673-680.
6. X. Carreras, L. Marquez, Boosting Trees for Anti-Spam Email Filtering *Proceedings of RANLP-01, 4th International Conference on Recent Advances in Natural Language Processing*, Tzigov hark, BG, 2001.
7. L. Chittaro and P. Dal Cin , Evaluating interface Design Choices on WAP Phones: Navigation and Selection, *Personal and Ubiquitous Computing*, 6(4), 237-244, 2002.
8. W. Cohen, Learning Rules that Classify E-Mail, Papers from the AAAI Spring Symposium on Machine Learning in Information Access, 1996, pp. 18-25.
9. E. Crawford, J. Kay, E. McCreath, Automatic Induction of Rules for e-mail classification, *Proceedings of the Sixth Australian Document Computing Symposium*, Coffs Harbour, Australia, Dec. 7, 2001
10. N. Ducheneaut and V. Bellotti, Email as Habitat. *Interactions*, September/October 2001.
11. M. Jones, G. Buchanan, H. Thimbleby *Sorting Out Searching on Small Screen Devices*, In F. Paterno, (Ed.), Proceedings of the 4th International Symposium on Mobile HCI, Pisa, Italy, 2002, LNCS 2411, pp 81-94, Springer.
12. W. Mackay, Diversity in the Use of Electronic Mail: A Preliminary Inquiry. *ACM Transactions on Office Information Systems*, 6(4), 380-397, 1988
13. G. Manco, E. Masciari, M. Ruffolo, A. Tagarelli, Towards An Adaptive Mail Classifier, *Atti dell'Ottavo Convegno AI*IA 2002*, Siena, Italy, 2002, pp. 63.
14. E. McCreath, J. Kay, Iems: Helping Users Manage Email, In P. Brusilowski, A. Corbett, F. de Rosis, *User Modeling 2003 9th Intl. Conf. on User Modeling, UM2003*, Springer Verlag, LNAI 2702, 2003, pp. 263-272
15. M. Minio, C. Tasso, User Modelling for Information Filtering on Internet Services: Exploiting an Extended Version of the UMT Shell, *UM96 Workshop on "User Modeling for Information Filtering on the WWW"*, Kaiula-Kona, Hawaii, USA, January, 2-5, 1996.
16. M. Minio, C. Tasso, IFT: un'Interfaccia Intelligente per il Filtraggio di Informazioni Basato su Modellizzazione d'Utente, *AI*IA Notizie IX(3)*, 21-25, 1996.
17. S. Mizzaro and C. Tasso, Ephemeral and Persistent Personalization in Adaptive Information Access to Scholarly Publications on the Web. In P. De Bra, P. Brusilovsky, R. Conejo (Eds.), *Adaptive Hypermedia and Adaptive Web-Based Systems, Second International Conference AH 2002, LNCS 2347*, pages 306-316, Malaga, 29-31 May 2002. ISBN 3-540-43737-1
18. I. Moulinier, G. Raskinis, J. G. Ganascia, Text Categorization: a Symbolic Approach, *In Proceedings of the Fifth Annual Symposium on Document Analysis and Information Retrieval,* Las Vegas, April 1996, pp. 87-99.
19. Nokia Corporation, *WML to XHTML migration. Version 2.0*, http://www.nokia.com/, 2002.
20. Openwave Systems Inc., *Openwave Usability Guidelines for WAP Applications*, http://developer.openwave.com/support/techlib.html/, 2001.
21. Openwave Systems Inc., *Migrating to WML with GUI Extensions and XHTML Mobile Profile,* http://developer.openwave.com/support/techlib.html/, 2001.
22. Open Mobile Alliance. http://www.wapforum.org/.
23. P. Pantel, D. Lin, Spamcop: A spam classification & organization program, in *Proceedings of AAAI-98 Workshop on Learning for Text Categorization*, pages 95-98, 1998.

24. T. Payne, P. Edwards, Interface agents that learn: An investigation of learning issues in a mail agent interface, *Applied Artificial Intelligence,* Vol. 11, pp. 1-32, 1997.
25. Phone.com Inc., *WML Application Style Guide,* http://www.openwave.com/, 2000.
26. J. D. M. Rennie, ifile: An application of Machine Learning to E-Mail Filtering, *In Proceedings KDD00 Workshop on Text Mining,* Boston, 2000.
27. M. Sahami, S. Dumais, D. Heckerman, E. Horvitz, A bayesian approach to filtering junk e-mail, in *AAAI-98 Workshop on Learning for Text Categorization,* 1998.
28. F. Sebastiani, Machine Learning in Automated Text Categorization, *ACM Computing Surveys*, 34(1), 1-47, 2002.
29. R.B. Segal, J.O. Kephart, Incremental Learning in SwiftFile, *Proceeding of the International Conference on Machine Learning,* IBM, San Francisco, 2000, pp. 863-870.
30. C. Tasso and M. Armellini, Exploiting User Modeling Techniques in Integrated Information Services: The TECHFINDER System. In E. Lamma and P. Mello (Eds.) *Proceedings of the 6th Congress of the Italian Association for Artificial Intelligence*, Bologna, I, September 14-17, 1999, Pitagora Editrice, Bologna, 2000, pp. 519-522.
31. K. Van Rijsbergen, *Information Retrieval*, 2nd ed. Butterworths, London, UK, 1979. http://www.dcs.gla.ac.uk/Keith/pdf.
32. G. Venolia, L. Dabbish, J.J. Cadiz, A. Gupta, Supporting Email Workflow. Microsoft Research Tech Report MSR-TR-2001-88
33. S. Whittaker and C. Sidner, Email Overload: Exploring Personal Information Management of Email. *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI 1996)*, pp. 276-283.
34. Y. Yang and X. Liu, A re-examination of text categorization methods, *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Berkley, CA, USA, August 15-19, pp. 42-49, 1999.

Mobile Access to the Físchlár-News Archive

Cathal Gurrin, Alan F. Smeaton, Hyowon Lee, Kieran McDonald, Noel Murphy, Noel O'Connor, and Sean Marlow

Centre for Digital Video Processing
Dublin City University
Ireland
{cathal.gurrin}@computing.dcu.ie

Abstract. In this paper, we describe how we support mobile access to Físchlár-News, a large-scale library of digitised news content, which supports browsing and content-based retrieval of news stories. We discuss both the desktop and mobile interfaces to Físchlár-News and contrast how the mobile interface implements a different interaction paradigm from the desktop interface, which is based on constraints of designing systems for mobile interfaces. Finally we describe the technique for automatic news story segmentation developed for Físchlár-News and we chart our progress to date in developing the system.

1 Introduction

The growth in volume of multimedia information, the ease with which it can be produced and distributed and the range of applications which are now using multimedia information is creating a demand for content-based access to this information. At the same time, digitised video content is becoming commonplace through the development of DVD movies, broadcast digital TV, and video on personal computers for both entertainment and educational applications. Besides the growth in volume of multimedia content, we can also observe an increasing and complex range of user scenarios where we require content-based access to such information. Users require access when in a desktop environment, but also, we believe, when using wireless devices in a mobile scenario, each of which will require different access methodologies to be employed. In this paper we discuss mobile access to a video archive of digitised news programs, which can be accessed using desktop devices, PDAs operating on a wireless LAN or XDAs on a GPRS[1] mobile phone network. In this way, and through these different access devices, we support mobile access to a digital video library of broadcast news. Our belief being that mobile users have a demand for wireless access to news content.

[1] GPRS is a packet switching technology for GSM mobile phone networks. A GPRS connection is 'always on' and a single user connection allows 21.4Kbps, but combining connections (time slots) can reach a theoretical speed of 171.2Kbps. However there are a limited number of time slots on a GPRS network.

F. Crestani et al. (Eds.): Mobile and Ubiquitous Info. Access Ws 2003, LNCS 2954, pp. 124–142, 2004.

In addition to simply providing access to digital video archives across a wireless network, we are also working on new methodologies for presenting information to mobile users. In this paper we report on our work on developing an information retrieval system (which supports mobile access) for one type of multimedia information (digital video), of one type of video genre (broadcast TV news) and targeted at one type of user information need, namely a user of Físchlár-News who is not necessarily interested in viewing all the news, but wishes to be kept up-to-date with developing news stories of interest without being restricted to always using a desktop device (i.e. mobile access).

Built on a currently existing system [1], but incorporating mobile access to daily news video, Físchlár-News is based on two new and key underlying technologies:

- Automatic news story segmentation, and;
- Personalisation by means of news story recommendations tailored to user interests of individual users.

In this paper we describe mobile access to the Físchlár-News system and how the fully-automated version of the system operates. We begin in section 2 by describing the desktop version of Físchlár-News (incorporating news story segmentation) which is built upon a news retrieval system that has been operational for last 2 years within the university campus. Section 3 introduces mobile access to Físchlár-News, and discusses the different interaction paradigm that is required for mobile access when compared to Físchlár-News on the desktop. We also discuss how personalised presentation of news stories is being incorporated into the Físchlár-News system to support mobile access. In section 4 we describe how Físchlár-News actually works, and we discuss automatic story segmentation and how recommendation and personalisation is achieved. Finally in section 5, we discuss our progress to date with the development of the system, describe a transitional system that we used during development and finally, we outline our future plans for Físchlár-News.

2 Físchlár-News Video Archive

The Físchlár-News Video archive is one of the results of research in analysis, browsing and searching of digital video content carried out at the Centre for Digital Video Processing in Dublin City University. It is one of four versions of a digital video archive system that we maintain within the centre. Físchlár (all four versions) is an MPEG-7 based digital video content management and retrieval system which supports digital video browsing, searching and on-demand playback using both fixed and mobile devices. The four versions of Físchlár are Físchlár-TV, Físchlár-News, Físchlár-TREC (2002 & 2003) and Físchlár-Nursing. At the time of writing, Físchlár have over 2,500 registered users, of whom about half are “active” and regular users.

The Físchlár-TV system has been in operation on the university campus for over three years and can be accessed via a web browser on a desktop computer. Perceived as a web-based video recorder, registered users have been using the system to record and watch the TV programmes from both the university campus residences and from

computer labs [1]. The Físchlár-Nursing system provides access to a closed set of thirty-five educational video programmes on nursing, and is used by staff and students of the university's nursing school, while the Físchlár-TREC systems were developed for our participation in the interactive search task of the annual activity at the TREC Video Track, in both 2002 and 2003[2].

Físchlár-News, the focus of this paper, automatically records the thirty minute, 9pm, main evening news programme every day from the Irish national broadcaster RTÉ1 and thus has only TV news programmes in its collection. With its web-based interface, the system is accessible with any conventional web browser and now is also accessible from mobile devices. Currently several months of recorded daily RTÉ1 news is online within the Físchlár-News archive (with a total of two year's news archived). This archive is made available to university staff and students, and is also conveniently accessible from any computer lab, library or residence from within the campus. We have chosen the Físchlár-News application as our test-bed for providing mobile access to our Físchlár systems.

In order to facilitate accessing Físchlár-News from a number of different devices (both desktop and mobile based), the entire Físchlár system is based on XML technologies, which by incorporating XSL transformations for each new device required, can easily be extended to incorporate new access methodologies, devices and standards. Fig. 1 shows the basic architecture of the Físchlár-News system which illustrates both desktop and mobile access and the process by which automatic news story segmentation takes place.

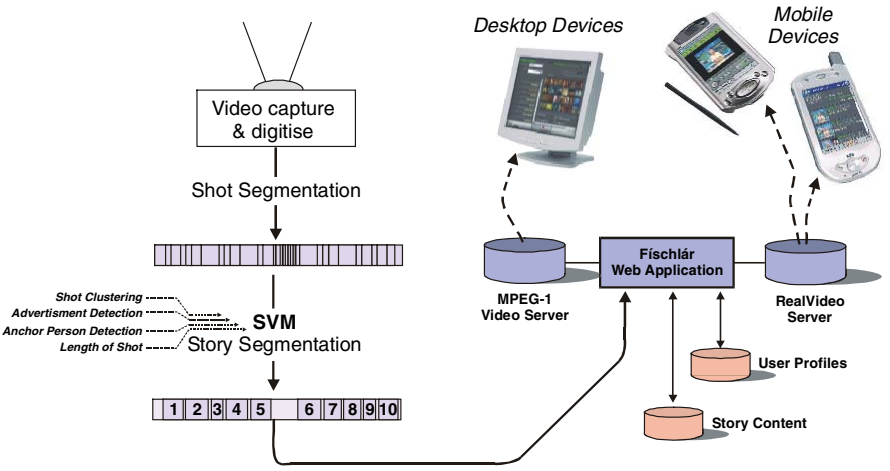


Fig. 1. Architecture of Físchlár-News

In Físchlár-News, mobile access to the news archive is supported for both PDAs (Compaq iPAQ on a wireless LAN) and XDAs, each of which plays RealVideo encoded content, which has been encoded at 20Kbps in order to support streaming across a mobile phone network to an XDA. In a desktop environment, a user can use a conventional web browser (using MPEG-1 video streaming) as shown in Fig. 1. The in-

clusion of XDA support (using a GPRS connection) allows us to prototype a version of our Físchlár-News system in a truly mobile environment, where access is dependent only on the availability of a GPRS connection.

In realising such mobile device interaction for Físchlár-News, two essential technologies are required, namely the segmentation of news programs into a collection of news stories and a facility to automatically recommend these news stories to individual users based on their preferences. We will discuss these aspects of the system in later sections of this paper. Previous versions of Físchlár News have focussed on providing browsing and search support at the shot level by automatically segmenting captured video content into its constituent shots and presenting video to the user as a collection of these shots. However, our current system (discussed in this paper) incorporates search and retrieval of content at the news story level which we feel is more intuitive to a user than at the shot level because a news story is a self-contained and logical unit of data and is more likely to be of benefit to a user than a full news program or a single camera shot from a news program.

2.1 Content Access to the Físchlár-News Archive

When using Físchlár-News on a desktop device, there are a number of ways of accessing news stories, described in the next sections.

2.1.1 Browsing News by News Program

This is the basic level of access in Físchlár-News and is shown in Fig. 2. As can be seen, a listing of news programs grouped by month is displayed on screen.

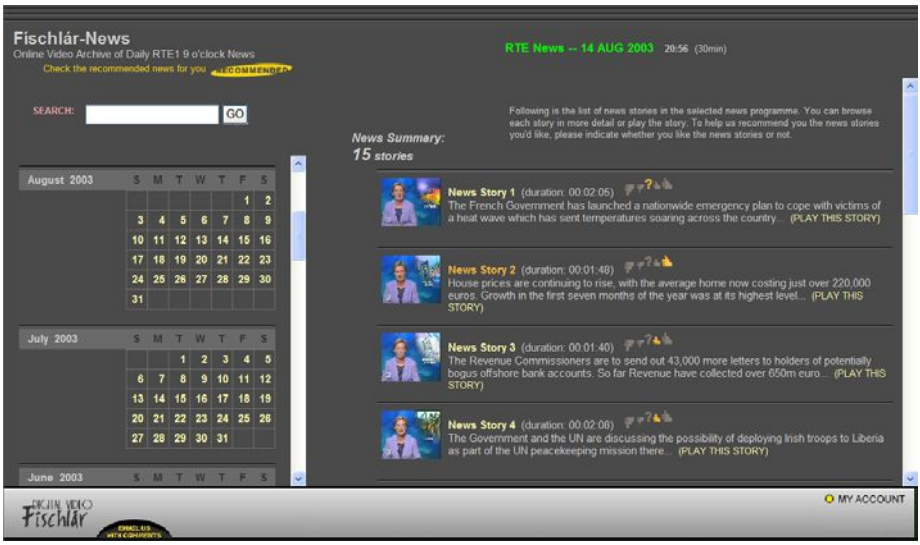


Fig. 2. Físchlár-News (with stories from one program)

Currently this list extends to include news content from April 2003. Selecting any news program will display a list of the news stories from that program (Fig. 2). Each news story is represented by a keyframe (chosen so as to contain the anchorperson and if possible an image in the background associated with the story) and a textual description of the story.

When presented with a listing of news stories there are two options available to the user, the first of these being to playback the news story by clicking on the "PLAY THIS STORY" link which will commence playback (in a new window) from this point onwards (Fig. 3).

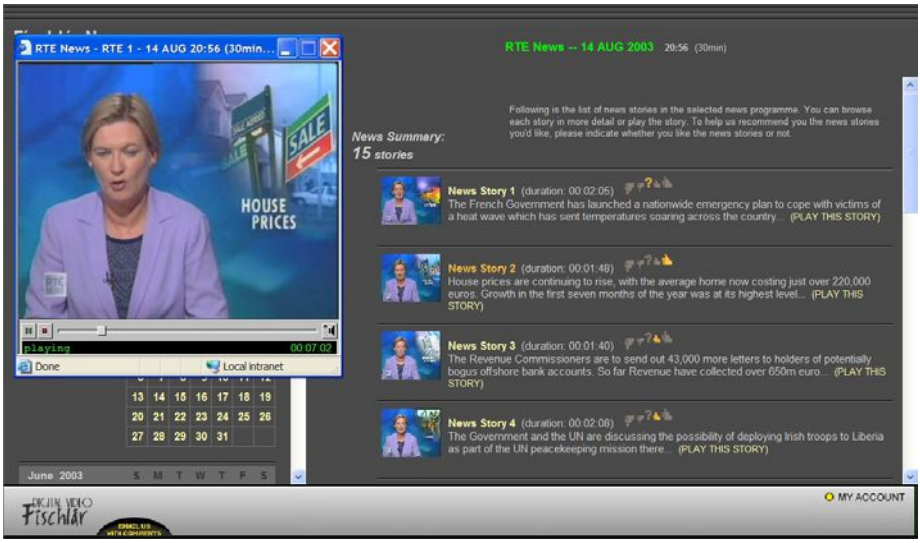


Fig. 3. Playing back a news story

Alternatively, when presented with a listing of news stories the user may examine the news story at the shot level by clicking on either the keyframe or the numbered news story title. If this option is taken the user is presented with a detailed listing of all the camera shots, which have been automatically extracted from that story, as well as the closed caption[2] text that is associated with that story, as shown in Fig. 4. In this way the user can browse through the content of a given story. Clicking on any of the keyframes will commence playback from that point.

However, given that the Físchlár-News archive extends to include several months of news programs, with an additional two years archived, and is growing daily, supporting user navigation throughout this archive of many thousands of stories is essential. The desktop version of Físchlár-News supports a user searching through news stories based on textual content and browsing through the news story archive by fol-

[2] Closed caption text (or teletext) is a textual description of the spoken content of a programme that accompanies certain programmes when broadcast. Most programs now transmitted on TV now have associated closed-caption text.

lowing automatically generated links between news stories. We discuss both search and linkage now.

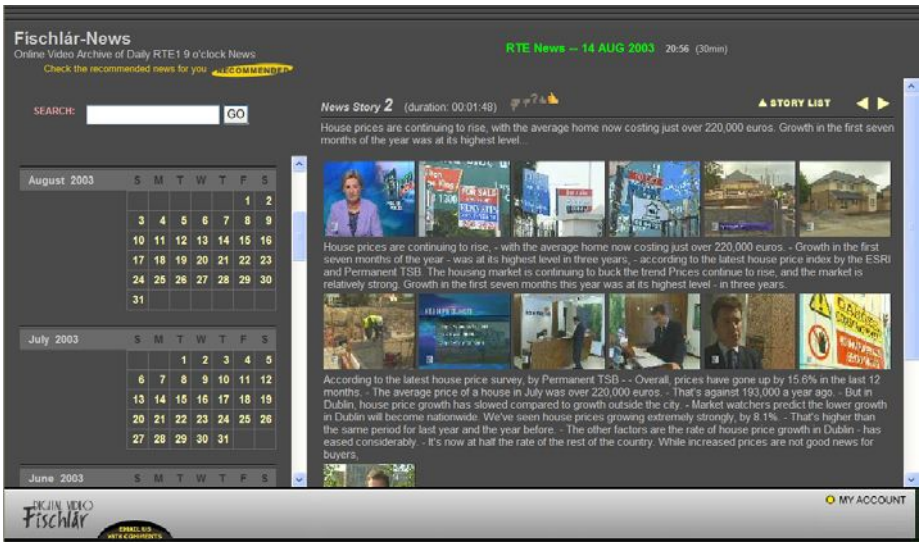


Fig. 4. Shot-level browsing of a news story

2.1.2 Content Searching for News Stories

Given that there are a large number of stories in the Físchlár-News system, one of our support measures is content-based search and retrieval of news stories. This is achieved by representing each news story by a textual description, which has been automatically extracted from the closed caption text and subsequently supporting user queries over the story archive. This facilitates content-based retrieval of news stories based on textual queries. For example, in Fig. 5, a query "house prices" has been presented to the Físchlár-News system.

The results of the search (166 news stories) are presented on the right side of the screen in decreasing order of relevance. Once again, clicking on 'PLAY THIS STORY' commences playback and clicking on the story title or keyframe takes the user to a shot listing. However, when a list of relevant news stories is presented to the user, another option exists which allows a user to view the story in the context of that day's news by following the date link which displays a listing of news stories from the news program recorded on that date.

The third access methodology employed in Físchlár-News is in following automatically generated links between related stories.

2.1.3 Following Automatically Generated Links between News Stories

Using the closed caption transcripts for a given news story it is possible not only to provide for text based search and retrieval of news content at the story level (as just described), but also to identify similar stories to any one given story, and to provide

the facility for content-based linkage of news stories using only the closed caption transcripts. Therefore, on a desktop device, for any story that a user is currently accessing, Físchlár-News automatically generates a ranked list of story-links to the ten most similar news stories, which we refer to as 'Related Stories' (see Fig. 6 which shows a listing of related stories to a story about house prices).

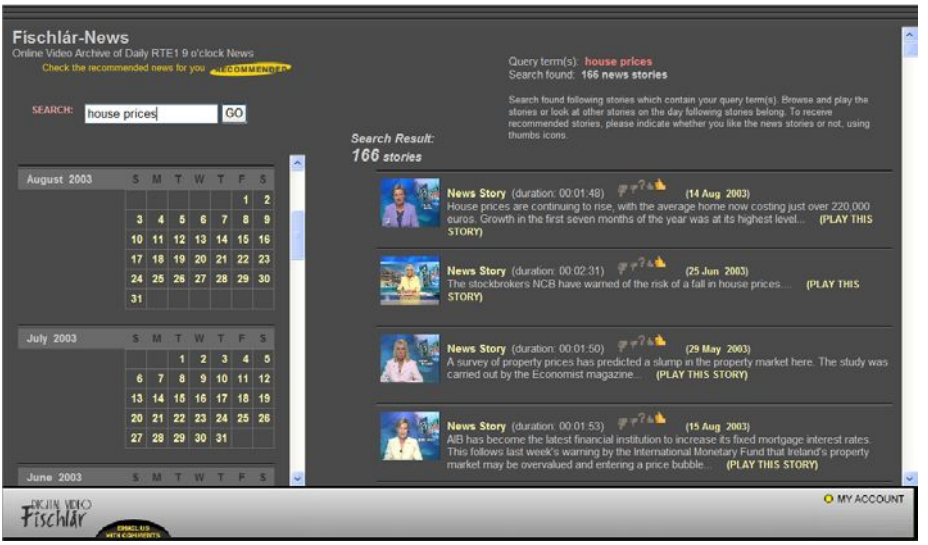


Fig. 5. Content searching the news archive

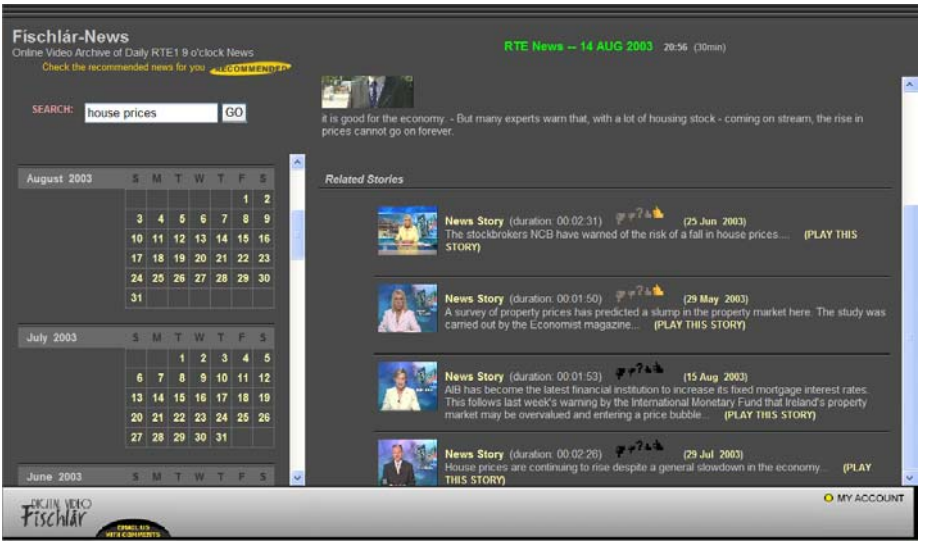


Fig. 6. Illustrating related stories

2.2 Gathering User Feedback

In order to provide personalisation and recommendation to users accessing the system using mobile devices, we gather user feedback and preferences when the user accessed Físchlár-News from the desktop environment. At any point while browsing either the archive or a particular news story (on a desktop device), the user is presented with the opportunity to rate a particular story on a five point scale from “do not like” (thumbs down) to “like very much” (thumbs up). This can be seen in Fig. 7 where a user has rated a news story as being one that she likes very much (a large thumbs up). In this way we explicitly capture a user’s preferences for news topics that they are interested in. This will allow us to match individual users together based on complementary preferences and to recommend news stories for a user based on this collaboration graph.

Fig. 7. The five-point story rating scale

In addition to this process of explicitly gathering data from a user, usage data is automatically gathered (on the desktop device) as the user plays back news stories or browses news stories. This information is then used (along with the explicitly gathered data) for recommending news stories to users. So, for example, if a particular user liked news stories on a given topic, and watched these stories, then additional news stories could be recommended to the user based on the viewing habits, or user ratings, of similar users. These recommendations are used as one of the two primary access mechanisms for the mobile version of the Físchlár-News system.

3 Mobile Access to Físchlár-News

Small display size, awkward methods of data input and distractive environments have been noted as major constraints in designing systems for mobile platforms [3, 4, 5]. For example, a typical mobile device, the Compaq iPAQ has a 3.8" TFT screen which operates at a resolution of 240 x 320 (portrait orientation) in 16-bit colour. Compare this to a conventional desktop device, besides having larger storage and memory, faster processors, the supported resolution on any such device (in recent years) is at least 1024 x 768 (800 x 600 as a standard safe-resolution for design), with 24-bit colour and a 15” diagonal display with a landscape orientation.

In order to stream video to such mobile devices taking into account resolution issues and bandwidth (we accommodate GPRS 21.4Kbps as a minimum), the entire video must be downsized from the MPEG-1 (352 x 288) resolution at 25fps used for Físchlár-News on the desktop to RealVideo format (156 x 128) at 30 fps. This equates to 13.5Kbps for the video and 6.5Kbps for the audio data. MPEG-1 streaming for the desktop requires about 1Mbps.

Consequently, there have been suggestions on devising different interaction paradigms suitable for the mobile environment rather than simply following the conventional direct manipulation interfaces successfully used in desktop platforms [6], [7], [8]. More and more qualitative studies are appearing which help us better understand how people use and interact with mobile devices, and the kinds of context they experience when doing so [9], [10], [11]. The general consensus is that a mobile interface should require a different interaction style from that of the GUI desktop interface, and that attempts to replicate all the functionality of desktop system into a mobile device are a mistake [12], [7], [6], [3].

Though the current literature alerts to the fact that we do not have any established or known methodology on which to base an interface design for a mobile platform, a number of rough design guidelines have been suggested based on experiences of individual researchers. These include the following:

- minimise user input where applicable, provide simple user selections such as yes/no options, simple hyperlinking by tapping, etc. instead of asking the user to articulate query formulation or use visually demanding browsing that requires careful inspection of the screen,
- filter out information so that only a small amount of the most important information can be quickly and readily accessed via the mobile device (e.g. use of automatic recommendation as provided in the Físchlár TV system [21]),
- Proactively search and collect potentially useful pieces of information for a user and point these out, rather than trying to provide full coverage of all information via an elaborate searching/browsing interface.

In terms of developing any system for a mobile device which is to support searching and information retrieval tasks, all these guidelines point to more pre-processing on the system's side in order to determine what information a particular user will most likely want to see. This encourages the development of systems that proactively recommend a particular piece of information (or pointers) to the user, and consequently demand less interaction on the user's part. This aspect is even more important in the case of information retrieval from a video archive where browsing is such an important component of video access. What all this means is that in the development of search systems to be accessed from mobile platforms, the information retrieval functionality should be hidden as much from the user as is possible, and should form part of the data pre-processing. In supporting mobile access to the Físchlár-News archive, our approach has been in line with these guidelines by incorporating the personalised list of news stories as the primary access point for mobile users and providing a personalised window on these news stories based on each user's individual preferences. Secondary access points include archive browsing.

Fig. 8 illustrates the logical breakdown of news programs into stories and associated shots based on keyframes. It is our belief that story based presentation can be supported using both mobile and desktop devices, however, if finer granularity of retrieval is required (shot level browsing with stories) then desktop devices are essential due to interaction design methodologies for mobile devices [13] as well as the bandwidth limited nature of some such devices, e.g. the XDA we use to prototype our mobile access.

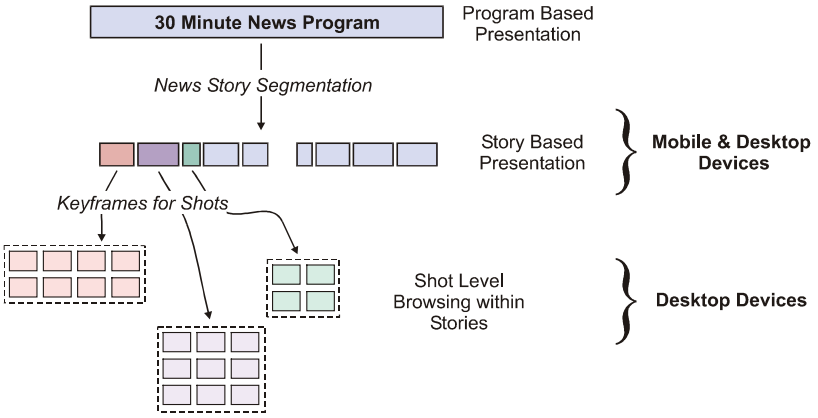


Fig. 8. A logical breakdown of news programs

Given that user interaction with a mobile device should be limited to a subset of the functionality of the desktop version for reasons outlined in the previous section, the functionality of the mobile device is to support two methods of using Físchlár-News:

- providing personalised access to the news archive by presenting the user with a listing of news stories of interest to the user (Section 3.1), or;
- supporting the user access news stories in the archive by browsing the reverse chronologically ordered listing of news programmes (i.e. Programme Browsing in section 3.2).

3.1 Personalised Presentation of News Stories

The primary access mechanism for the mobile device is based on personalisation of news stories tailored to individual user preferences. Each user's personalised view of the news archive is based on similarity of program content to previously rated programs and also to the concept of collaborative filtering. Collaborative filtering, in what is perhaps its most famous form, is employed by Amazon.com when making user recommendations based on a users previous purchases or recently viewed items. In the case of Físchlár-News collaborative filtering is employed based primarily on previously gathered user ratings of any given news stories as well as news story usage histories. We will outline our collaborative filtering mechanism in greater detail in section 4.

Upon accessing Físchlár-News using a mobile device, a user has the option of being presented with a personalised listing of recent news stories (see Fig. 9), that it is hoped will be of interest to that user, based on program content and the output of the collaborative filtering process. Each story in this list will be represented by a short description (similar to the desktop device), generated from the closed caption text, and a keyframe. The only user input that is required from a user's perspective is to select a news story to playback (Fig. 10), which causes the story to be streamed in RealVideo format.

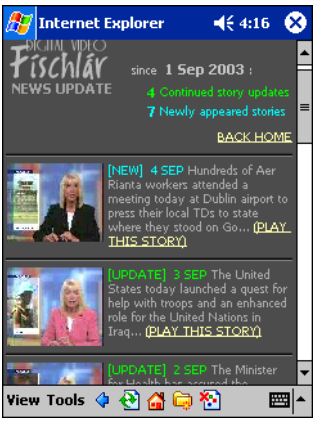


Fig. 9. Personalised story recommendations

Fig. 10. Playback on a mobile device

By incorporating this personalisation aspect of Físchlár-News on a mobile device we are minimising user input by filtering out content that the user may not be interested in, where this filtering is based on news story rating data and content similarity from the desktop device. For more complete rationale on the interaction design approach taken and the detailed consideration for this particular interface for a PDA, see [13].

3.2 Programme Browsing

An alternative to personalised news story presentation is provided, to enable a user to access news programmes regardless of their presence or absence in the personalised list. In this way, a user is not limited to only viewing stories that the system chooses, but is presented with a reverse chronological listing of recorded news programmes (not unlike the desktop interface in Section 2) so that the user may browse the entire news story archive (Fig. 11). Upon selecting a news programme, the user is presented with a listing of news stories from within that programme (Fig. 12), in a similar manner to the listing of personalised stories, with each news story represented by the keyframe and the textual description of the story.

4 Físchlár-News, How It Works

In realising such mobile device interaction for Físchlár-News, two essential technologies are required, namely the segmentation of news programs into a group of news stories and a facility to automatically recommend these news stories to individual users based on their previously gathered preferences and the preferences of others. In the following sections we describe automatic story-based news video retrieval and the mechanisms we employ for automatic recommendation.

Fig. 11. Reverse chronological daily news listing on a mobile device

Fig. 12. Story listing on a specific date

4.1 Automatic Story Segmentation

As we have stated, Físchlár-News operates over news stories as the primary unit of retrieval, which is especially important in the mobile environment, but this requires a method of segmenting an entire news programme into a listing of its constituent news stories. If done manually this is a time-consuming task and if done automatically, which is essential for any large-scale archive of story-based news content (such as Físchlár-News) is an extremely difficult task. However, given that news programs from one broadcaster (RTE in our case) represents a very constrained domain this makes automatic story segmentation somewhat easier to accomplish. For example, there are a lot of features (sources of evidence) that can be extracted automatically from the video stream to aid the segmentation process, if it is known in advance what to look for (i.e. in a constrained domain). We have tested and integrated into Físchlár-News an automatic news story segmentation system that is based on a combination of a number of different sources of evidence automatically extracted from the digitised news video.

There has been previous research in this area (Table 1), upon which we now report.

Table 1. Comparison of Approaches to Automatic Segmentation of News Stories

Evidence System	Visual	Audio	Closed Caption	Combination Method
Informedia [14]	Shot boundary detection; face detection; OCR; black frame	Speech recognition; silence detection; acoustic environment change; signal-to-noise ratio	“>>>”, “>>”, absence of text	Step-by-step (ad-hoc)

Table 1. (continued)

Evidence System	Visual	Audio	Closed Caption	Combination Method
VISION [15]	Shot boundary detection	Audio energy-based shot merging	">>>", ">>", absence of text, word identification for topic distance calculation	Step-by-step (shot boundary detection followed by audio-based merging followed by closed caption-based adjustment)
BNE & BNN [16]	Black frame; logo detection; anchor booth & reporter scene detection	Silence	Named entity heuristics in captions, ">>>", ">>"	Finite State Automation enhanced with time transitions
Topic Browser [17]	No	No	Morphological analysis	(only Closed Caption used)
ANSES [18]	Shot boundary detection	No	Lexical chaining	Step-by-step (shot boundary detection followed by Lexical chaining-based merging)
Físchlár-NEWS	Shot boundary detection; face detection	No	No	Support Vector Machine

4.1.1 Previous Research

Many of the current studies in news story segmentation make use of multiple evidences for segmentation from visual content, audio content and closed caption text associated with a particular news programme. Visual evidences currently studied and used include shot boundaries (helping to identify possible story boundaries), blank frames (indicating story boundaries), anchorperson (indicating start/end of stories). Audio evidences studied include existence of speech/music, silence (indicating story boundaries) and audio energy level. Use of closed caption text, often used as the primary source of evidence, has been more extensively studied with linguistic analysis to detect news story boundaries. Evidences in closed caption text includes simple clues such as complete absence of the closed caption (an indication of a commercial break), welcome phrases such as "hello and welcome" (indicating the start of the news), "back to you in <location>" (indicating reporter to anchorperson), etc. and manual marking[3] ">>" (indicating speaker change) or ">>>" (indicating story change), as well as sophisticated topic change detection by lexical chaining analysis. Using only closed

[3] Unfortunately not all closed caption broadcasts contain such manual markings, RTÉ1 news is one such example. Even if such manual markings were available, most closed caption text is not perfectly aligned with the video content and must be realigned in order to produce accurate story segmentation results.

caption analysis for news story segmentation also gives acceptable results [18]. Combining individual evidences into more reliable story segmentation is conducted in different ways, but most often follows sequential processing in which visual analysis (shot boundary detection) followed by audio analysis (merging back related shots) as done in [14, 15, 17], or the use of a state transition map to classify different states of scene changes in news programmes [16]. Table 1 (above) shows a summary of analysis methods and combination methods used in six news video retrieval systems, including Físchlár-News. In the Físchlár-News system, we use an SVM (Support Vector Machine) to combine various evidences automatically extracted from the video content.

4.1.2 Automatic Story Segmentation in Físchlár-News

For automatic news story segmentation, we analyse various visual features in the news programmes to automatically determine story boundaries. This involves the utilisation of algorithms for anchorperson detection using shot clustering [19], which detects when an anchor person is on screen, as well as advertisement detection [20], which determines when advertisements occur and face detection which detects human faces in the video content [21]. In addition we are considering the use of speech/music discrimination [22, 23].

All of the analysis techniques mentioned above for automatic story segmentation take place at the shot level (recall Fig. 1) and have been combined to create an automatic story segmentation system. The output from the advertisement break detection algorithm is used to pre-process the shots, discarding as candidates for story boundaries any shots which are part of an advertisement break.

The combination of the other analysis outputs is being supported through the use of Support Vector Machines [24] and initial results suggest that this technique can effectively and efficiently combine these diverse analyses. Each shot that comprises a news programme is described by a feature vector made up of the outputs from the various analysis tools, and the Support Vector Machine is trained to classify shots into those which signal the start of a new story and those which do not, hence we are then able to detect story boundaries in a TV news programme.

In order for an SVM to operate, it must undergo a training process. This we have done using a training set consisting of 435 example shots, 86 of which are positive examples of news story boundaries and 349 of these are negative examples. Following from this we tested the performance of our SVM, with very promising results, on a small test set of six news programmes with precision and recall figures of 1.0 and .859 respectively. We appreciate that this test set is small and we are currently testing the SVM for story bound segmentation on a larger test set of news programs as part of the TREC Video Track 2003, which will give us a better indication of SVM performance. The automatic segmentation system is operational since October 2003, when it replaced a temporary system which utilised manual story segmentation.

For each automatically segmented news story, a textual description will be extracted from the closed-caption text as well as a keyframe automatically extracted for each story. Our belief is that the (single) keyframe chosen to represent each news story should (where available) contain the anchorperson as well as a background image,

which represents the story. In order to automatically achieve this we will incorporate both temporal and anchorperson detection knowledge.

4.2 Físchlár-News Story Recommendation and Personalisation

Given that we have developed the Físchlár-News system with a mobile user in mind, the most important news stories that a mobile user requires should be presented to the user with the minimal user intervention or required data input. In order to facilitate this we have put great emphasis on supporting news story recommendation and personalisation. In a desktop environment Físchlár-News supports these features along with story-based retrieval using textual queries and story linkage. However, in a mobile environment, personalisation and recommendation is a central aspect of user interaction with Físchlár-News, which helps to address some of the major constraints in designing systems for mobile platforms [3, 4, 5]. One highly important aspect of this personalisation and recommendation is Collaborative Filtering.

4.2.1 Collaborative Filtering

In another application, Físchlár-TV, we have been using the ClixSmart engine [25] to provide collaborative filtering based recommendations of TV programs for recording and for playback from those recorded and available in the Físchlár-TV library. The ClixSmart engine is a collaborative filtering system that recommends items based on the actions of equivalent users. For additional information on how collaborative filtering works within the Físchlár system in general see [26].

In Físchlár-News, personalisation is employed based on a combination of content similarity of news stories and collaborative filtering. As stated, Físchlár-News on a mobile device will filter out news stories that will not be of interest to the user based on past history, in addition to supporting temporal based browsing of stories from within the news archive. In order for collaborative filtering aspect of personalisation to be effective, all required data must be gathered by the system from the desktop interface. The data gathered is as follows:

- explicit user ratings as described previously in section 2.2,
- usage data on a per-user basis from story playback logs,
- usage data on a per-user basis from story access logs

This data is automatically gathered while a user uses the desktop interface and is used to populate a story-by-user matrix, which is used in the collaborative filtering process. Therefore, we can see that the mobile interface is supported and works in parallel with the desktop interface, by the desktop interface collecting and processing user data to support the personalisation process in the mobile environment.

5 Conclusion

In this paper we have described our efforts at supporting mobile access to a large-scale library of digital video news content. Our efforts have focused on incorporating story segmentation and personalisation into the Físchlár News system in order to support

access using mobile devices. This mobile access is made possible by careful development of the Físchlár-News system, its interface and browsing and retrieval methodologies to support the bandwidth limited, screen size limited, mobile user using mobile devices, such as XDAs on a GPRS mobile network.

These mobile devices have a number of key features which limit how we interact with them, These include small display size, awkward methods of data input and in some cases (such as the XDA) limited bandwidth. Conventional wisdom suggests that different interaction paradigms should be devised for the mobile environment rather than simply following the conventional direct manipulation interfaces successfully used in desktop platforms. Mobile access should require, minimal user input and filtered information presentation based on background data collection so that only a small amount of the most important information can be quickly and readily accessed via the mobile device. Table 2 shows a summary of the interaction mechanisms on the mobile and desktop devices for Físchlár-News.

Table 2. Summary of Desktop v.s. Mobile Device Interaction

	Desktop Device	Mobile Device
Programme Browsing	Y	Y
Story Browsing	Y	Y
Shot Browsing	Y	N
Text Querying	Y	N
Related Stories	Y	N
Personalisation	N	Y

In Físchlár-News, the mobile interface is supported and works in parallel with the desktop interface, in that the background data collection to support personalisation and recommendation is mined from observing user activities in a desktop environment and used to support personalisation in the mobile environment.

5.1 Our Progress to Date

In realising the underlying technology required for story-based and recommendation-based mobile access to the news story archive, we built and for a number of months, used a manually segmented version of Físchlár-News to kick-start the fully-automated system that is in operation since October 2003. From April 2003 until October 2003, recorded daily news programmes were manually segmented into stories (in XML format) to generate an initial library of news stories. The manually marked XML files are uploaded into the system, which then incorporates them into the archive. Manual segmentation is a time consuming process in which each news story identified from a given programme is represented in XML format by the following information; a start-time and end-time, a representative keyframe and representative text to describe the story, which had been extracted from the closed captions.

This initial manual segmentation just described served to start collection of initial user ratings of news stories required for collaborative filtering while the automatic segmentation mechanism was being prototyped.

Collaborative filtering is only of benefit if users of the system access or watch news stories (mined from user logs) and/or rate news stories using the thumbs-up and –down indication (as discussed in section 2.2). In order to collect data to support the collaborative filtering process, a core group of regular users of the Físchlár-News system have been encouraged to rate news stories since the end of April, 2003. To date (mid-October 2003) we have received over 22,000 individual story recommendations from these users. The reason for doing this was that in October 2003, when the fully automated Físchlár-News system went live, it was immediately able to generate recommendations of stories for all users based on collaborative filtering using the judgements of this core group of users.

5.2 Future Plans

Given that the Físchlár-News system outlined in this paper is a live system based on research being carried out within the Centre for Digital Video Processing, it will be subject to modification and improvement. Our future plans include identifying what other functionality (from the desktop) can be included in the mobile version that fits in with the design guidelines for mobile devices.

Currently a daily reminder email is sent out to each user of the Físchlár-News system reminding them that the latest news programme has been processed and available for browsing and searching. This email is currently identical for all users, however, the facility exists for us to tailor or personalise each daily reminder email based on the users previous preferences for news story content.

Finally, it is possible using SVMs to incorporate additional sources of evidence into the automatic segmentation process if this is deemed necessary. The results of our larger test of the performance of the SVM will dictate whether this is required.

Acknowledgements

This material is based upon work supported by the IST programme of the EU in the project IST-2000-32795 SCHEMA. The support of the Informatics Directorate of Enterprise Ireland is gratefully acknowledged. We have benefited considerably from collaboration with Prof. Barry Smyth in UCD on the personalisation and recommender work reported in this paper.

References

1. Smeaton, A.F.: Challenges for Content-Based Navigation of Digital Video in the Físchlár Digital Library. In: Proceedings of CIVR-2002 (London, UK, July 2002). Lecture Notes in Computer Science (LNCS) 2383.
2. Guidelines for the TREC Video Track: http://www-nlpir.nist.gov/projects/t01v/. Last visited October 2003.
3. Longoria, R.: Designing mobile applications: challenges, methodologies, and lessons learned. In: Proceedings of HCI-2001. (New Orleans, Louisiana, 5-10 August 2001).

4. Sacher, H., and Loudon, G.: Uncovering the new wireless interaction paradigm. ACM Interactions Magazine, 9(1), 2002.
5. Pascoe, J., Ryan, N., and Morse, D.: Using while moving: HCI issues in fieldwork environments. ACM Transactions on Computer-Human Interaction (TOCHI). 7(3), 2000.
6. Kristoffersen, S., and Ljungberg, F.: "Making place" to make IT work: empirical explorations of HCI for mobile CSCW. In: Proceedings of ACM SIGGROUP Conference on Supporting Group Work, 1999.
7. Marcus, A., Ferrante, J., Kinnunen, T., Kuutti, K., and Sparre, E.: Baby faces: user-interface design for small displays. In: Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems "Making the Impossible Possible", (Los Angeles, CA, April 18-23, 1998).
8. Rist, T.: A perspective on intelligent information interfaces for mobile users. In: Proceedings of HCI-2001, (New Orleans, Louisiana, 5-10 August 2001).
9. Palen, L., and Salzman, M.: Beyond the handset: designing for wireless communications usability. ACM Transactions on Computer-Human Interaction (TOCHI), 9(2), 2002.
10. Perry, M., O'Hara, K., Sellen, A., Brown, B., and Harper, R.: Dealing with mobility: understanding access anytime, anywhere. ACM Transactions on Computer-Human Interaction (TOCHI), 8(4), 2001.
11. Jordan, P., Peacock, L., Chmielewski, D., and Jenson, S.: Disorganization and how to support it - reflections on the design of wireless information devices. In: Proceedings of IHM-HCI 2001, (Lille, France, September 10, 2001).
12. Thomas, P., Meech, J., and Macredie, R.: A Framework for the Development of Information Appliances. In: Proceedings of ACM Symposium on Applied Computing, 1995.
13. Lee, H., and Smeaton, A.F.: Searching the Físchlár-News Archive on a Mobile Device. In: Proceedings of ACM SIGIR 2002, Workshop on Mobile Personal Information Retrieval (Tampere, Finland, August 2002).
14. Hauptmann, A., and Witbrock, M.: Story Segmentation and Detection of Commercials in Broadcast News Video. Advances in Digital Libraries Conference, (Santa Barbara, CA, 22-24 April, 1998).
15. Gauch, J., Gauch, S., Bouix, S., and Zhu, X.: Real time video scene detection and classification. Information Processing and Management, 35(5), 1999.
16. Merlino, A., Morey, D., and Maybury, M.: Broadcast News Navigation Using Story Segmentation. In Proceedings of ACM Multimedia 1997 (Seattle, WA, November 1997).
17. Ide, I., Mo, H., Katayama, N., and Satoh, S.: Topic-based structuring of a very large-scale news video corpus. AAAI Spring Symposium on Intelligent Multimedia Knowledge Management, (Stanford University, 24-26 March, 2003).
18. Pickering, M., Wong, L., and Ruger, S.: ANSES: summarisation of news video. In: Proceedings of CIVR-2003, (University of Illinois, IL, USA, July 24-25, 2003).
19. O'Connor, N., Czirjek, C., Deasy, S., Marlow, S., Murphy, N. and Smeaton, A.F.: 2001. News Story Segmentation in the Físchlár Video Indexing System. In: Proceedings of ICIP 2001, (Thessaloniki, Greece, 7-10 October 2001).
20. Sadlier, D., Marlow, S., O'Connor, N., and Murphy, N.: Automatic TV Advertisement Detection from MPEG Bitstream. Journal of the Pattern Recognition Society, 35(12), 2002.
21. Czirjek, C., O'Connor, N., Marlow, S. and Murphy, N.: Face Detection and Clustering for Video Indexing Applications. In: Proceedings of ACVIS 2003 (Ghent, Belgium, 2-5 September 2003).
22. Jarina, R., Murphy, N., O'Connor, N. and Marlow, S.: Speech-Music Discrimination from MPEG-1 Bitstream. In: Advances in Signal Processing, Robotics and Communications, WSES Press, 2001, 174-178.

23. Jarina, R., O'Connor, N., Marlow, S. and Murphy, N.: Rhythm Detection for Speech-Music Discrimination in MPEG Compressed Domain. In: Proceedings of DSP 2002, (Santorini, Greece, 1-3 July 2002).
24. Burges C.: A Tutorial on Support Vector Machines for Pattern Recognition. Data Mining and Knowledge Discovery, 2(2), 1998, 121-167.
25. Smyth, B., and Cotter, P.: A Personalized Television Listings Service. Communications of the ACM, 43(8), 2000.
26. Wilson, D., Smyth, B., and O'Sullivan, D.: Improving Collaborative Personalized TV Services - The Study of Implicit and Explicit User Profiling. In: Proceedings of the 22nd SGAI International Conference on Knowledge Based Systems and Applied Artificial Intelligence. (Cambridge, UK, December 10-12, 2002).

A PDA-Based System for Recognizing Buildings from User-Supplied Images

Wanji Mai[1], Gordon Dodds[1], and Chris Tweed[2]

[1] Virtual Engineering Centre, Queen's University Belfast, Cloreen Park, Malone Road, Belfast, BT9 5HN, Northern Ireland, UK
{w.mai,g.dodds}@ee.qub.ac.uk
[2] School of Architecture Queen's University Belfast, 2 Lennoxvale, Belfast BT9 5BY
c.tweed@qub.ac.uk

Abstract. This paper reports on research into and development of portable hardware that will enable users in the field to send images, and associated positional data from a PDA to a server for processing. The central aim is to provide navigational and informational services to an urban mobile user based on building recognition. The paper begins by describing the hardware before presenting research into server-side building recognition methods that operate by comparing user-supplied images with images generated by an existing 3d virtual model.

1 Introduction

Recent advances in the development of personal digital assistants (PDAs) and wireless communication networks enable a new generation of sophisticated mobile applications. PDAs can now support a range of add-on devices, such as digital cameras, and communicate using a variety of networking protocols, such as GPS, WiFi, and Bluetooth. With the increased availability and advanced features of these low-cost, portable and mobile system devices, there is a potential to develop a wide range of applications [9, 2]. The combination of mobile computational, imaging and positioning capabilities and network access opens the door to a variety of novel applications, such as pedestrian navigation aids, mobile information systems and other applications usually referred to as 'location services' [3]. [3] improves the GPS accuracy using GPS data, orientation data, image, and Hough Transforms. The hardware system in [3] is quite similar with ours except that their system is very heavy and uses careful calibration. However, this research seeks to exploit the capabilities of the Personal Digital Assistance (PDA) and the imaging functions of a PDA camera for building recognition.

There have been several digital city projects [1, 6, 7, 8, 14] concerned with cityscape and city models in the past decade. And most of them are designed to be an integrated information and service environment for everyday life and tourism [1, 6, 7, 8]. Some of them also put much effort into the 3D model for city promotion applications [16], etc. This project also requires a 3D virtual model for reference. However the application of this system is to help tourists identify buildings on the road and also

F. Crestani et al. (Eds.): Mobile and Ubiquitous Info. Access Ws 2003, LNCS 2954, pp. 143–157, 2004.

provide immediate information in real-time. Image processing for object recognition is one of the main aspects of this project.

Visitors to a city sometimes find problems in understanding maps or guidebooks, even guidebooks with symbols. In another project, surveys of pedestrians found that a significant proportion (12% of males, 24% of females) had difficulty in locating themselves on a printed map [12]. However, the system described here helps visitors to identify their locations and get information about urban objects using the object images from a portable commercial PDA.

Unlike kiosks, or other fixed information stands, this system is much more flexible and dynamic. People can stop at any interesting object, take a picture of it and send the image with the corresponding GPS and orientation sensor data from the devices equipped with the PDA system to the web server. The server can identify the object using an online images generated from a 3d city model. If the building is found to be the same as the one in the city model, the server can provide the user with relevant information. This system also allows PDA users to save their pictures in a public database on the server temporarily and download them after they have returned home. This solves the problem of the limited size of the memory card in portable devices.

To detect objects reliably, a model of the object is needed. Thus, one of the key components of our approach is a 3d city model. The system described in this paper uses a relatively small 3d model of the square in front of the University as a reference study. The model was constructed from a manual survey and the texture maps were derived from photographs. At the same time, a Geographical Information System (GIS) has been used to provide location information and to enable the transformation of the coordination between the GPS and city model components.

The overall plan for this project is described below. Users are equipped with some hardware devices to obtain different data, e.g. the GPS data, orientation data and the image. The use of images removes some of the problems of low GPS and orientation data accuracy in urban areas. Then users need to send the data to server for further processing. On the server side, this project is designed to do image processing for building recognition. If the image is confirmed to be part of the model, the image has been identified. In this way, users are able to obtain the information on their location and also the objects they are interested in. Position from the GPS receiver will never change if a user reports from the same location. However the building located there may possibly be changed with time. This system not only provides a way for tourists to travel around the city, especially as this system is very handy and low cost, but also a possible way to keep the model updated. A detailed flow chart for the user operation sequence is shown in section 2.4.

Most of the previous research in this area has been concerned with the location-based services. This paper presents a ‘location and image-based service’, which delivers information about a specific building of interest in real-time to a mobile user through the Internet by identifying the building from an image supplied by the user. To realize the image-based service, requires not only the location data (GPS data) of the user but also the direction and tilt data (from the orientation sensor attached on the PDA).

Section 2 describes the system design, including each component of the hardware devices, software application and networks. Methods for object recognition are described in section 3 and some results of this project are presented in section 4. Section 5 summarises the main points of the paper and discusses further research.

2 System Design

The system consists of three main parts: the client side, server side and the connecting networks. Fig. 1 shows the relationships between these different parts.

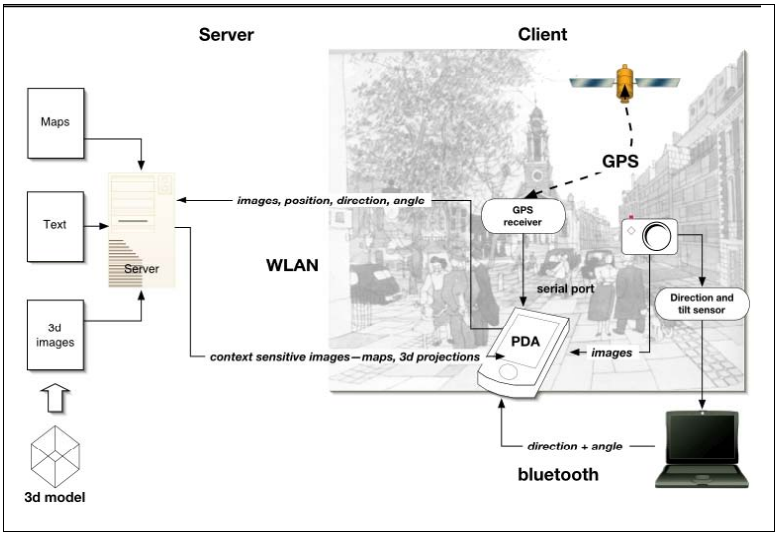


Fig. 1. System components diagram.

2.1 Client Side

The client side is a portable PDA system. The system includes an *iPAQ 3870*, *Nexi-Cam* PDA camera with resolution 600x800, orientation sensor and GPS receiver. Because the PDA development was still at an early stage when this project started, PDA is not able to provide enough interfaces for all the devices we wish to use-for example, the *iPAQ 3870* provides one expansion connector for its expansion pack and one universal connector, which can be converted to a serial port and is integrated with *Bluetooth* and GPRS. But the USB interface sensor is not able to connect to the Pocket PC. So in this project, a laptop is used to receive the data from the USB port and Bluetooth supports the communication between the PDA and laptop. However, this problem should be resolved in the near future by the next generation of PDAs. The GPS receiver and camera are connected to the PDA respectively using the universal connector and expansion connector. A WLAN card is plugged in the camera, which allows the PDA to access Internet through WLAN.

2.2 Server Side

3D City Model

The selected city model is being built from site surveys based on the GIS map, conducted using Cyrax2500 laser scanner and *3DMax* software. Digital images will be used to record details and texture of the buildings. Images from the PDA users will

keep this database up-to-date. A 3d model is constructed from the point clouds provided by the scanner and is imported into *3DMax* for visualization. With this 3d model, it is possible for us to generate images from any position and direction, such as those returned from the PDA client. This model image provides a reference for building recognition. One part of the generation of the city model image will be implemented by *3DMax*. Figure 2 shows the building model of the Lanyon Building of Queen's University Belfast, which has been implemented as a pilot study. On the top right corner is the script window (with words inside). This script sets the camera to the correct position and adjusts the camera angle based on the incoming GPS and orientation data from the user. Camera is the white object in the left top window. Right bottom window is to render the reference image.

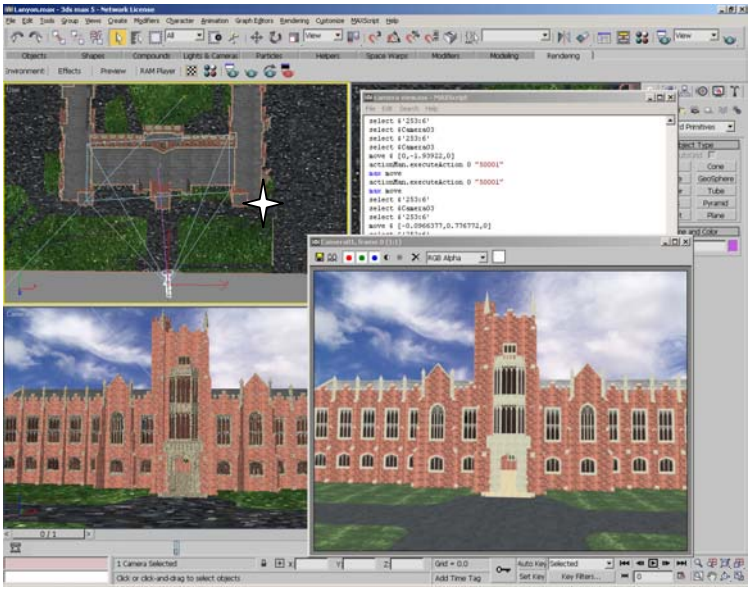


Fig. 2. City model in *3DMax* environment.

GIS System

With the GPS data from the receiver, users are able to identify their locations in the GIS. The GIS is also used to link the building components to a GPS position. For example, in the virtual model, the origin is at the right front corner of the building, the 4-point star in the top left window in Figure 2. The GIS system converts the *3DMax* co-ordination to GPS co-ordination.

Applications

Some applications are available on the server.

The first application is to identify the buildings from user-supplied images and provide information in real-time, e.g. transportation, accommodation, history and events. *Matlab* is used for the image processing and some building recognition methods, like line detection and segmentation are applied.

The second application is the 'public space'. This space is for users to save their travelling pictures temporarily. As the size of memory cards for PDA cameras is limited, it will be helpful if the server can provide this public space for users, who can later download the pictures to a desktop computer. Normal security will be provided.

The final application is the position display. With the GPS data from the PDA user, server is able to display his location in a 2D GIS map in real-time.

2.3 Networking

Bluetooth

A Bluetooth SDK from WIDCOMM has been used to develop Bluetooth applications for communication between the PDA and laptop. This program provides three functions. The first is to synchronize the time between PDA and laptop. The second is to transfer the GPS data from the PDA to the laptop. And the last one is to instruct the laptop when a picture is taken and send the required data to the server through the Internet.

Before any communication, this Bluetooth application must synchronize the time between the two devices, as all the data are time-ordered.

GPS data are received and transferred to the laptop from PDA through Bluetooth every second. Sensor data are also received every second and saved in the laptop together with the GPS data with the same received time. To summarise, in the file, for each second, there is a pair of GPS and orientation data. When the user takes a picture and sends it to the server for processing, system also automatically send the corresponding GPS and orientation data to the server. "Corresponding" means the data received time is the same as the picture taken time.

WLAN

Several WLAN access points have been set up in the city so that PDA user can access Internet through WLAN with higher network quality and greater speed in that area, which is always crowded and with bad network.

GPRS

GPRS (General Packet Radio Service) is integrated with the *iPAQ 3870*. This is the normal way to access the Internet when WLAN is not available.

2.4 Flowchart of User Operation Sequence

Before user takes pictures for recognition, he must first start a custom-built application on the PDA. This program is to receive the GPS and orientation data from the devices, to transfer data between the laptop and PDA in Bluetooth, to detect if any picture is taken and also to request the corresponding GPS and orientation data (with PDA system time corresponding with the picture) from the laptop and send them to the server together with the image.

User then starts the camera application (this application is from the camera manufacturer and different from the custom-built application) on PDA. When the user presses the button to take any picture, background program (custom-built application) records this PDA system time for this action After he is satisfied and decides to use

the picture for building recognition, background program will request the corresponding GPS and orientation data with corresponding PDA system time from the laptop and send them to the server together with the picture. When the data are received by the server, server first runs 3DMax to generate a reference image from the same position and angle (based on the incoming GPS and orientation data from the user) in the 3d model online (as showed in Figure 5). And then attempts to match the user PDA picture with the reference image. If these two pictures are identified to be the same building, server will provide information on the object and location. Otherwise, a notice "object unidentified" will be sent to PDA user. But user location in a 2D map is still available for the user (operation is showed as a flowchart in Figure 3).

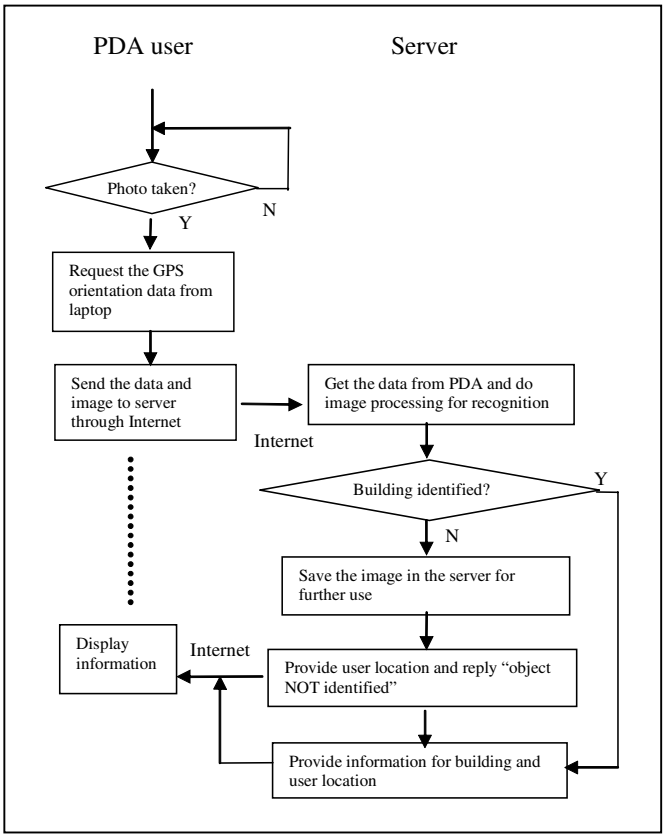


Fig. 3. Flowchart for user.

3 Object Recognition Methods

There are two building recognition methods applied in this project, line detection and colour based segmentation.

3.1 Hough Transform

The first method for object recognition relies on line detection. Several methods of line detection have been developed in the past decade. Hough Transform (HT) [15, 16, 11] and Radon Transform (RT) [13] are the two most important among them. These two methods can transform two-dimensional images with lines (original coordinate plane) into a domain (Hough space) of possible line parameters, in which each line in the image will produce a peak positioned at the corresponding line parameters. In the original coordinate space (image coordination), lines are represented using the form $y = ax + b$. However, in the Hough space (parameter coordination), lines are described in other forms. The most popular form expresses lines among them is in the form *rho = x*cos(theta) +y*sin(theta)* [4], where *theta* is the angle and *rho* the smallest distance to the origin of the coordinate system, also known as a polar coordinate system. In the image space, a line is made up of dots. However, a dot is displayed as a sine wave in the parameter space. The intersection of different sine waves represents the line, which is made of all these points, as shown in Figure 4. The intersection with more waves going through means there are more points located on this line. We call this intersection a "peak". After sampling the image, we are able to find peaks in the parameter coordination that represent the main lines in the image.

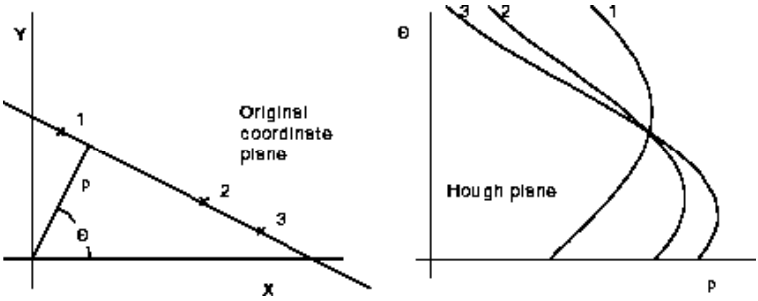


Fig. 4. Hough/Radon Transform.

In this experiment, RT is applied with some modifications.

First, the sample angle *theta* was set to sample more points in the vertical and horizontal areas and fewer in the other directions, as lines in buildings tend to be found in these areas.

Another is to do some pre-processing and post-processing on the detected lines. Pre-processing includes different filtering and colour space conversions. The method for post-processing is that we set the difference between the lines parameters we found in the image must not be within the areas we defined, e.g. the difference between two line angles must be more than 10º and *rho* to be more than 30 pixels. If parameters are within these areas, it means these two lines are very close and they are probably the same lines (see Figure 7 (b)).

3.2 Segmentation

The segmentation of images based on colour and texture cues is formulated as a clustering problem. Small image pixels are grouped together on the basis of local colour

space statistics, which is captured by Gaussian models. Clustering is one of the fundamental methods for image segmentation [5, 10]. It is implemented in the following steps [5]:

1. Data representation: The data types represent the objects in the best way to stress relations between the objects, e.g. similarity.
   There are three major types of data representation, vectorial data, distributional data and proximity data. In this project, data are represented in distributional data, which are described by an empirical probability distribution or histogram $P\{x|o\}$ over features. $x \in F$.
2. Modelling: How to formally characterize interesting and relevant cluster structures in data sets.
   The goal of modelling is to assign objects with similar properties to the same clusters and dissimilar objects to different clusters. For histogram data, distribution clustering objects are grouped according to the similarity of their histograms $\hat{P}\{x|o\}$ with a cluster specific prototypical distribution of features which is parameterised by θ_α . The natural distortion measure between two histograms is defined by the Kullback-Leibler divergence.
3. Optimisation: How to efficiently search for cluster structures.
   K-Means clustering is applied. It is a least squares partitioning method that divide a collection of objects into K groups.
4. Validation: how to validate selected or learned structure.

4 Results

4.1 City Model Image

The model image (Figure 5) is treated as a reference. So, generating the correct image is very important to this project. Figure 5 shows the image rendered by a *3DMax* script based on the user GPS and orientation data. This reference image will be attempted to match the user image (Figure 6).

4.2 Radon Transform

Figure 7(a) and (b) show the lines found before and after the post-processing, which was mentioned in Section 3. Figure 7(a) contains more errors in vertical line detection (inside the oval in long dash). This is because around those edge areas, the dots are very dense and noisy (caused by the pattern in the real image), and the computer will misinterpret a single line as several lines. The parameters (*rho* and *theta*) of these lines are very close to each other. Based on this knowledge, we can apply some post-processing to eliminate this error. After the modification, lines in Figure 7(b) are more reasonable. Figure 8(b) shows the lines from the model image after modification. You can see some different between Figure 7(b) and Figure 8(b). This is caused by the difference on the pattern and also the accuracy of the orientation and GPS data. However, as we will modify the GPS to DGPS (Differential GPS) data in the near future, this result will be improved afterwards.

Fig. 5. Reference Image from the 3D model.

Fig. 6. User image from client PDA camera (600x800 resolution).

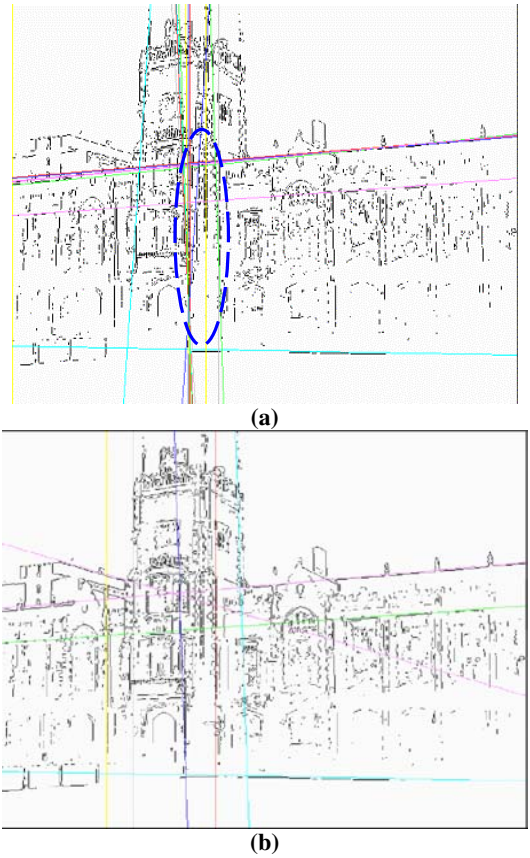
Table 1 displays the parameters for the user image before and after post-processing and the reference image (from the 3D model) after modification. In this table, each pair of *rho* and *theta* defines a peak in Hough space (shown as the stars in Figure 8(a)). Figure 8(a) represents the peaks in figure 8(b). In image space, each of these peaks represents a line as Figure 8 (b).

The line parameters for Figure 7(b) and Figure 8(b) do not perfectly match. However, if we allow a small difference between the line parameters, e.g. $\Delta theta <= 3°$ and $\Delta rho <= 50$ pixels (actually this is a small error), then Figure 7(b) and Figure 8(b) have 6 lines matched. In Figure 7(b), Line 02, Line 03, Line 04, Line 07, Line 08 and Line 10 are respectively matched Line 04, Line 01, Line 06, Line 03, Line 07 and Line 10 from Figure 8(b). This method will be used in conjunction with the segmenta-

tion approach and modelled data will be compared with sampled images in order to determine the usefulness of each method.

Table 1. Line parameters.

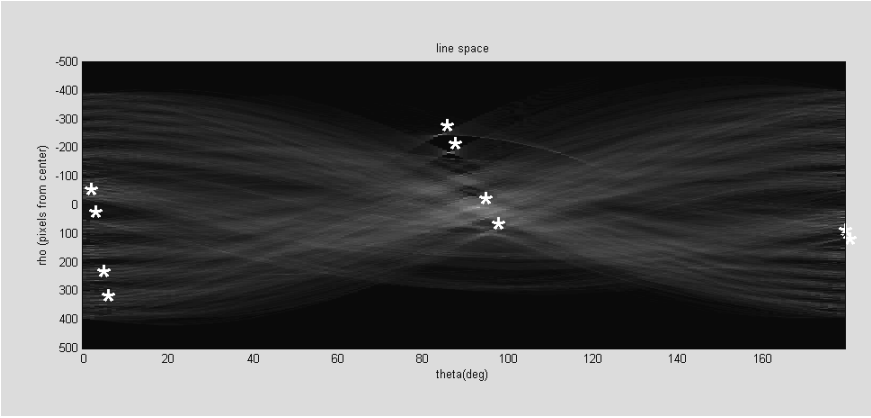
	Lines in Figure 7(a)		**Lines in Figure 7(b)**		**Lines in Figure 8(b)**	
	rho	*theta*	*Rho*	*theta*	*rho*	*theta*
Line 01	-398	0	-398	0	-183	86
Line 02	67	95	67	95	.-244	84
Line 03	-222	89	-222	89	9	93
Line 04	-71	0	-71	0	96	96
Line 05	-91	0	-91	0	56	1
Line 06	66	95	-117	0	-21	0
Line 07	-117	0	8	94	151	179
Line 08	-74	2	200	176	264	3
Line 09	-119	0	19	78	349	4
Line 10	8	94	139	179	120	178



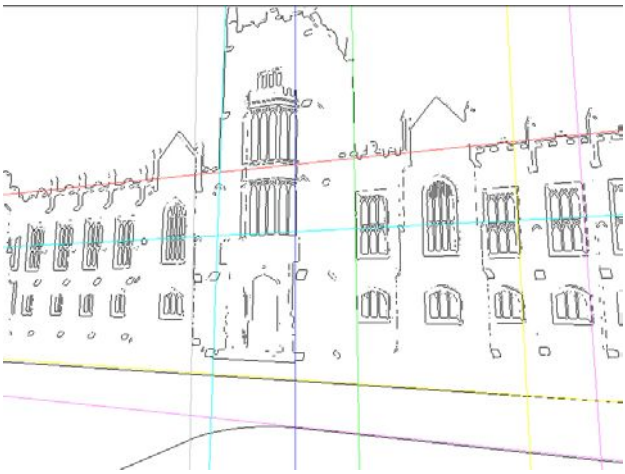
(a)

(b)

Fig. 7. Lines detected in the user images using the Radon Transform (a) before post-processing and (b) after post-processing.



(a)



(b)

Fig. 8. Lines detected using the Radon Transform (a) shows the lines of (b) in Hough space, and (b) are the lines in Figure 5 after postprocessing.

4.3 Segmentation

This experiment is to apply the segmentation method to each picture and find the centre for each group. The results show that the two different pictures for the same building have close centres while the centres for a different building are more different (showed in Figure 10).

In order to arrive at an initial estimate of the underlying Gaussian alphabet for the later stages of clustering, a conventional Gaussian mixture model estimation step is carried out with the colour values of the image pixels as input data (this experiment use Hue and Intensity). Each input data (the mean and standard deviation of a Gaussian model) is generated from a block of 6x8 image pixels. In other words, the total input data is 100x100 (as the image is 600x800). The following work is to cluster this

10,000 data set into 3 groups according by Hue or Intensity values. In this case, K-Means is applied to do the clustering. Results of the segmentation are showed in Figure 9.

Picture4

Picture5

Picture9

(a)

	Hue	**Intensity**
Picture 4		
Picture 5		
Picture 9		

(b)

Fig.9. Segmentation for 3 different pictures.

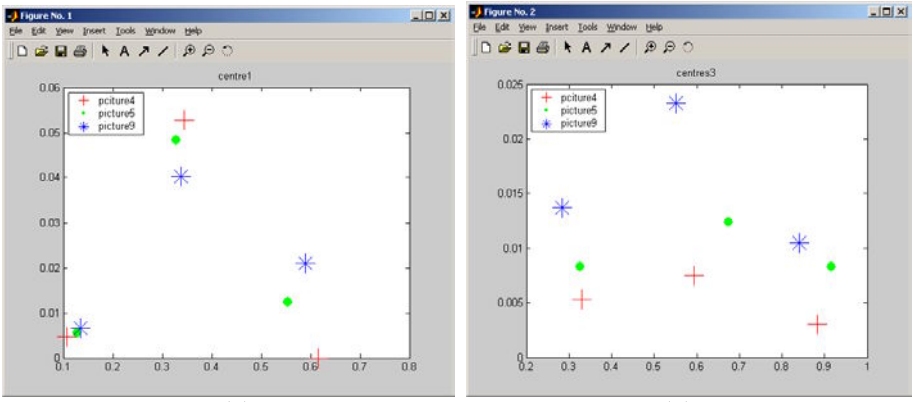
Figure 9 (a) shows the 3 PDA camera pictures (with resolution 600 x 800) taken by different people in different time. The problem with different users, taking different views, but expecting the same result, is clearly evident. This also allows the robustness of the building identification to be tested. Later versions of the system will direct the user to take further pictures or assist them in taking better pictures. Figure 9 (b) shows the segmentation based on the Hue and Intensity value, where the image has been processed so that each pixel in hue or intensity belongs to one of three groups.

Table 2 shows the average and standard deviation of each of these groups. For example, based on Hue, there are 3 centres (for 3 groups) for Picture 4 [0.1077 0.0046], [0.3444 0.0527 and [0.6160 0.0149]. In the first centre [0.1077 0.0046], 0.1077 is the mean value of this group Gaussian model and 0.0046 is the standard deviation. These results are all plotted in Figure 10. The centres for the pictures in Figure 9(a) are from two different red-brick buildings, but even here, it can be seen that the hue and intensity values of the building show significant differences from the other. Work is presently developing appropriate limits for the segmentation sets.

Table 2. Centre for each group in segmentation.

	3 group centres based on Hue	**3 group centres based on Intensity**
Picture 4	Centres-hue = 0.1077 0.0046 0.3444 0.0527 0.6160 0.0149	Centres-intensity = 0.8831 0.0030 0.5934 0.0075 0.3293 0.0053
Picture 5	Centres-hue = 0.1286 0.0056 0.3282 0.0483 0.5537 0.0123	Centres-intensity = 0.9162 0.0083 0.3261 0.0083 0.6748 0.0124
Picture 9	Centres-hue = 0.1353 0.0065 0.3386 0.0403 0.5901 0.0209	Centres-intensity = 0.2836 0.0137 0.5516 0.0233 0.8409 0.0105

In Figure 10, the cross, dot, and star respectively represent the centres for the hue and intensity values in picture 4, picture 5 and picture 9. The X and Y value of the centre also stand for the mean and standard deviation of the Gaussian model of each group. In other words, these centres are in some way representing the features of the image.



(a) **(b)**

Fig. 10. Centres for each group (a) shows the centres from Hue segmentation and (b) shows the centres from Intensity segmentation.

5 Conclusion

In this paper, a system to help people acquire urban information, including the building and geographical information is presented. It is integrated with different hardware devices, software applications and networks.

With this system, the city model is not only for the use of city promotion, indoor, environment planning or architectural design, but it also offers a useful database for tourists and travelers. We believe our system provides a good demonstration of a PDA application and is especially useful for tourists for its mobility. Its main contribution is that people can travel around without having to refer to maps and guidebooks. The city model is fully used to provide information to people at any time and anywhere, in contrast to fixed kiosks and indoor presentations. Some public space on the server is available for user to keep their pictures temporarily to overcome the limitation of the memory card. Two methods for object recognition have been described and improvements have been discussed.

As PDA is still not a complete system for this research, there have been difficulties in working around the limitations of the device, e.g. as described in section 2.1, PDA does not provide enough interfaces.

The battery is not able to work for a long time. Since this application is especially designed for tourists. It is quite important for the PDA to work for a long time while it is not possible to charge it all the time.

Further works will include:

- Continue the custom-built program, e.g. the management between orientation data and GPS data and how to catch the “take picture” action
- Applying segmentation for building recognition
- Automating of the whole system

Acknowledgements

The authors wish to acknowledge the financial support of the Virtual Engineering Centre, Queen’s University of Belfast, (www.vec.qub.ac.uk).

References

1. American Online’s Digital City http://www.digitalcity.com
2. Banerjee S. et al, Rover Scalable Location-Aware Computing, Computer Science IEEE, Oct 2002.
3. Böhm J., Haala N., Kapusy P., 2002. Automated appearance-based building detection in terrestrial images. International Archives on Photogrammetry and Remote Sensing IAPRS', Volume XXXIV, Part 5, pages 491-495, ISPRS Commission V Symposium, Corfu, September 2002
4. Bock, R. K., Krischer W. Data Analysis BriefBook, Springer-Verlag New York, In corporated, Version 16, 1998. ISBN**:** 354064119X.
5. Buhmann J. Data clustering and learning in Handbook of Brain Theory and Neural Networks, Bradfort Books/MIT Press, 1995

6. Digital City Amsterdam, http://www.dds.nl
7. Digital City Kyoto http://www.digital.city.gr.jp
8. Ding P., Mao W. L et al. Digital City Shanghai: Towards Integrated Information & Service Environment. Digital Cities: Experiences, Technologies and Future Perspectives, Lecture Notes in Computer Science, 1765, Springer-Verlag, pp. 125-139, 2000.
9. Donham J., Fitterman B. et al, 2002. Mobile Computing technology at Vindigo. IEEE Wireless Communications, pp. 50-58, Feb 2002.
10. Puzicha J., Hogmannm T., Buhmann J. M., 1999. Histogram clustering for unsupervised segmentation and image retrieval, Pattern Recognition Letters, pp. 899-909, 1999.
11. Richard O. Duda amd Peter E. Hard, 1972. Use of the Hough Transformation to detect lines and curves in pictures. Pictures Communications of the ACM. Vol. 15, No. 1, pp. 11-15, 1972
12. Sutherland, M. Tweed, C. Teller, J. and O. Wedebrunn, (2002) Identifying the relations between historical areas and perceived values: Field tested methodology to measure perceived quality of historical areas. Unpublished report, School of Architecture, Queens University Belfast.
13. Toft P. The Radon Transform - Theory and Implementation, Ph.D. thesis. Department of Mathematical Modelling, Technical University of Denmark, 1996
14. Virtual Los Angeles http://www.ust.ucla.edu/ustweb/ust.html
15. Walsh D., Raftery A. E., 2001. Accurate and efficient curve detection in images: the Importance sampling Hough Transform. Pattern Recognition, volume 35, 2002
16. Xu L., OJA E. and Kultanen P., 1989. A new curve detection method: Radomized Hough Transform (RHT), Pattern Recognition Letters 11

SmartView and SearchMobil: Providing Overview and Detail in Handheld Browsing

Natasa Milic-Frayling[1], Ralph Sommerer[1], Kerry Rodden[2], and Alan Blackwell[3]

[1] Microsoft Research, 7 J J Thomson Avenue,
Cambridge, United Kingdom
{natasamf,som}@microsoft.com
[2] Instrata Ltd., 62 Kingston St.,
Cambridge, United Kingdom
Kerry.Rodden@instrata.co.uk
[3] Computer Laboratory, University of Cambridge,
Cambridge, United Kingdom
Alan.Blackwell@cl.ac.uk

Abstract. Handheld devices, like PDAs and mobile phones, are increasingly used to access information on the Web. However, because most Web pages are designed for desktop PC screens whereas these devices have small screens, only a small region of a page is visible at a time. Reading and finding information is therefore difficult and requires extensive amount of scrolling, both horizontally and vertically. One way to address this problem is an *overview plus detail* approach to the design. We describe SearchMobil, a working system that supports the user in viewing and searching the Web with a PDA. It is based on our SmartView technology that performs content and layout analysis of Web pages and thus provides foundations for SearchMobil features. We present the results of a user study that shows the utility of the SearchMobil approach and provides further insight in challenges and opportunities that the mobile Web presents.

1 Introduction

With the proliferation of Internet capable mobile devices, such as Personal Digital Assistants (PDAs) and Web enabled mobile phones, deficiencies of current Web publishing practices have become apparent. Mobile devices have small screens; yet most Web pages are designed on the assumption that they will be viewed from a standard desktop screen. Those with complex layout require a certain minimal screen space which mobile devices cannot provide. Such pages are therefore difficult to view on a mobile device without extensive scrolling, both horizontally and vertically.

This problem could be alleviated by a document format that allows authors to describe conventional layout features, such as multiple columns, sidebars, menus, etc., in an abstract and flexible way, and that can degrade gracefully when features cannot be represented reasonably on a given screen size. When the screen size is incompatible with the layout it should preserve as much of the designer's intent as the device can accommodate. Such a document format, however, does not exist yet, and HTML as

F. Crestani et al. (Eds.): Mobile and Ubiquitous Info. Access Ws 2003, LNCS 2954, pp. 158–171, 2004.

the Web's main document format doesn't provide any of the mentioned features. As a consequence, Web designers usually hard code their layout intentions using HTML tables with fixed column widths and small blank images for spacing, effectively turning HTML into a layout description language. This results in rigid, inflexible, fixed-size Web page layouts that cannot be re-flowed to preserve the logical structure when viewed on smaller screens.

Fig. 1. A Web page with a complex design, as seen on a Pocket PC. The original page width is three times the width of the Pocket PC display

Currently, far too little published material on the Web is suitable for mobile devices, despite the fact that most of today's mobile phones have Internet capabilities. This material includes pages with simple layout that display well on any screen, or material specifically targeted to small devices, for example, by use of the *Wireless Application Protocol* (WAP). The vast amount of standard unmodified Web contents, however, remains effectively inaccessible to mobile devices. Thus, a number of attempts have been made to improve upon this situation.

Some of them focus on eliminating the negative effects of (horizontal) scrolling. Breaking a page into discrete sections, i.e., "sub-pages" that can be accessed by clicking 'Previous' and 'Next' links or buttons, provides a decent alternative in some cases ([4], [9], [12]). Wrapping the text to fit the width of the screen, or "linearizing" the two-dimensional layout of the page into a vertical axis, completely eliminates the need for horizontal scrolling. However, the results are long – sometimes very long – documents.

A technique particularly useful for mobile phones involves the suppression or elimination of original data. Instead of the full content of the page, the user is pre-

sented with key sentences or navigation elements, such as links, which serve as a content summary ([1], [2]). The interface allows the user to request additional details about a particular summary element. This approach requires robust summarization techniques if it is to be applied generally. More than often, though, quality summaries require assistance by human editors.

A further concern is the issue of consistency in the search and browsing experience across devices. The underlying premise is that the same 'look and feel' of the Web content across devices is desired by both authors and users ([3], [5], [6], [7], [14]). The task, therefore, is first to design representations of the content that connect the user with the familiar content presentation, as experienced in the desktop environment. Because of the space constraint this is likely to be an *overview* rather than a directly readable and consumable format. The second challenge is to devise interaction techniques that enable quick and effective access to the *detailed view* of the relevant content.

SmartView [6], [7], the Web page analysis technique developed in [3], and WebThumb [14] all provide Web page overviews in the form of static or zoomable thumbnails. Interaction with the graphical overview ranges from simple tapping on a specific region of the graphical overview to more sophisticated WebThumb interactions that include 'picking', 'zooming' and 'panning'. These are non-standard for current PDA browsers but likely features of future versions. A similar *overview plus detail* approach is exploited in SearchMobil [10], an application built on SmartView technology, that supports the user in a variety of search situations: from Web search, facilitated by on-line search engines, to search focused on pages seen by the user, to a simple, within-page 'find' function that helps locate relevant portions of the text quickly and effortlessly.

It is this aspect, the consistent experience across devices that is of interest here. We shall describe in more detail the SmartView technology and its application within SearchMobil. Usability studies of the two provide valuable insight into benefits and drawbacks of this particular overview plus detail approach and shed more light on the general issues of Web access on small devices.

2 SmartView Technology

The SmartView approach recognizes the importance of the content's intended layout, as specified by the author, and the fact that Web pages typically consist of a number of coherent logical units of the content. Whereas in the HTML implementation these units are not explicitly marked, we discover them by analyzing the structure of the page layout. We can then allow the user to select any of these units and view it independently from the rest of the document. These portions are usually simple, non-structured HTML fragments, and can be re-flowed easily to accommodate the narrower screens of mobile devices. The page thumbnail overview is displayed with superimposed outlines indicating the segments (fig. 2, left). The user can navigate to a specific detail region by tapping on one of the outlined areas with the stylus (fig. 2, right).

The user can quickly switch back and forth between the thumbnail overview and the detailed view. While the browser is in the SmartView mode, subsequent access to pages is facilitated by corresponding thumbnail overviews, created as the user executes the links.

Fig. 2. Web page thumbnail providing an overview and indicating the logical segments (left). Detailed view of a selected segment (right), displayed for optimal viewing and reading

2.1 Page Analysis and Decomposition

SmartView relies only upon geometric properties of the HTML page and is, therefore, language independent. It identifies geometric characteristics of page elements by downloading the page content, including all images, and rendering it to a standard width, suitable for viewing on a desktop computer (e.g., 800 pixels wide). From this layout, we create a thumbnail image, sized to fit the screen of the handheld device. The corresponding document object model (DOM) allows us to access and inspect individual HTML elements of the page, such as tables, cells, and forms. We recursively traverse the HTML DOM and consider the sizes and arrangements of these elements. Based on simple heuristics about their width and height, we decide whether a table or a cell should be marked as a "logical section" or whether we should continue the process of subdividing or merging individual elements.

The result of this analysis is a vector of nodes that correspond to tables, cells within tables, rows, and similar elements from the DOM, which are identified as logical sections. When the user requests such a section for viewing, we create an HTML document by extracting the HTML representation of the selected node and all its contents. This HTML segment is wrapped by additional HTML code to obtain a representation of the full path from the root of the DOM down to the node. In this manner we provide

a minimal, yet structurally consistent HTML document that can be displayed by the Web browser in the standard manner.

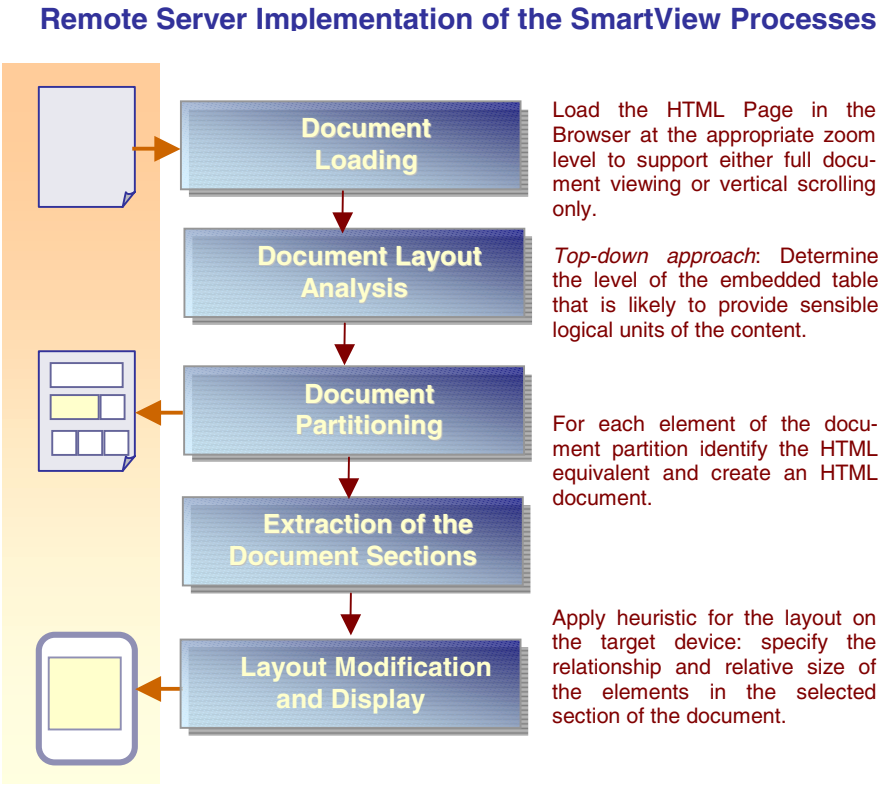


Fig. 3. Steps involved in creating SmartView of a Web page

The current SmartView implementation relies upon a service, hosted outside the device, which performs the analysis of the page layout and page partitioning, thumbnail creation, and layout modification on behalf of the client. The SmartView client is simply an HTML page, with scripts running on the device and forwarding the processing requests to server. With new releases of the browser software for PDAs it will be possible to implement the SmartView feature completely on the device, if desired. It is likely that the thumbnail overview will be replaced by a zoomed out version of the live page, displayed in the scaled down browser window, in the manner similar to WebThumb ([14]). Alternatively, this service can become a part of the publishing process. As the author completes the page design, all the elements, the thumbnail overview, the page analysis, and HTML documents corresponding to individual sections, could be prepared and stored on the Web server for consumption by the client. Even dynamically generated contents could be handled similarly by placing more control over the analysis and delivery of individual sections into the hands of publishers.

An approach similar to SmartView, but more elaborate in the attempt to classify page elements into sidebars, body of the document, menus, and similar, is explored in [3]. Evaluation statistics show that page analysis based on simple geometric properties results in satisfactory page decomposition in 90% out of 50 Web sites tested in the study ([3]).

3 SearchMobil – Search Support for PDAs

SmartView provides a simple yet effective way of making Web pages with complex layout more accessible to mobile devices. However, its overview plus detail approach also provides a framework for exposing a variety of information about the document or its sections. In particular, it can expose the details related to their usage or processing by other services, such as search.

It has been observed that the users often resort to a multi-stage strategy while searching the Web. They specify a broader query to obtain potentially useful pages and then weed out irrelevant ones by skimming the text and carefully inspecting those that seem likely to contain information they seek. On mobile devices this process is far more difficult because of the small screen size. Direct access to relevant parts of the document is thus invaluable. That is the objective of developing the SearchMobil application.

SearchMobil supports the user when he or she has submitted a query to a Web search engine, and is browsing through the results. It facilitates:

Annotation of overview and detailed view with search hits. While the search engine produces a ranking according to its estimate of which documents are most relevant to the query, SearchMobil provides an indication of which part of a particular document is most relevant to the posed query. It adds search hits annotations to the overview and detail presentation of the document to assist the users in judging the relevance of each page region. It quickly directs their attention to the most promising parts of a document.

In the overview of a page, small squares are placed in each region to indicate the number of query term hits it contains. The region with most hits is outlined in red instead of green (fig. 4, (a)). In the detailed view, the query terms are highlighted (fig. 4, (b)). These enhancements are facilitated by providing a local search capability on the device or through a remote service.

Refinement and focusing of search. The set of the top ten result pages is automatically downloaded, and is itself presented in an overview plus detail form: a tabbed 'booklet', as shown in figure 4. The tabs along the right-hand side of the page provide an overview indicator that shares the screen with the current detail region, allowing direct access to each search result. When the user taps on one of the tabs, he or she is presented with the annotated overview of that document.

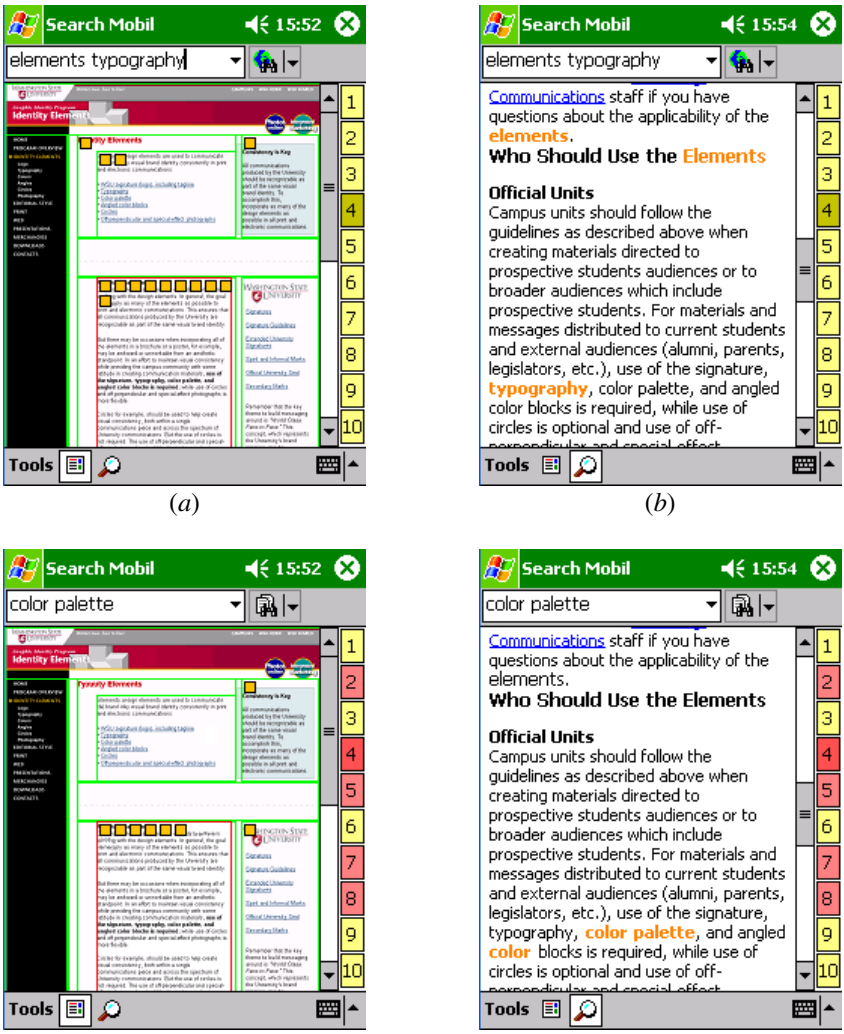


Fig. 4. The SearchMobil booklet interface showing documents in the context of a set of search results: (a) result 4 of a global search for "elements typography"; (b) a selected segment from this page, with highlighted query terms; (c) a local search over the original result set, for "color palette" (the tabs of relevant pages have changed colour), showing the new term hits; (d) the same segment as in (b) with the new query terms highlighted

The user can specify an alternative search query that will be applied to all the documents in the booklet through the local search facility. Tabs of those documents that contain the new query will change the colour (fig. 4, (c)) and annotated with hits accordingly, in the thumbnail overview and the detailed view (fig. 4, (d)).

3.1 User Study

Users engage in a variety of information seeking tasks on the Web. Studies have indicated that "finding" a specific, well defined piece of information, and "gathering information" as a more open ended, research oriented activity, are among the common tasks ([11]). While users are likely to use the desktop to gather information, when mobile they may use their PDAs or mobile phones for fact finding, for example. We wish to investigate how useful SearchMobil is in that context. We are particularly interested in learning whether the current indicators of the "best" region in the document overview help the user to locate information quickly and reliably.

We assume that the "finding" task starts with a query that does not necessarily contain terms that describe information sought for. Indeed, the user is typically looking for a detail that he or she does not know. Thus, the query is only a vehicle to get the user closer to the portion of the text which may contain the relevant detail. The success rate depends on the likelihood that the query terms and the answer co-occur within the same document. In case of SearchMobil, where the user is inspecting individual sections of documents, this requirement is even stricter: query terms and the answer would ideally co-occur in the best scoring section since that one is most prominently marked by the system. Otherwise, the user may be misled by the relevance indicators and it may take them longer to 'recover' and find the correct region. Having this in mind, we designed the study that covers three situations, two of which we expect to reveal possible weaknesses of the current user interface design. We look at the tasks of:

- *Type X*: where the page can be divided up into sections by SearchMobil, and where the answer is in the section that is outlined in red in the overview (i.e., marked as most relevant, according to the search terms used). We expected SearchMobil to perform better than the Pocket Internet Explorer (Pocket IE) browser in these tasks.
- *Type Y*: where the page has only a single section. We expect that SearchMobil would perform slightly worse than Pocket IE, because the overview adds no additional information to the detail view while it presents an intermediary step that takes time to load and review.
- *Type Z*: where the search result page can be divided up into sections by SearchMobil, and where the correct answer is *not* in the section that is outlined in red in the overview. This situation may arise when the user enters search terms that are more general than the actual information requirement. For example, the goal of the task Z2 in Table 1 is to find the postal address of the charity "Shelter", but the search term is simply "shelter". Because the participants are being directed to a section that does not contain the correct answer, SearchMobil will probably perform slightly worse than the standard browser, in terms of time spent searching for an answer.

We selected 12 questions (Table 1) from TREC-9 and TREC-10 Web query collections from query logs of on-line search engines ([13]). For each question we specified the search terms ourselves and submitted them to Google (http://www.google.com).

From each set of search results we retained one document that contains the answer to the question and use it in the experiments. The distribution of answers was such that six were selected for category X, three for Y and three for Z. 24 subjects, 16 male and 8 female, ages 20 to 42, were searching for answers to the questions among selected documents, either using SearchMobil or the Pocket IE browser, on the Compaq iPAQ 3760.

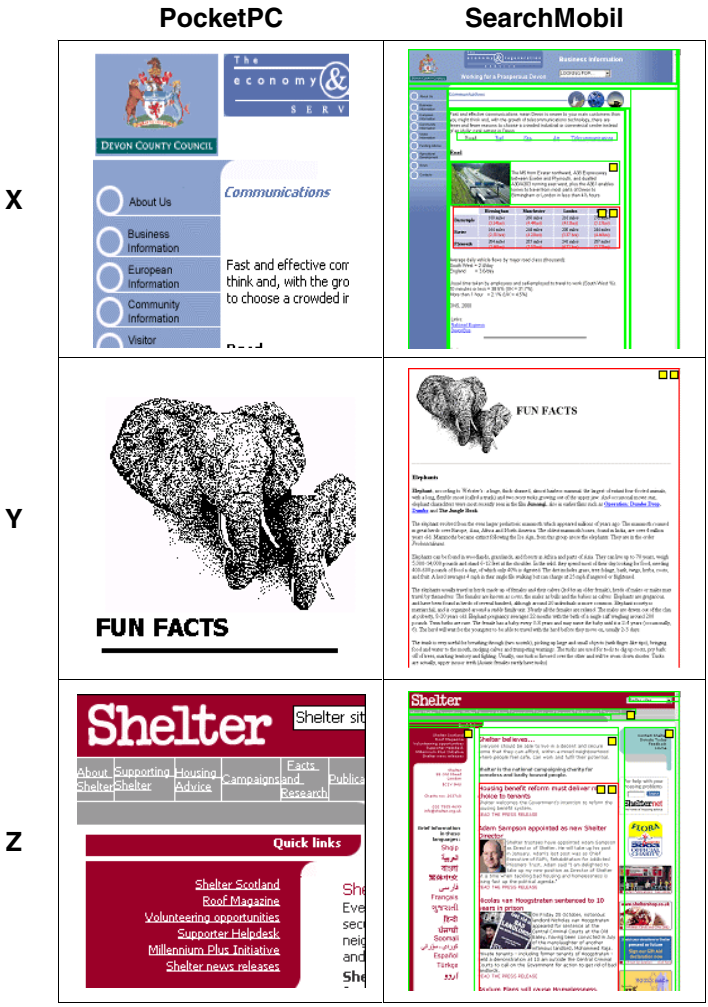


Fig. 5. Examples of the three types of task, showing the portion of the answer page that was visible without scrolling for both types of browser: PocketIE in the left column and SearchMobil page view on the right. In the SearchMobil overviews, the recommended region is outlined in red, with two yellow squares in the top right corner

We measured the time it took to complete each task. Based on Analysis of Variance (ANOVA) we found that there was no major effect of the browser type on a task ($F(1,238) = 1.26$, $p=0.23$); overall, there was very little difference between the two browsers, with the mean time 17.87 seconds for Pocket IE and 18.13 seconds for SearchMobil. However, as we expected, there was a significant difference in performance between tasks ($F(11,238) = 25.0$, $p<0.001$), and a significant interaction between the browser and the task ($F(11,238) = 3.83$, $p<0.001$). These differences are enumerated in Table 2.

Table 1. The questions (and their associated search terms) used in the experiment

Questions and search terms	ID
How many hexagons and pentagons are there on a football? *hexagons pentagons football*	X1
How tall is the Sears Tower, in feet? *sears tower height*	X2
In which year did Hawaii become a state of the USA? *hawaii became state*	X3
Who is credited with inventing the paper clip? *paper clip invented*	X4
What is the salary of a UK member of parliament? *uk mp salary*	X5
How many miles is it from London to Plymouth? *miles london plymouth*	X6
How much was a third-class ticket for the ship "Ttanic"? *titanic ticket cost*	Y1
Which polymer is used to make bulletproof vests? *polymer bulletproof*	Y2
How long is the average elephant pregnancy? *elephant pregnancy*	Y3
What is the telephone number of the University of Sussex? *university sussex*	Z1
What is the postal address of the charity "Shelter"? *shelter*	Z2
Which metal has the highest melting point? *metal highest melting point*	Z3

3.2 Conclusions of the Study

Type X. As we expected, SearchMobil generally outperformed Pocket IE in these tasks as the answer to the question was in the section marked as most relevant. Task X3 was the only exception since only a small part of the most relevant section was visible (at the bottom of the screen) without vertical scrolling. Many of the participants did not see the outlined red section and clicked on the top section instead.

Type Y. We expected that SearchMobil would perform slightly worse than Pocket IE for these tasks, because the SearchMobil detailed view of the page is simply a single long document. This was generally true, although in the case of task Y2, where the answer was quite far down the page, it seems that SearchMobil's term highlighting helped participants to find the answer more quickly. In pages of this type, the user should probably be taken straight to a suitable detail view, instead of navigating via an intermediate overview. In the future, as an alternative visualization of page structure, such documents could be segmented at the paragraph level.

Type Z. Again, we expected that SearchMobil would perform worse than Pocket IE for these tasks. Task Z1 was relatively easy in both browsers – the document was short and, although in the SearchMobil case the answer was not in the red-outlined section, there were only three other sections to check, all of which contained very little text.

Table 2. Mean completion times (in seconds) for each combination of task and browser. Numbers in bold indicate the tasks for which SearchMobil outperforms Pocket IE

	X1	X2	X3	X4	X5	X6	Y1	Y2	Y3	Z1	Z2	Z3
Pocket IE	9.67	25.2	12.7	9.67	18.3	18.1	8.25	29.3	18.3	9.58	31.2	24.3
SearchMobil	**6.42**	**18.8**	19.7	**7.92**	**13.9**	**11.2**	11.1	**18.8**	20.2	9.83	36.1	47.6

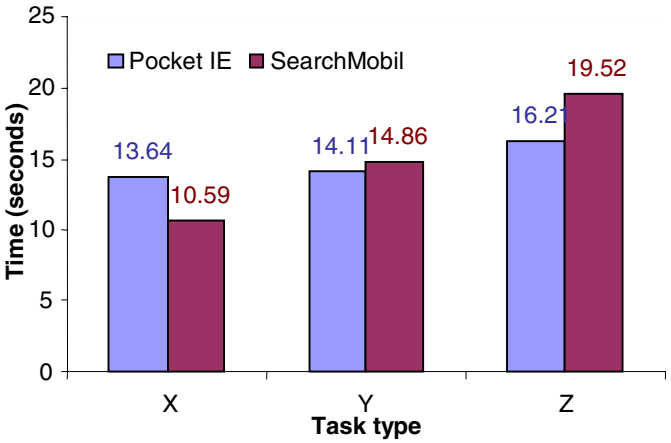


Fig. 6. Histogram derived from the averages of *log(time)* statistics, calculated over all the tasks of a given type (X, Y, or Z). For the obtained average we apply the inverse of the *log* to arrive at the presented time statistics

A written questionnaire completed by the participants provided us with additional insight in the clues that the users rely upon while searching for information ([10]). For example, participant 18 described in note form his search strategy using SearchMobil as follows: "Addresses, and the like, usually located at edge of page, so look there first for those. Other content, check main body of page, top to bottom; if information not found, try another part of the page. Very easy to move around the document and focus

in on particular sections. Having an overview, with a little practice, allows you to guess where the content you want may be fairly well – especially for addresses, etc."

3.3 Analysis of the Task Distribution

In order to assess the practical value of the SearchMobil approach, it is important to investigate how often the user faces each of the task types X, Y, and Z during on-line search. For that purpose, we performed an automated analysis of search results for a sample of queries from the same TREC query collection ([13]).

We selected 116 focused, fact finding queries whose "correct answer" can be matched by a regular expression. Just like in the user study, for each query we manually selected query terms, submitted them to Google, and collected the top 10 search results. We processed the retrieved documents with SearchMobil to obtain information on the query term hits in individual sections of the page and determine the section with the highest relevance score. At the same time we verified whether the answer to the question (as expressed by the regular expression) happens to be in the best scoring section. Incidentally, only 57.5% of the total 1,160 search results contain the right answer (Figure 7, (a)).

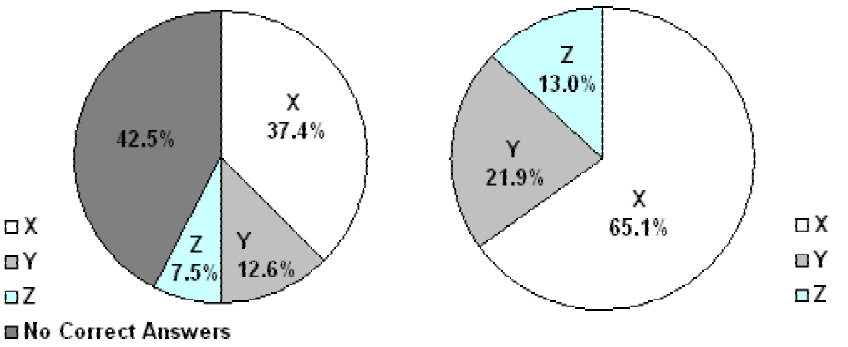


Fig. 7. Distribution of X, Y, and Z types of tasks in: (a) the collection of all search results and (b) the collection of results which contain the correct answers

We found that 65.1% of pages that do contain the correct answer are of type X; thus, the region highlighted by SearchMobil as the most relevant to the query also contains the correct answer. About 21.9% of the pages are of type Y, having no sub-partition of the page into logical units. Only 13% fall into the category Z where the correct answer lies outside the best scoring region for the query terms. For this, relatively small number of pages, the indication of the best region may have an adverse affect on the speed of locating the answer, as observed in the user study.

The fact that about 21.9% of the pages with the correct answer (equivalent to 12.6% of all pages) fall into Y category, provides an opportunity for further refine-

ment of the page analysis to give a finer level feedback, perhaps on the paragraph level, about the relevance of the page content.

4 Conclusions

In order to make optimum use of the small displays on mobile devices for Web searching, it is necessary to separate overview and detail concerns of the search task into different visual renderings. We have discussed three designs that achieve this in different ways. SmartView provides page analyses and a compressed overview visualization to facilitate navigation to structurally significant regions of a page. SearchMobil annotates that overview to show the location of search terms of interest and the most relevant region for the particular search context. The SearchMobil booklet view presents the overview of a set of retrieved pages, using visual tab properties to indicate the degree of their relevance. As with all overview plus detail visualizations, these solutions suit some tasks and information structures better than others. Our evaluation has confirmed this dependency, and highlights the kind of tasks that SmartView and SearchMobil can facilitate.

References

1. Buyukkokten, O., Garcia-Molina, H., Paepcke, A., and Winograd, T.: Power Browser: Efficient Web Browsing for PDAs. Proc. ACM Conference on Computers and Human Interaction (CHI'00), 2000
2. Buyukkokten, O., Garcia-Molina, H., and Paepcke, A.: Seeing the Whole in Parts: Text Summarization for Web Browsing on Handheld Devices. Proc. 10th World Wide Web Conference (WWW10), 2001
3. Chen, Y., Ma, W-Y., and Zhang, H-J.: Detecting Web Page Structure for Adaptive Viewing on Small Form Factor Devices. Proc. 12th World Wide Web Conference (WWW 2003), Budapest, May 2003
4. Jones, M., Marsden, G., Mohd-Nasir, N., and Boone, K., and Buchanan, G.: Improving Web Interaction on Small Displays. Proc. 8th World Wide Web Conference (WWW8), Toronto, Canada, May 1999
5. Jones, M. and Marsden, G.: From the Large Screen to the Small Screen – Retaining the Designer's Design for Effective user Interaction. IEEE Colloquium on Issues for Networked Interpersonal Communicators. 239(3), pp 1-4., 1997
6. Milic-Frayling, N. & Sommerer, R.: SmartView: Flexible viewing of web page contents, On-line Proc. 11th World Wide Web Conference (WWW 2002), Hawaii, May 2002 http://www2002.org/CDROM/poster/172/index.html
7. Milic-Frayling, N. and R. Sommerer, R.: SmartView: Enhanced document viewer for mobile devices. Microsoft Technical Report MSR-TR-2002-114, November 2002
8. Nielsen, J.: Changes in Usability since 1994. http://www.useit.com/alertbox/9712a.html. December 1997
9. Olsen Jr., D.R.: Bookmarks: An Enhanced Scroll Bar. ACM Transactions on Graphics, 11(3), pp 291-295, July 1992

10. Rodden, K., Milic-Frayling, N., Sommerer, R., and Blackwell, A.: Effective Web Searching on Mobile Devices. Proc. HCI Conference, Bath, September 2003
11. Sellen, A.J., Murphy, R., and Shaw, K.L.: How knowledge workers use the web. Proc. ACM Conference on Computers and Human Interaction (CHI'02), pp 227-234, 2002
12. Spence, R.: Information Visualization. ACM Press, New York, 2001
13. Voorhees, E.: Overview of the TREC 2001 Question Answering Track, Proc. TREC 2001, NIST, pp 42-51, 2001
14. Wobbrock, J.O., Forlizzi, J., Hudson, S.E., and Myers, B.A.: WebThumb: Interaction Techniques for Small-Screen Browsers, UIST'02, Paris, October 2002

Compact Summarization for Mobile Phones

Yohei Seki[1], Koji Eguchi[1,2], and Noriko Kando[1,2]

[1] Department of Informatics, The Graduate University for Advanced Studies (Sokendai)
seki@grad.nii.ac.jp
[2] National Institute of Informatics (NII)
Tokyo 101-8430, Japan
{eguchi,kando}@nii.ac.jp

Abstract. In this paper, we propose a new summarization method appropriate for sending text to mobile phones. In mobile access research, an important issue is how to display compact and informative summaries on a screen much smaller than that of an ordinary computer. Documents with varieties of genres presenting information such as opinions, evaluations, etc. have been published on the Web. Most previous summarization research, however, has focused on factual information and topics in documents. For a document that asserts the author's opinion, we assumed that combining factual information and subjective information such as opinions would be effective to produce short but informative summaries adequate to comprehend the contents of the original documents. We propose a summarization method that exploits the typical text structure of the genre. We test the effectiveness of the proposed methods by asking three users who use the genre of "columns" in ordinary life to evaluate summaries in aspect of the recognition test of important sentences and to demonstrate their comprehension of original documents. With the comprehension test, our method which was based on the usage of sentence types was evaluated to be more informative than the existing methods.

1 Introduction

In the field of natural language processing, automatic summarization research has played an increasingly important role [1]. In mobile access research, an important issue is how to display compact and informative summaries on a screen much smaller than that of an ordinary computer. In this paper, we propose a new summarization method appropriate for mobile phones. Documents with varieties of genres presenting information such as opinions, evaluation, etc. have been published on the Web. *Automatic summarization* can be defined as employing technological methods to present the important content of texts in a condensed way, such that it will still meet the user's information needs. Summarization research, however, has mainly focused on factual information and topics in documents heretofore. We use the genre property of the original document to produce an informative summary. Where authors have stressed their assertions, it is also effective to extract subjective information such as opinions, evaluations, prospectives, speculations, attitudes, and emotions.

To produce summaries of the important contents of original documents, it is effective to use the genre of the original documents and the typical structure of the text.

F. Crestani et al. (Eds.): Mobile and Ubiquitous Info. Access Ws 2003, LNCS 2954, pp. 172–186, 2004.

For documents that assert authors' opinions, we assumed that as well as factual information, summaries that contain the combination of factual information and subjective information such as opinions, evaluations, and prospectives would be effective in comprehending the original documents.

In this research, we chose columns in newspaper articles that report the economic situation as examples of documents that assert subjective information such as opinions: i.e., Japanese Nikkei Business Daily column articles "Business Today", and Japanese Nikkei Financial Daily column articles "Position". We assumed the situation of time-pressed businessmen obtaining compact current economic information through mobile phones.

To implement our summarization method, we surveyed the text structure of columns as recognized by four assessors. Our approach was similar to the approaches proposed for scientific articles [2] and legal texts [3], but we focused explicitly on the subjective information.

In addition, summarization is a process of changing the lengths of input sentences, and so information presentation technology for restricting size is essential [4]. Several approaches for summarization for mobile phone or small screens [5–7] have been proposed. Corston-Oliver [6] proposed several heuristic compaction rules for character-sensitive reduction. The problem of sentence-weighting approaches for small screens has not been discussed sufficiently.

Sweeney et al. [5] showed that headlines and three versions of summaries for different compression rates were effective for information retrieval. The summaries for this usage were called "indicative" summaries. Mani [1, p. 8] explained this term as follows: *an indicative abstract provides a reference function for selecting documents for more in-depth reading*. In contrast, "informative" summaries for small screens were not discussed sufficiently. Mani [1, p. 8] also explained this term: *an informative abstract covers all the salient information in the source at some level of detail*. Our focus is different from others in that we have focused on the informativeness of the summarization.

For evaluation, we experimented using tests of recognition of important sentences and comprehension of original documents by three business persons with sufficient expertise in the economic field. To display summaries on the very small screens of mobile phones, the original documents must be edited to be brief. Our goal was to produce informative summaries and we evaluated summaries from the viewpoint of "information per character".

This paper consists of six sections. In Section 2, we propose our summarization method based on text structures for columns. In Section 3, we show our results. In Section 4, we explain the evaluation using "important sentence recognition" tests and "questions for user comprehension of original documents" tests. In Section 5, we introduce our implementation briefly. Our conclusions are presented in Section 6.

2 Methodology

In this section, we explain the methodology of our proposal. We describe our motivation and overview of our new method based on text structure to meet user's factual and subjective information needs.

2.1 Our Proposal: Summarization Based on Text Structure

We propose a summarization method based on text structure appropriate for mobile phones to produce compact, balanced, and informative summaries. We chose newspaper column articles about economics as examples. These texts contained both factual content and subjective information such as opinions, evaluation, etc. In the case that a document asserts an opinion, we assume that combining factual information and subjective information should be an effective way to produce short but representative summaries with sufficient information to comprehend the contents of the orginal documents.

This method exploits the typical text structure of the genre for factual/subjective information needs. In Section 2.2, we introduce existing summarization methods for factual information. Some of these were employed as weighting functions in our summarization strategy. In Section 2.3, the text structure of columns is detailed. Our summarization strategy was shown in Section 2.4 to balance factual and subjective information.

2.2 Existing Summarization Methods for Factual Information

In this section, we explain summarization methods that we have already proposed to extract sentences including factual information [8].

The summarization process was carried out in two stages: important sentence extraction and then the transformation processes. Important sentence extraction is based on five weighting approaches, discounted by sentence length. In this experiment, we used three conventional weighting approaches [9], such as "sentence position weighting", "words weighting in headlines", and "words weighting based on TF*IDF", and two combination of these. We used the lead method, which is effective for newspaper summarization, as a baseline and compared it with the results of important sentence extraction based on the five weighting approaches [8].

Therefore, the summarization methods that we used were as follows:

(a) The lead method (baseline)
(b) Position weighting
(c) Headline weighting
(d) TF*IDF weighting
(e) Position weighting multiplied by headline weighting (i.e., (b) $\times$ (c))
(f) Position weighting multiplied by the sum of headline weighting and TF*IDF weighting (i.e., (b) $\times$ ((c) + (d))).

Three sentences were extracted based on these strategies. We collected 10 column articles randomly and extracted three sentences from each based on these methods.

We detail the lead method and the five basic weighting approaches.

(a) The lead method
The lead method is a popular and effective method for summarization of newspaper articles. In this strategy, the first three sentences are extracted.

Table 1. (Sentence) position weighting

Relative position	$0 < x \leq 0.1$	$0.1 < x \leq 0.2$	$0.2 < x \leq 0.3$	$0.3 < x \leq 0.4$	$0.4 < x \leq 0.5$
Weights	0.585	0.115	0.07	0.04	0.025
Relative position	$0.5 < x \leq 0.6$	$0.6 < x \leq 0.7$	$0.7 < x \leq 0.8$	$0.8 < x \leq 0.9$	$0.9 < x \leq 1$
Weights	0.02	0.03	0.02	0.02	0.075

(b) Position weighting
We investigated position weights using NTCIR-2 [10, 11][1] Text Summarization Challenge (TSC) data, when we could evaluate summaries of newspaper articles that were 20% of the original article's length. We computed the relative positions of important sentences with these data, computed as sentence number divided by total number of sentences. The weights of relative position were divided in 10 increments from zero to one. The result shows that the heading-part weight ($0 \sim 0.1$) was not the highest. Therefore, we modified the other part weights ($0.1 \sim 1$) by dividing by two and the total resulting weights were added to the heading-part weight. The resultant position weights are shown in Table 1.
We used these weights divided by each sentence's length (in characters) and extracted three sentences.

(c) Headline weighting
We also evaluated a sentence extraction method that was weighted by words contained in the headline. We used the Japanese morphological analyzer[2] and extracted compound noun phrases and verbs from headlines. We weighted each sentence if it contained an extracted phrase. We also used this weighting divided by sentence length to extract three important sentences.

(d) TF*IDF weighting
TF*IDF was computed based on the product of the noun term frequency and the logarithm of inverse document frequency among the column articles for one year (about 240 articles for each column). This weight was also divided by each sentence length (in characters).

(e) Position weighting multiplied by headline weighting
We extracted three sentences based on the product of position weighting and headline weighting.

(f) Position weighting multiplied by the sum of headline weighting and TF*IDF weighting
We extracted three sentences based on the product of position weighting and the sum of TF*IDF weighting and headline weighting.

2.3 Text Structure for Columns

In this section, we detail the text structure of column articles to implement our new approach that will be described in Section 2.4.

[1] http://research.nii.ac.jp/ntcir
[2] http://chasen.aist-nara.ac.jp

Table 2. The number of sentences for each type

	S1	S2	S3	SA	Avg.	Auto.
Main descrption	17	22	14	10	15.8	10
Elaboration	92	28	58	73	62.8	90
Background	66	120	181	86	113.3	75
Opinion	12	27	48	26	28.3	64
Prospectives	16	27	17	17	19.3	21

S1 $\sim$ S3: three annotators' annotations. SA: the first author's annotation.
Avg. = Average number of sentences over four annotators.
Auto. = System annotation.

To investigate the typical text structure of the genre for columns, we asked four assessors to annotate 10 columns with five sentence types that indicate the communicative information type. The definitions of the five sentence types are as follows [12]:

1. "Main description": the main contents in a document.
2. "Elaboration": "main description" is detailed.
3. "Background": history or background are described.
4. "Opinion": author's opinion.
5. "Prospectives": future prospects are expressed.

Then, we investigated the tendency of types for important sentences judged by three business persons who have sufficient expertise in the economic field.

We chose three human annotators (not the same people as the important sentence extractors) and compared results of important sentence extraction. Annotators were graduate university students and did not have much expertise in the topics covered by the column articles. Therefore, the annotation results were different from each other. The number of sentences for each type is shown in Table 2. The first author (one of four annotators) annotated sentence types. The "precision" and "recall" for the five types of the three annotators' results against the first author's annotation are shown in Table 3. We regard "precision" as $\frac{|Agreed\_annotation|}{|Annotator's\_annotation|}$ for each type and "recall" as $\frac{|Agreed\_annotation|}{|First\_author's\_annotation|}$ for each type here. We use the "F-measure", defined as a convenient way of reporting precision and recall in one value : $\frac{2 \times Precision \times Recall}{Precision + Recall}$.

We tested Cohen's κ (the kappa coefficient) which was also used in [2, 13] among four annotators' annotations with SPSS as in Table 4. This compares the total probability of agreement to that expected if the rating were statistically independent: κ varies between 1 when agreement is perfect and -1 when there is a perfect negative correlation; $\kappa = 0$ is defined as no correlation. These results showed that four annotator's annotations had a "slight" or "fair" correlation [2]. "Main description" type showed good agreements among annotators, but "Elaboration" type did not show good agreements among annotators. The first annotator (S1) did not show good agreements with other annotators, but the second annotator (S2) showed better agreement with other annotators. Recent investigations have demonstrated that the agreements between annotators could be improved with inter-annotator interview session or instruction modification such as bias-correction approach [13], but we did not take this approach due to time

Table 3. Three annotators' precision, recall, and F-measure if the first author's annotation is gold standard

	Precision (%)				Recall (%)				F-measure (%)			
	S1	S2	S3	Average	S1	S2	S3	Average	S1	S2	S3	Average
Main description	29.4	40.9	42.9	37.7	50.0	90.0	60.0	66.7	37.0	56.3	50.0	47.8
Elaboration	22.8	28.6	31.0	27.5	28.8	11.0	24.7	21.5	25.5	15.8	27.5	22.9
Background	33.3	40.8	32.6	35.6	25.6	57.0	68.6	50.4	28.9	47.6	44.2	40.2
Opinion	50.0	29.6	29.2	36.3	23.1	30.8	53.8	35.9	31.6	30.2	37.8	33.2
Prospectives	25.0	29.6	17.6	24.1	23.5	47.1	17.6	29.4	24.2	36.4	17.6	26.1
Macro avg.	32.1	33.9	30.7	32.2	30.2	47.2	45.0	40.8	29.5	37.2	35.4	34.0
Micro avg.	28.6	36.6	31.4	32.2	27.4	38.7	47.2	37.7	28.0	37.6	37.7	34.4

S1 $\sim$ S3: three annotators' annotations.
Macro avg. = Average values without consideration of population for each category.
Micro avg. = Average of the agreement number for each category divided by the sum of population.

Table 4. Cohen's κ among four annotators for each type

	S1*S2	S1*S3	S1*SA	S2*S3	S2*SA	S3*SA	Average
Main description	0.429	0.425	0.346	0.59	0.544	0.482	0.469
Elaboration	0.141	0.104	0.017	0.11	0.044	0.103	0.087
Background	0.209	0.029	0.087	0.239	0.254	0.147	0.161
Opinion	0.11	0.222	0.281	0.302	0.242	0.309	0.244
Prospectives	0.233	0.267	0.203	0.176	0.322	0.133	0.222
Macro average	0.224	0.209	0.187	0.283	0.281	0.235	-
All Types	0.158	0.087	0.102	0.216	0.214	0.147	0.154

S1 $\sim$ S3: three annotator's annotation
SA: the first author's annotation
'*': combination of annotators

constraints and the fact that our annotations appeared to be sufficiently consistent, as discussed below.

We investigated the relationship between annotations and important sentences using the following procedure:

- We assigned scores for the extracted sentences according to their ranks in the following manner: 1st rank: 5 points; 2nd rank: 4 points; ...; 5th rank: 1 point.
- We then compared "Relative extraction score" using the following equation:

$$\Sigma(score\_of\_sentence_i)/\Sigma|sentence_i| \quad (1)$$

- '$Sentence_i$' indicates a sentence judged as being of type i, as shown at the beginning of Section 3.

For each of the annotations assessed by the three annotators and the first author, we sorted the sentence types in order of priority, as the results of "Relative extraction score" as follows:

1. Annotator 1
   $Main\ description$ (1.88) > $Prospectives$ (1.81) > $Opinion$ (0.97) > $Elaboration$ (0.5) > $Background$ (0.43) > $None$ (0.14)

2. Annotator 2
   $Main\ description\ (2.2) > Elaboration\ (0.70) > Prospectives\ (0.51) > Opinion\ (0.46) > Background\ (0.24) > None\ (0.18)$
3. Annotator 3
   $Main\ description\ (2.8) > Opinion\ (0.65) > Elaboration\ (0.61) > Prospectives\ (0.35) > None\ (0.30) > Background\ (0.17)$
4. First author's annotation
   $Main\ description\ (3.3) > Elaboration\ (0.79) > Opinion\ (0.56) > Prospectives\ (0.41) > Background\ (0.182) > None\ (0.176)$

These four results were combined as follows:

$$\{Main\ description\} > \{Prospectives, opinion, elaboration\} > \{Background, none\}$$

2.4 Our Summarization Strategy

In Section 2.3, five sentence types were defined and the priority of sentence types for extraction was obtained from four annotators' manual data. There were fewer "opinion" and "prospectives" sentences than "elaboration" sentences as shown in Table 2.

Following these results, we propose a new method, which incorporates the weighting approach (f) described in Section 2.2, considering for a typical text structure, as follows:

1. The "main description" sentence is selected as the first sentence. All sentences were weighted by $(TF * IDF\ weighting\ +\ headline\ weighting) \times position\ weighting$. A sentence with the highest weight is annotated as "main description".
2. The "elaboration" sentence that has the highest weight among sentences of that type is selected as the second sentence.
3. The "opinion" or "prospectives" sentence that has the highest weight among sentences with either of these types is selected as the third sentence.

We postulate the merit of this method as follows:

1. Using sentence types, summaries can be produced from functional aspects.
2. In column articles, sentences with opinion or prospective types reflect the author's subjectivity or arguments.

The results of this method (g) and the syntactic reduction results from it (g') were evaluated in the same way as the six strategies in Section 2.2. The weighting method was based on method (f). In Section 4, we evaluate this method with the important sentence recognition test and user comprehension test.

3 Results

In this section, we explain the results of the automatic annotation of sentence types. Then, we show the results of summarization with the length and compression rates of extracted sentences. The evaluations of the summarization results are described in Section 4.

3.1 Automatic Annotation of Sentence Types

We implemented an automatic annotation program for the five sentence types. The three functional aspect types (i.e., "background", "opinion", and "prospectives") were annotated based on auxiliary verb cues. Using separate newspaper articles over two days (604 articles in total), two annotators manually annotated the sentence types cooperatively. Then, we extracted opinion sentences (113), prospective sentences (97), and background sentences (247), and computed three inverse word trigrams for each sentence type. Then, $10 \sim 20$ cue phrases for each type were extracted.

A sentence that contained headline phrases with the highest weight was annotated as a "main description" type. The remaining candidate sentences were annotated as "elaboration" types. This annotation algorithm was implemented as follows:

1. Mark "main description" candidates
   (a) Analyse the headline with the Japanese morphological analyser and extract noun phrases and verbs.
   (b) Mark sentences containing these extracted phrases as "main description" candidates.
2. Weighting all sentences
   (a) All sentences were weighted by $(TF * IDF\ weighting\ + headline\ weighting)\ \times\ position\ weighting.$
3. Determine "main description"-type
   A sentence with the highest weight is annotated as "main description". The remaining candidate sentences were marked as "elaboration" candidates.
4. Mark "background" candidates
   All sentences except the "main description" sentence were marked as "background" candidates with cue phrases like "year", "month" or past/perfect tense auxiliary verbs.
5. Mark "prospectives" candidates
   All sentences except the "main description" sentence are marked as "prospectives" candidates. Cue phrases contain some nouns ("possibility"), verbs ("expect"), and modal auxiliary verbs ("may").
6. Determine "opinion" and other types
   All sentences except the "main description" sentence are determined as "opinion" types on condition that the sentences contain any subjective auxiliary verbs. The excluded candidates "elaboration"-, "background"-, or "prospectives"-type sentences are allocated accordingly.

At present, we have not implemented annotation methods for the two topical aspect types (i.e., "main description" and "elaboration") using elaborated style such as rhetorical structure.

With this algorithm, 337 sentences in 10 columns were automatically annotated. The columns were subdivided into two types: six Nikkei financial newspaper column articles and four Nikkei industrial newspaper column articles. The five sentence types were annotated as follows: 10 "main descriptions", 90 "elaborations", 75 "backgrounds", 64 "opinions", 21 "prospectives", and 77 sentences with no type.

Table 5. System's precision, recall, and F-measure on the basis of manual annotation

	Precision (%)					Recall (%)					F-measure (%)				
	S1	S2	S3	SA	Avg.	S1	S2	S3	SA	Avg.	S1	S2	S3	SA	Avg.
Main desc.	30.0	50.0	30.0	40.0	37.5	17.7	22.7	21.4	40.0	25.5	22.3	31.2	25.0	40.0	29.6
Elaboration	31.1	10.0	22.2	22.2	21.3	30.4	32.1	34.5	27.4	31.1	30.7	15.2	27.0	24.5	24.4
Background	24.0	50.7	57.3	56.0	47.0	27.3	31.7	23.8	48.8	32.9	25.5	39.0	33.6	52.2	37.6
Opinion	9.4	18.8	31.3	20.3	19.9	50.0	44.4	41.7	50.0	46.5	15.8	26.4	35.8	28.9	26.7
Prospectives	9.5	9.5	9.5	9.5	9.5	12.5	7.4	11.8	11.8	10.9	10.8	8.3	10.5	10.5	10.0
Macro avg.	20.8	27.8	30.1	29.6	27.1	27.6	27.7	26.6	35.6	29.4	21.0	24.0	26.4	31.2	25.7
Micro avg.	21.9	25.4	33.9	31.2	25.0	28.1	29.5	27.7	38.2	30.2	24.6	27.3	30.5	34.3	29.2

S1 $\sim$ S3: three annotators' annotations. SA: the first author's annotation.
Macro avg. = Averaging values without consideration of the population for each category.
Micro avg. = Averaging the agreement number for each category divided by the sum of population.

The "precision", "recall", and "F-measure" for the five types against the four manual annotations are shown in Table 5. We regard 'precision' as $\frac{|Agreed\_annotation|}{|System\_annotation|}$ for each type, and 'recall' as $\frac{|Agreed\_annotation|}{|Annotator's\_annotation|}$ for each type here. We use the F-measure, defined as a convenient way of reporting precision and recall in one value : $\frac{2\times Precision\times Recall}{Precision+Recall}$. The accuracy of annotations made by the system was not so good, but it performed well enough to be used to filter original documents and produce an informative summary. Our system annotated "main description" and "background" better, but did not annotate "prospectives" well.

3.2 Summarization Results

The results for sentence length and compression rates for each summarization method are shown in Table 6. These results show that our summarization method (g) had a tendency to be long as such, and it would be better if the results were edited by syntactic reduction.

4 Evaluation

In this section, we evaluate the seven methods (including one baseline method) without considering syntactic reduction using the important sentence recognition test. The first test corresponds to the ordinary coverage evaluation for summarization. Then, we evaluate four methods based on tests of user comprehension of original documents. In the second test, we test the effectiveness of the proposed methods by asking three users who use the genre of "columns" in ordinary life to evaluate summaries to demonstrate their comprehension of original documents.

4.1 Important Sentence Recognition Test

We consider a typical situation such as a time-pressed businessman obtaining current economic information in compact form on the train. We postulated a mobile phone

Table 6. Sentence length and compression rates

Method	Task	1	2	3	4	5	6	7	8	9	10	Avg.	StDev.
	# of original documents	1202	1255	1137	1281	1281	1215	2110	1865	2115	2144	1560.5	414.7
(a) Lead method	#	131	124	161	136	199	110	148	170	175	170	152.4	25.9
	Compression rates	10.90	9.88	14.16	10.62	15.53	9.05	7.01	9.12	8.27	7.93	9.77	
(b) Position weighting	#	120	124	125	136	111	110	128	123	175	139	129.1	17.6
	Compression rates	9.98	9.88	10.99	10.62	8.67	9.05	6.07	6.60	8.27	6.48	8.27	
(c) Headline weighting	#	51	124	139	79	53	86	61	85	124	71	87.3	29.8
	Compression rates	4.24	9.88	12.23	6.17	4.14	7.08	2.89	4.56	5.86	3.31	5.59	
(d) TF*IDF weighting	#	95	81	163	195	51	132	76	94	107	90	99.4	41.1
	Compression rates	7.90	6.45	14.34	8.20	3.98	10.86	3.60	5.04	5.06	4.20	6.37	
(e) Position × headline	#	137	134	161	136	133	110	129	123	133	139	133.5	12.2
	Compression rates	11.40	10.68	14.16	10.62	10.38	9.05	6.11	6.60	6.29	6.48	8.55	
(f) Position × (TF*IDF + headline)	#	166	89	161	136	199	110	129	177	133	170	147.0	31.7
	Compression rates	13.81	7.09	14.16	10.62	15.53	9.05	6.11	9.49	6.29	7.93	10.01	
(f') Position × (TF*IDF + headline) (Reduced)	#	97	76	100	88	128	69	87	75	89	104	91.30	16.3
	Compression rates	8.07	6.06	8.80	6.87	9.99	5.68	4.12	4.02	4.21	4.85	6.27	
(g) Sentence type	#	132	134	149	147	199	113	129	178	170	232	158.30	34.7
	Compression rates	10.98	10.68	13.10	11.48	15.53	9.30	6.11	9.54	8.04	10.82	10.14	
(g') Sentence type (Reduced)	#	69	71	69	94	128	70	87	72	124	157	94.10	29.9
	Compression rates	5.74	5.66	6.07	7.34	9.99	5.76	4.12	3.86	5.86	7.32	6.03	

'#' indicates number of characters.

screen to display information, and a single view on a mobile phone screen contains only 80 characters. Therefore, our gold-standard summary is very small.

In this section, we compare several methods of producing such small summaries. We collected 10 column articles randomly. Five important sentences were manually extracted by three assessors. These sentences were ranked in the order of importance from first to fifth. The assessors were experts in the field of economics and they understood the column contents thoroughly.

Three sentences were extracted based on the seven methods given in Section 2. We implemented sentence extraction systems and evaluated them. The three assessors extracted five important sentences. 'Scores' were computed for three gold- standard manual summaries with five points for first rank in importance, four points for second rank, ..., and one point for fifth rank. The results are shown in Table 7. Then, the maximum score for each task was calculated, $(3+4+5) \times 3 \times 3 = 108$ points. The average document length for the 10 documents was 33.7 sentences. Therefore, if we select three sentences randomly, the expected score is $(1+2+3+4+5)/33.7 \times 3 \times 3 = 4.005$.

We compare the fixed-length lead method, the five weighting strategies, and our new approach. The lead method is a popular and effective method for summarization

Table 7. Comparing sentence extraction methods

Method	Task	1	2	3	4	5	6	7	8	9	10	Avg.	Summ.	Worse than Lead
	# of original documents	1202	1255	1137	1281	1281	1215	2110	1865	2115	2144		Rank	
(a) Lead method	Score	26	8	22	18	11	24	15	19	18	10	17.1	1	-
	Score/#	0.20	0.06	0.14	0.13	0.06	0.22	0.10	0.11	0.10	0.06	0.118		
(b) Position weighting	Score	23	8	19	18	10	24	10	14	26	0	15.2	5	
	Score/#	0.19	0.06	0.15	0.13	0.09	0.22	0.08	0.11	0.15	0	0.119		
(c) Headline weighting	Score	18	12	27	7	0	12	15	9	13	1	11.4	6	*
	Score/#	0.35	0.1	0.19	0.09	0	0.14	0.25	0.11	0.1	0.01	0.134		
(d) TF*IDF weighting	Score	9	2	4	0	0	15	0	4	1	0	3.6	7	**
	Score/#	0.09	0.02	0.02	0	0	0.11	0	0.04	0.01	0	0.031		
(e) Position $\times$ headline	Score	24	12	22	18	13	24	15	14	15	0	15.7	4	
	Score/#	0.18	0.09	0.14	0.13	0.1	0.22	0.12	0.11	0.11	0	0.119		
(f) Position $\times$ (TF*IDF $+$ headline)	Score	24	12	22	18	11	24	15	14	14	10	16.4	3	
	Score/#	0.14	0.13	0.14	0.13	0.06	0.22	0.12	0.08	0.11	0.06	0.118		
(g) Sentence type	Score	23	17	21	8	11	24	15	10	26	14	16.9	2	
	Score/#	0.17	0.13	0.14	0.05	0.06	0.21	0.12	0.06	0.15	0.06	0.115		

'#' indicates number of characters.
'*' indicates "$0.01 <$ P-value ≤ 0.05 with T-test".
'**' indicates "P-value ≤ 0.01 with T-test".

of newspaper articles, and this strategy was also effective for our experiment's data. We use this method as a baseline.

With a T-test, the difference of the score between the lead method (a) and our new sentence type approach (g) or the three weighting approaches (b, e, f) was not statistically significant. The difference between the two weighting approaches (c, d) and the score of the lead method (a) was significant. We compare the combination weighting method (e, f) and our sentence-type method (g).

For method (e), the result in Table 7 shows the score is 15.7 and the average length is 133.5 characters in Table 6. The "score per character" was 0.119 and this result was better than for the lead method. For method (f), the result was slightly better then (e) but not statistically significant. The result in Table 7 shows a score of 16.4 and the "score per character" was 0.118. For method (g), the sentence-type method, the weighting method was based on method (f). Compared to method (f), the score in Table 7 improved, but the "score per character" worsened. This seemed to be because sentences with opinion or prospective types tend to be longer than the others.

We tested six sentence-length-sensitive compaction methods and compared them with each other and with the lead method. Three of the methods were based on weighting approaches, two were based on combinational approaches, and one was based on sentence types. From the results of these weighting approaches, we found that the two basic weighting approaches were less significant than the lead method, but the other approaches were not significantly less than the lead method. In addition, the combination approach also gave better results than the lead method for "score per character". It would be expected to be useful for mobile phones.

Our goal was to provide economic information effectively for business people. Our experiments showed the weighting combination approach improved summarization for

Table 8. Evaluation of test questions for user comprehension of original documents

	(a.) Lead method				(f.) Position × (TF*IDF + headline)				(g.) Sentence type			
	E1	E2	E3	Sum	E1	E2	E3	Sum	E1	E2	E3	Sum
Average Score	6.9	5.0	5.9	17.8	6.7	5.9	5.2	17.8	6.1	6.7	6.3	19.1
Score/#	0.045	0.035	0.041	0.121	0.044	0.042	0.038	0.124	0.039	0.044	0.042	0.125
					(f'.) Syntactic reduction				(g'.) Syntactic reduction			
Average Score					5.0	4.7	4.4	14.1	4.9	5.5	5.4	15.8
Score/#					0.056	0.055	0.051	0.162	0.051	0.066	0.064	0.181

$E1 \sim E3$: Question sets for user comprehension of original documents prepared by three assessors.
'#' indicates number of characters.

mobile phones, but this evaluation was based on gold-standard summaries manually created by humans. Hand-annotated summaries may afford some indicative elements of the original documents, but we could not decide whether they also afford much information or not. Next, we will describe investigations with questions for user comprehension of original documents.

4.2 Questions for User Comprehension of Original Document through Edited Sentences

In the "lead method"-based summary, either specific or more general information from original documents is extracted. These sentences could provide a reference for selecting documents. However, they may not provide enough information.

A strong case can be made for taking the "sentence type" approach. In task 6, the "sentence type" approach (g) extracted the opinion sentence: "We can say that the reduction in Japanese Yen rates in monetary structure had the effect of downgrading the status of Yen." With this sentence, we can answer comprehension questions such as: "How did the international status of the Japanese Yen change in the past five years?" In contrast, the "weighting approach" (method (f)) extracted the factual sentence: "The IMF established SDR in 1969 to cover the special case of lack of liquidity." This sentence might be useful to answer definition type questions, but we cannot use it to answer questions about comprehension of the original documents that query the author's opinions.

To evaluate whether the information is sufficient, five questions for user comprehension of the original document were created by three assessors. They were business people with sufficient expertise in the economic field and were also the sentence extractors.

The five questions were ranked in order of comprehension degree for the original document. Therefore, scores were computed as the summary evaluation. "Score per character" ("Score/#") was also computed. These results are shown in Table 8.

In contrast to Table 7, the score for comprehension using the sentence-type method improved over the lead method. For "score per character" ("Score/#"), the sentence-type method also had improved scores over the lead method.

The sentence-type method contains opinion and prospective sentences. The output summary contained both factual and subjective information. The sentences in the sentence-type-based summary tend to be long compared to the extracted sentences using the weighting method. Therefore, we tried to reduce the unnecessary parts with syntactic and character reduction rules.

Our system's syntactic parser was written with Perl and the reduction rule was implemented with XSLT. Rule examples are as follows:

- If the parser fails to analyze some part in a sentence, that part is marked with "unparsed" and removed.
- Japanese numeric expressions tend to be long. The system converts them to Arabic numeric expressions.
- Adverbs, adnominal phrases, conjunctions at the beginning of a sentence, and modality elements except negation are removed.

These reduction rules were applied to the sentence type summary. We evaluated the reduced summary with questions for user comprehension of the original documents. The results are shown in Table 8, and the ratio of the average score of (g') for (g) is $\frac{15.8}{19.1} \times 100 = 82.7\%$. Therefore, more than 80% of the content for comprehension of original documents was retained. The average "score per character" was 0.181, and this value was more than was obtained for the lead method (0.121), the weighting method (0.124), its syntactic reduction method (0.162), and the original sentence type (0.125). The average three-sentence length of (g') was 94.1 characters.

5 System Implementation

In this section, we explain our system overview. At present, the Web contents must be adjusted according to whether they are being viewed from a mobile phone, PDA, handheld device, TV, or, of course, from an ordinary personal computer. On a browser with a small screen, an appropriately edited summarization is preferable to avoid excessive scrolling.

There are a variety of services that allow access to information on the Internet from a mobile terminal. For example, Japanese Nikkei News provides information for a fee. Presentation languages such as XHTML mobile profile [14, 15] or Wireless Markup Language (WML) [16] have been developed. To change output languages according to the medium, extensible Stylesheet Language Transformations (XSLT) are generally used.

The summarization process was carried out in two stages: important sentence extraction, and then the transformation processes. Important sentence extraction was based on several basic weighting approaches, discounted by sentence length. The results were stored in XML formats and then the revision process according to syntactic information was carried out. The summarization result was represented in Openwave System's "Openwave simulator"[3]. The screen was limited to approximately 80 Japanese characters at one time before scrolling became necessary. The flowchart of our system is shown in Figure 1.

[3] http://japan.openwave.com/products/sdk/index.html

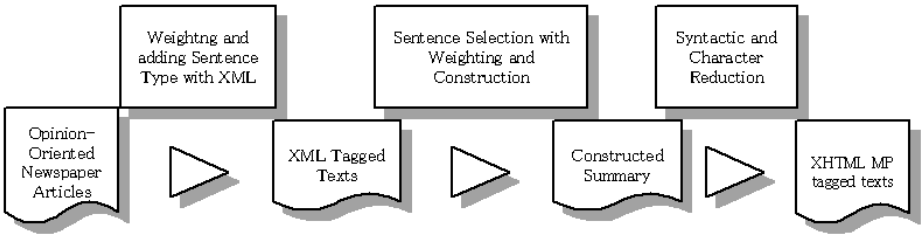


Fig. 1. System flowchart

6 Conclusions

In recent years, many digital documents that include subjective information such as opinions, prospectives, and evaluations have been published on the World Wide Web. We proposed a new summarization method for these documents to combine existing summarization approaches for factual information with the sentence-type approach to convey not only factual but also subjective information. In addition, we investigated the feasibility of sentence-type-based approaches and found a priority among sentence types for gold-standard summaries from hand-annotated documents. We applied this priority to our extraction method.

To generate summaries for mobile phones, we compared several weighting strategies, and combinations of these approaches improved the results of "score per character" over the lead method. The extracted sentences based on our method improved the evaluation based on tests of user comprehension of the original document. This result shows that sentence-type-based summaries are more informative than those produced using the lead method.

References

1. Mani, I.: Automatic Summarization. First edn. Volume 3 of Natural Language Processing. John Benjamins, Amsterdam, Philadelphia (2001)
2. Teufel, S., Moens, M.: Summarizing scientific articles: Experiments with relevance and rhetorical status. Computational Linguistics **28** (2002) 409–445
3. Grover, C., Hachey, B., Korycinski, C.: Summarizing legal texts: Sentential tense and argumentative roles. In: Proc. of the HLT-NAACL 03 Text Summarization Workshop, Edmonton, Canada (2003) 33–40
4. Reiter, E.: Pipelines and size constraints. *Computational Linguistics* **26** (2000) 250–259
5. Sweeney, S.O., Crestani, F., Tombros, A.: Mobile delivery of news using hierarchical query-biased summaries. In: Proc. of ACM SAC 2002, ACM Symposium on Applied Computing, Madrid, Spain (2002) 634–639
6. Corston-Oliver, S.: Text compaction for display on very small screens. In: The Second Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL 2001) Workshop on Automatic Summarization, Pittsburgh, Pennsylvania (2001) 89–98

7. Boguraev, B., Bellamy, R., Swart, C.: Summarization miniaturization: Delivery of news to hand-helds. In: The Second Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL 2001) Workshop on Automatic Summarization, Pittsburgh, Pennsylvania (2001) 99–108
8. Seki, Y., Kando, N.: Dynamic document generation based on tf/idf weighting. In: Mobile Personal Information Retrieval: Workshop held at the 25th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'2002), Tampere, Finland (2002) 57–63
9. Edmundson, H.P.: New methods in automatic extracting. In: Journal of the Association for Computing Machinery 16 (2) 264-285. Reprinted in *Advances in Automatic Text Summarization*, Mani, I., Maybury, M. T., eds. MIT Press (1999) 23–42
10. Kando, N.: Overview of the second NTCIR workshop. In: Proceedings of the Second NTCIR Workshop on Research in Chinese & Japanese Text Retrieval and Text Summarization. National Institute of Informatics (2001)
11. Fukusima, T., Okumura, M.: Text summarization challenge: Text summarization evaluation at NTCIR workshop2. In: Proceedings of the Second NTCIR Workshop on Research in Chinese & Japanese Text Retrieval and Text Summarization. National Institute of Informatics (2001)
12. Kando, N.: Text structure analysis based on human recognition: Cases of Japanese newspaper articles and English newspaper articles (in Japanese). *Research Bulletin of National Center for Science Information Systems* **8** (1996) 107–126
13. Bruce, R., Wiebe, J.M.: Recognizing subjectivity: a case study in manual tagging. Natural Language Engineering **5** (1999)
14. WAP Forum: XHTML Mobile Profile WAP-277-XHTMLMP-20011029-a. Technical report, Wireless Application Protocol Forum (2001) http://www.openmobilealliance.org/wapdownload.html.
15. Wugofski, T.: Creating contents for mobile phones (Japanese language edition). In Boumphrey, F., Greer, C., Raggett, D., Raggettet, J., Schnitzenbaumer, S., Wugofski, T., eds.: Beginning XHTML. Impress Corporation (2001) 613–640
16. WAP Forum: WAP WML WAP-238-20010911-a. Technical report, Wireless Application Protocol Forum (2001) http://www.openmobilealliance.org/wapdownload.html.

Supporting Searching on Small Screen Devices Using Summarisation

Simon Sweeney and Fabio Crestani

Dept. Computer and Information Sciences
University of Strathclyde
Glasgow, Scotland, UK
{simon,fabioc}@cis.strath.ac.uk

Abstract. In recent years, small screen devices have seen widespread increase in their acceptance and use. Combining mobility with their increased technological advances many such devices can now be considered mobile information terminals. However, user interactions with small screen devices remain a challenge due to the inherent limited display capabilities. These challenges are particularly evident for tasks, such as information seeking. In this paper we assess the effectiveness of using hierarchical-query biased summaries as a means of supporting the results of an information search conducted on a small screen device, a PDA. We present the results of an experiment focused on measuring users' perception of relevance of displayed documents, in the form of automatically generated summaries of increasing length, in response to a simulated submitted query. The aim is to study experimentally how users' perception of relevance varies depending on the length of summary, in relation to the characteristics of the PDA interface on which the content is presented. Experimental results suggest that hierarchical query-biased summaries are useful and assist users in making relevance judgments.

1 Introduction

The recent trend towards pervasive computing, information technology becoming omnipresent and entering all aspects of modern living [3], means that we are moving away from the traditional interaction paradigm between human and technology being that of the desktop computer. This shift towards ubiquitous computing is perhaps most evident in the increased sophistication and extended utility of mobile devices, such as mobile phones, PDAs, mobile communicators (telephone/PDA) and Pocket PCs. Advances in these mobile device technologies coupled with their much-improved functionality means that current mobile devices can be considered as multi-purpose information tools capable of complex tasks. In terms of services that are available for mobile devices, there are currently thousands of applications for handheld devices for the different handheld operating systems (PalmOS, Windows CE). In fact, many of these devices are now capable of supporting tasks that are normally only associated with the desktop PC, such as creating word-processed documents, spreadsheets, presentation slides. Similarly, for WAP mobile phones there exists a wide variety of network-based services.

F. Crestani et al. (Eds.): Mobile and Ubiquitous Info. Access Ws 2003, LNCS 2954, pp. 187–201, 2004.

However, there remains a significant challenge in the presentation of information on mobile devices given the inherent constraints of a low-resolution small display area and, in the case of mobile phones, limitations on interaction [10]. And whilst the amount of information available on the web is ever increasing the same degree of proliferation of content has not yet been matched for mobile devices [18]. A possible reason for this lag in growth of content for mobile devices could be that there are accepted approaches for supporting information access on the desktop PC, whereas the same approaches may not be appropriate for mobile devices.

Significant improvements have been made in the interface design of applications for mobile device platforms, particularly for supporting web access. WAP is designed specifically for small handheld wireless devices and the limitations of small display screens and the interaction constraints. However, aside from WAP few Web browsers currently available for handheld devices (PDAs) render content to take into account the very different display capabilities of handheld devices.

In this paper we shall highlight work that has been carried out to improve browsing and searching on small screen devices, as a starting point we shall discuss how searching is currently supported on the desktop PC. We shall then briefly describe some automatic summarisation approaches that have been applied in the context of information retrieval (IR). We shall then present the results of our latest experiment that measured user performance in conducting relevance judgments using summaries presented on a PDA device.

The paper is structured as follows. Section 2 describes existing work on supporting searching for both desktop and small screen devices, also including a brief review of some previous work relating to the use of query-biased summarisation. Section 3 outlines an experiment we carried out investigating the effectiveness of presenting hierarchical query biased summaries on a PDA device. Section 4 presents the details of the experimental set-up used. Subsequently, Section 5 presents the experimental results and analysis. Finally, Section 6 reports the conclusions from our findings and present future extensions of this work.

2 Background

The subject of Information Retrieval (IR) is well established and is an underlying feature of the Internet. The function of an IR system is to locate and retrieve relevant data that is subsequently presented to the user. Traditionally, automatic IR systems are accessed using a desktop PC where the results of a search are presented on a large screen display and where user interaction is supported using a mouse or a keyboard. Users range from experienced experts, with possibly formal training in conducting information searches, to novice users who are at the very least computer literate.

The increased capability of mobile computing devices and the development of infrastructures for supporting intensive wireless communications means that mobile devices can now be considered as information terminals. However, as discussed in [13], information access in a mobile environment is considerably different to conventional IR. Firstly, mobile devices by design are multi-purpose and as a consequence may compromise certain useful features to maximise mobility and diversity. These devices tend

to have low-resolution small display areas and limitations on interaction, particularly in the case of mobile phones. Using these devices can at times be challenging despite the continued improvements to device displays and means of interaction (stylus, T9 predictive text). Any difficulties experienced may be magnified when the functionality of these devices is extended to support new tasks, such as searching for information on the web. Secondly, the profile of a typical mobile device user differs from that usually associated with an IR system user in the sense that computer proficiency may not be assumed. This is apparent when considering the variety in user profiles of the current mobile phone user population. Finally, the retrieval task differs from that assumed under normal IR circumstances due to the nature of conducting searches in a mobile environment. There is a greater risk that user performance may be influenced by outside factors with the increased potential for distractions of noise and interruptions [8]. Further, a user maybe engaged in other activities at the time of searching. There is also the need to consider the type of information being sought, as there may be significant temporal dependencies. For example, consider the scenario of finding information about possible tourist sites available for visiting. In such a case it would be useful that any suggested tourist sites are first checked to see if they are open given the current time of day, and the expected travelling time to the site. All of these factors then influence the way mobile users will conduct searches and view search results.

2.1 Searching on the Desktop

Most search systems present the results of a user query as a list of documents, spanning possibly a number of pages that may or may not be ranked. Users are required to assess each document individually on the basis of relevance to their submitted query. This can be a lengthy process given the often long list of retrieved documents. Approaches have been introduced aimed at reducing the overheads involved in working through the list of retrieved documents, assisting the user in completing their information discovery task.

Ranking the list of retrieved document according to relevance aids the user in this process by presenting those documents the systems considers as best matching the users query higher in the list [1]. Techniques may focus on attempting to improve the quality of the results increasing the number of relevant documents in the retrieved document result set. Relevance feedback is an example of such a technique, where by the system refines a set of results or perform a further search on the basis of user correction [7].

Research in information visualisation focuses on exploring alternative schemes to traditional ranked lists as a means of presenting search results. Many of these schemes make use of colourful highlighting and graphical features to capture aspects of the information access process, with content that is dynamic and can be manipulated by the user [1]. For example, the use of concept ‘landscapes’ to represent document clusters displayed graphically, in 2D as a ‘jigsaw’ with the clusters forming the individual pieces, or in 3D as a ‘map’ with contours describing document similarity and where peaks indicate concentrations of similar documents [5] [25].

Another variation to a plain list of document titles is to include additional information relating to the retrieved document. This additional information then function as a document surrogate providing the user with metadata, such as date of publishing, source, and length of the document, to give more indication about the content of a

document [1]. The inclusion of document surrogates in search results lists has become a standard feature among web search engines, possibly the most widely used of the search systems.

Some systems extend document surrogates to include a short automatically generated extract, which may take the form of the first few lines of the document text. And, in recent times there has been an interest in enhancing document surrogates to better represent the content of the source documents. By applying techniques developed in the field of automatic summarisation the properties of the document surrogate can be improved. The outcome of which is document surrogates that are more representative of the document source and can be tuned to be either informative, contain such information from the document text that instantly fulfils the user's information need, or indicative, provide an indication of whether the particular document is relevant [2].

2.2 Searching on the Small Screen

Small screen devices provide many of the searching functionality found on the desktop PC, ranging from on-device information discovery to searching wider network accessed information resources, such as digital libraries or the WWW. However, whilst similar functionality is provided for such devices in practical terms using such services results in very different user experience [10] [12]. In general terms interfaces for searching on small screen devices have remained largely unchanged, querying is expressed by entry of plain text into a text field and search results are presented as a scrollable list of matches.

Some recent studies have found that supporting information discovery (browsing and searching) on small screen devices, such as PDAs, using interfaces designed for the display area of desktop PCs has a negative influence on task performance [10] [12].

Problems with search interfaces for small screen devices tend to revolve around the scrolling or page requirements when viewing content. Often to make content available for displaying on small screen devices it is not uncommon that long lists of search results are divided into separate pages that contain a reduced number of results. Breaking the content up into smaller manageable chunks is necessary for both transmission requirements and as a means of aiding presentation. However, such techniques have an associated cost that is page-to-page navigation is expensive in terms of user interactions and time [10], both of which may have financial implications (users are likely to be paying for wireless connections or the amount of data they transfer) and may have an impact on the way users use such services. Worst effects are observed if users are required to scroll horizontally [10]. In such cases, it is easy for users to become disorientated and lost within content designed for viewing on much larger screens.

Solutions then to aiding the user in making sense of search results on the small screen can be briefly outlined as follows. As mentioned, presenting only a limited number of results in each result page and limiting the amount of information displayed for each result (Google for the PDA) means that users will not have to view long lists of results. Combining relevance ranking with high precision performance would provide a trade-off to the splitting content over a number of pages and the associated navigation costs. Ideally, the most relevant results would appear in the first couple of pages and would fulfil the users information need reducing the need to go beyond the second page

of results [9]. Alternatively, using schemes such as, WebTwig [11] or PowerBrowser [4] are designed specifically to take account of the limited display area of small screen devices and adapt content presentation accordingly. The basis of these schemes is to provide a more direct, systematic approach to viewing content that requires much less scrolling [12]. It is interesting to observe that both these schemes for accessing web on handheld devices have recently incorporated features that use forms of summarisation.

2.3 Applying Summarisation to IR Results Presentation

Automatic summarisation has been used extensively in the content of IR. As a means of supplementing search results thus aiding the user to make the relevance assessments, and for making the IR process more efficient (using a summarised version of documents to build indexes or for storage, in place of the document full text).

Traditionally, automatic document summarisation has been based on sentence extraction approaches [2] [6] [14]. Advances in sentence extraction have seen the introduction of query-biased methods. Query-biased summarisation methods generate summaries in the context of an information need expressed as a query by a user. Such methods aim to identify and present to the user individual parts of the text that are more focused towards this particular information need rather than a generic, non-query-sensitive summary. Summaries of this type can then serve as an indicative function, providing a preview format to support relevance assessments on the full text of documents [16].

Highlighting recent research into the application of summarisation to aid information retrieval tasks, in particular the use of query-biased methods. Tombros and Sanderson investigated and illustrated the application of query-biased methods for text IR [22]. A later study by Tombros and Crestani looked at evaluating the effectiveness of presenting summaries by different means and the effect this has on users' perception of relevance [21]. Results from their study showed that users' ability to make relevance assessments of documents is highly affected by the way they are presented.

Extending the forms of presentation to include small screen devices, Sweeney, Tombros and Crestani looked at the use of query-biased hierarchical based summaries of newspaper articles presented to users on WAP mobile phones [20]. Defining hierarchical summaries as summaries of variable length, increasing from title only, 7%, 15%, to 30% of the original document length, the study investigated how users' perception of relevance varied depending on the length of the summary, and in relation to the specific characteristic of a typical WAP mobile phone interface. This study suggested that hierarchical query-biased summaries are useful when dealing with small screens and assist users in making correct relevance judgments. The results also highlighted, for WAP mobile phones, a preference for concise summaries that are relatively brief, 7% of the document length (up to a maximum of 3 sentences).

3 Presenting Hierarchical Summaries on a PDA Device

We now report a resent study that continues the theme of investigating users' ability to carry out relevance judgements on textual information presented on non-traditional

IR platforms. We use the same experimental procedure to the study previously mentioned [20], again using the hierarchical query-biased summaries, however, in this study we shall focus on the effects due to displaying content on a PDA interface. Again we are interested in assessing the variation of user performance in evaluating the relevance of full documents, given hierarchical query-biased summaries, and also determining whether there is an optimal size of summary for this type of interface. Similar to the previous experiments we assume the utility notion of relevance [23] as the basis for evaluating the summaries. Further details describing the context for the users' perception of relevance used in this study can be found in [21].

The summarisation system used to produce the summaries for the experiment was the same as that described in [20]. The system uses a number of sentence extraction methods [15] that utilise information both from the documents of the collection and from the queries used. A detailed description of the system can be found in [21] [22] here we shall only briefly describe the output of the summary generation process.

For the purposes of the experiment summary length was treated as a design variable in our system, corresponding to the level of information a user would be presented with in relation to the original document. Each level is intended to provide more information to the user. We consider this design as producing "hierarchical summaries", the root of which corresponds to the minimum level of information. Proceeding down the hierarchy, more and more information is made available to the user, up to a maximum, which corresponds to the full-text of the document.

Four different summary lengths were used in our experiments. It is established that titles convey useful clues about the contents of a document [17], and based on this fact we used titles as the first level of information (shortest summary) a user would be presented with. The other three summary length values were calculated as a percentage of the number of sentences in the original document. Therefore, for each document a number of sentences equal to the 7%, 15% and 30% of its length (up to a maximum of 3, 6, and 12 sentences respectively) were used.

For the experiment we used a HandSpring Visor running the AvantGo[1] web browser. Prior to the start of each user experiment, experimental content was transferred to the device such that users were only permitted to view content offline thus reducing effects of any outside factors that could influence the results, and ensuring consistency with previous experiments.

4 Experimental Settings

4.1 The Test Collection

The documents used were the same as those in the previous experiments, and are a subset of the 1990-92 Wall Street Journal (WSJ) collection of TREC [24]. The TREC-WSJ collection was used in the study both as a data source and as a standard against which the users' relevance assessments were compared, enabling precision and recall figures to be calculated. For this last purpose the relevance assessments that are part of the TREC collection and that were made by TREC "judges" were used (refer to later

[1] AvantGo. http://www.avantgo.com

discussion on 'Experimental Measures'). We used 50 randomly selected TREC queries and for each of the queries, the 50 top-ranked documents as an input to the summarisation system. The test collection then consisted of a total of 2,220 news articles. To provide an indication of the proportion of relevant documents within those used for the experiment, there was a total of 414 relevant documents in the collection with an average of 8.3 relevant documents per query.

4.2 Experimental Procedure

To enable comparisons the same experimental tasks were used: users were presented with a retrieved document list in response to a query (simulated query), and had to identify as many relevant documents as possible for that particular query within 5 minutes. The information presented for each document was automatically generated, query-biased summaries.

The experimentation was carried out with user group of 10 volunteers with above average experience of using computers and mobile devices (mobile phones, PDAs). Each user was initially briefed about the experimental process, and instructions were handed to the user by the experimenter. Any questions concerning the process were answered by the experimenter at this stage. Users were otherwise uninformed of the purpose of the experiments. Each user was assigned a set of five queries randomly chosen among the 50 used. For each query, the user was given the title and the description of each query (i.e., the "title" and "description" fields of the respective TREC topic[2]) providing the necessary background to their 'information need' to allow them to make relevance judgements. Once the user indicated to the experimenter that they were ready to proceed the experiment was started. At that point, timing for that specific query started and the user was presented with a ranked document list, composed of the 50 highest ranked documents, and would be allowed to interact with the PDA. Users could select any document from the list and read its contents (see Figure 1). The document title, and the three levels of summary were used to represent document content. Initially, a user would read the title and then make a decision as to whether to mark the document as relevant/non-relevant or to proceed to the next level of summary by selecting "Next". A user can navigate back to the retrieved document list at any point by selecting "Doc List". At any point the subject could stop the system and instruct it to move on to the next document, or instruct it to show again the previous summary of the current document. Documents judged relevant/non-relevant were marked so by the user on an answer sheet that was prepared for each query. In addition, the user marked the level of summary used to make their decision.

Once the assigned task was completed (i.e. all the documents were marked or the time elapsed), the user was given the next query and the process was repeated. At the end of the experiment the user was given a questionnaire. The purpose of the questionnaire was to gather additional information on the user's interaction with the system: the utility of the document descriptions, the clarity of reading the description through the PDA interface, the level of difficulty of using the interface, and the level of difficulty of the queries.

[2] Examples of TREC topics are available at http://trec.nist.gov/data/testq_eng.html

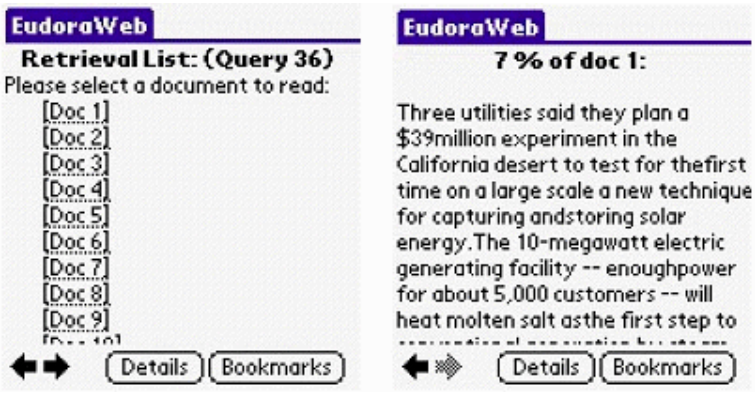


Fig. 1. Examples of screen shots.

There are some limitations to the methodology we used in our experiment. A first limitation pertains to the use of the TREC relevance assessments as the "ground truth" against which user judgments are compared in order to obtain precision and recall values. A second relates to assessing the form-factor of viewing textual content on a PDA device. Current web browsers for handheld devices do not take into account the different display capabilities, and the onus is on the content provider to produce suitable content. The HTML files viewed by user in the experiment were set to "word-wrap" to be consistent with previous experiments, and therefore only partially assessed the effect of page scrolling (horizontally) due to PDA web browser limitations. Finally, a further criticism of our experimental procedure may be the decision to ask the user to identify as many relevant documents as possible within the allotted time. It could be argued that by adopting this approach users maybe encouraged to decide upon the relevance of each document on the minimum amount of information. The result potentially leading to a bias in the decision threshold favouring a "relevant" response. Possibly a better approach would have been to explicitly mention to users that in addition to identifying relevant documents, they must also consider that their performance scores would be penalised if they make mistakes. However, it is fair to assume on the basis of our experimental results (see section 'Results') that the majority of the users (9 out of 10 users) correctly understood the experimental task using the full range of available summaries to make their decisions, with only one user possibly misunderstanding the task and basing their decisions on mainly document "titles".

4.3 Experimental Measures

Experimental measures we used to assess the effectiveness of user relevance judgements were the accuracy and speed of judgements. The speed of user judgements is the time that a user took to assess the relevance of a single document and, to quantify accuracy; precision, recall and decision-correctness were used. In our experiment we focus on the variation of these measures in relation to the different experimental conditions. This is in contrast to the absolute values normally used in IR research.

We define *precision* then as the number of documents marked correctly as relevant (in other words, found to be relevant in agreement with the TREC judges' assessments) out of the total number of documents marked, and *recall* as the number of documents marked correctly as relevant out of the total number of relevant documents seen. A further measure we used to quantify the accuracy of a user's judgment is *decision-correctness*, that is the user ability to identify correctly both the relevant document and the non-relevant (irrelevant) documents. We define decision-correctness as the sum of the number of documents marked correctly as relevant, plus the number of documents correctly marked as non-relevant out of the total number of documents marked for that query.

5 Results

We now report the results of the experimentation outlined in the previous section. A full analysis of all the data produced during the experimentation is outside the scope of this paper. Instead, we present those results that we believe to be most interesting.

Table 1 reports the average precision, average recall, and average time for each user regardless of the summary level used to make the relevance decision. The values report a variation in the precision and recall among the users. It is apparent that some users have high levels of precision (users 4 and 10), while others (in particular user 7) have very low levels. It may be reasoned that the low level of precision cannot be fully explained by a hasty decision, since the fastest two users both show higher levels of precision. It is interesting to notice that the user with the highest level of recall is also the one with the highest level of precision and among the fastest users. The slowest user (user 8) is among the users with the lowest levels of precision.

Table 1. Average precision, recall and time for the overall PDA experiment.

	User										Avg.
	1	2	3	4	5	6	7	8	9	10	
Avg. Precision (%)	66.67	50.00	50.00	83.33	67.50	54.71	32.54	50.00	69.05	73.02	51.20
Avg. Recall (%)	55.83	29.37	75.00	75.00	70.83	83.33	61.11	28.57	91.43	46.28	61.59
Avg. Time (secs)	38.24	21.20	39.29	36.79	31.65	42.62	27.05	47.71	22.89	32.85	34.23

Comparing these results with those of a similar study carried out on a WAP mobile phone interface[3] shown in table 2. We can observe that the overall performance in terms of effectiveness is better for the WAP experiment users. This is maybe the opposite of what we would expect, given that a larger display area allows more content to be read and one could argue therefore reduces some of the cognitive overhead of having to remember what was mentioned in earlier sentences that are out of view. The results for the PDA experiment are skewed by the precision of lowest performing user (user 7). One further interesting observation is that the two highest performing users were

[3] The results reporting in [20] contained errors and the values are incorrect. Instead please refer to [19] for the corrected results.

consistent in both experiments (users 4 and 10). A possible reason for this is the content of these queries may have been easier for the users to digest, the topic being either the subject of current affairs or common knowledge.

Table 2. Average precision, recall and time for the overall WAP experiment.

	User										Avg.
	1	2	3	4	5	6	7	8	9	10	
Avg. Precision (%)	46.43	44.05	25.00	75.00	35.00	54.50	41.00	66.67	48.95	71.43	55.84
Avg. Recall (%)	85.71	66.67	16.67	51.67	66.67	65.00	83.33	87.41	67.86	64.25	68.75
Avg. Time (secs)	25.03	42.14	21.43	31.57	35.57	27.70	36.76	22.64	23.82	23.71	29.02

Table 3 reports the average decision-correctness for each user for both the PDA and WAP experiments. These values maybe considered as reflecting the users ability to make correct decisions, identifying both relevant and irrelevant documents correctly. These results show that overall the differences in making correct decisions for the experiments is in fact smaller, but that performance of WAP users remains higher. A further interesting observation is that the lowest performing users in terms of precision for the PDA experiment (user 7) is actually among the users making the highest correct-decisions (user 7 correctly identified a number of irrelevant documents).

Table 3. Average decision correctness (DC) for the PDA and WAP experiments.

	User										Avg.
	1	2	3	4	5	6	7	8	9	10	
Avg. DC. PDA (%)	76.59	80.45	91.67	79.40	81.17	50.56	74.76	43.71	71.24	45.95	71.97
Avg. DC. WAP (%)	78.01	56.43	90.33	82.67	70.93	67.51	79.54	81.77	76.91	59.54	75.96

Analysing in more detail how users employed the different summary levels to make their decision, table 4 reports on the number of documents that were assessed by each user at different summary levels. In contrast to the users in the WAP experiment, the consistency among our PDA users on employing a particular length of summary is not as apparent, with the notable exception of the 7% summaries. Again, a similar pattern emerges that users tend to base their relevance decisions mainly on the shorter length of summaries (7% of the length of the document). There is however a slight increase in the use of the longer summaries and this has an impact on the total number of documents seen by users that participate in the PDA experiment.

Table 5 provides a better insight into the results reported in table 4 where the average precision for different users at different summary levels is reported. Comparing the overall values there is a slight decline in users ability to correctly identify relevant documents for the PDA experiment. This pattern is also evident in decision correctness despite higher values in terms of performance[4]. Within the values shown in table 5, the

[4] Due to constraints on paper length the full results for decision correctness are not report.

Table 4. Number of documents at the different levels of summary that users utilised to make decisions.

	User										Total PDA		Total WAP	
	1	2	3	4	5	6	7	8	9	10				
Title	9	44	13	6	17	10	29	0	18	25	171	34%	233	41%
7%	19	24	20	14	28	20	24	20	20	20	209	42%	271	48%
15%	11	5	6	12	6	9	5	12	24	0	90	18%	50	9%
30%	4	2	3	12	0	1	5	0	6	0	33	7%	16	3%
Total	43	75	42	44	51	40	63	32	68	45	503	100%	570	100%

Table 5. Avg. precision (%) for the different levels of summary.

	User										Total	Total
	1	2	3	4	5	6	7	8	9	10	PDA	WAP
Title	50.00	50.00	0.00	0.00	0.00	50.00	16.67	IND	48.75	43.75	53.19	54.10
7%	33.33	50.00	50.00	100.00	58.33	52.14	12.50	50.00	37.50	62.50	51.25	60.71
15%	50.00	100.00	0.00	50.00	50.00	0.00	100.00	33.33	50.00	IND	50.00	41.18
30%	0.00	0.00	50.00	50.00	IND	0.00	0.00	IND	100.00	IND	50.00	28.57
Total	66.67	50.00	50.00	83.33	67.50	54.71	32.54	50.00	69.05	73.02	51.20	55.84

occurrence of '0.0' precision refers to a decision that was marked as either non-relevant or a series of incorrect decision[5] and 'IND' denotes that a decision was not made using that particular level of summary. The user with the highest precision (user 4) was also amongst those users that showed the highest levels of decision correctness and the user with the lowest decision-correctness (user 8) was also among the users with the lowest precision.

Figure 2 reports another direct comparison, between the average precision for both experiments of the first and last queries presented to users. This comparison highlights the effect of fatigue in the relevance decision process. Whilst fatigue was not an experimental variable being measured it seems a likely reason for the observed drop in performance. This effect is important when comparing the results of the experiments (the first query PDA with the first query WAP). It can be noted that both sets of users perform better for precision (and recall) in the first query as opposed to the last. The effect of fatigue can also be seen in the average time taken to make the relevance decision (not shown in figure 2), users tend to be taking more time for their decision in the first queries, the time notably decreases in the last queries and this reflects on the accuracy of the relevance decision. Effects from fatigue are more apparent in the PDA experiment compare to the WAP experiment.

Another interesting comparison is reported in figure 3. This graph reports the average precision, average recall, and average time for short and long queries.

Long queries were defined as those above a median length value, and short queries defined as those below this value (less than or equal to 6 lines are considered as short). Although not highly pronounced, observations show a difference in average precision

[5] There was only one occurrence of consistently incorrect decisions that resulted in a decision-correctness of '0.0' (user 2 at 30%).

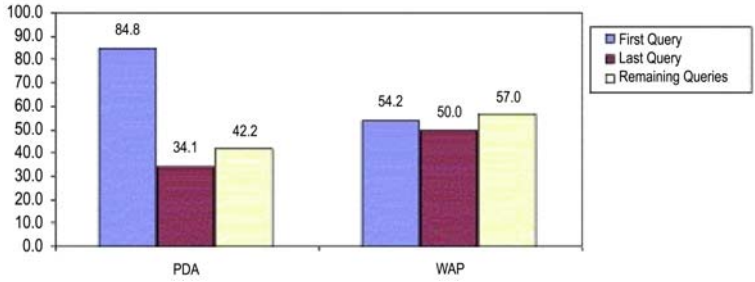


Fig. 2. Average precision for the first, last and remaining queries for the PDA and WAP experiments.

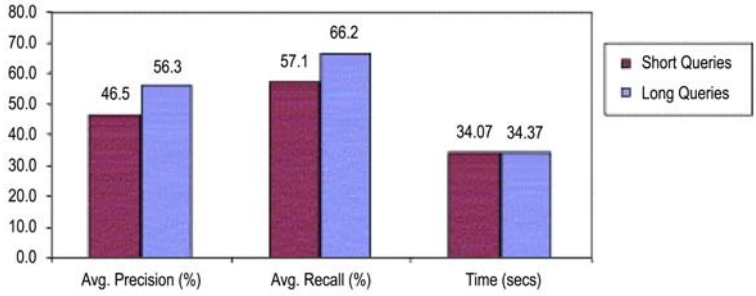


Fig. 3. Average Precision, Recall and Time for Long and Short Queries for the PDA Experiment.

recall for short and long queries. The results of this study agree with the findings of a previous experiment [21], that long queries contain more information for the relevance decision to be taken and therefore enable a user to produce higher levels of recall and precision. Another interesting finding, confirmed by the results, is that longer queries do not require longer times for the relevance decision, this can be attributed the techniques employed by the user to make the relevance assessments. This effect could be due to users employing a "Keyword Spotting" strategy to making their relevance decisions [22].

From tables 1-5 and figures 2-3, we can conclude that the presentation of documents like news articles on the small screen continues to be both feasible and relatively effective. In fact, considering only the 15% document summary length[6], the results show an improvement in levels of effectiveness compared to those found when users assess the relevance of documents on a WAP mobile phone interface [20]. In particular, the average levels of precision are similar to those found when documents summaries are

[6] To compare the results with those for other modalities only users' performance at the level of 15% summary length can be used. However, the findings of our experiment show that comparisons based on 15% summary length do not fully represent the overall performance of our PDA users since they were most effective with 7% summary lengths and actually less effective overall than users of WAP also using 7% summary lengths.

presented to a user on a computer screen [22] but have slightly lower levels of recall. The average levels of recall are similar to those found when documents summaries are presented in spoken form [21].

6 Conclusions and Future Work

In recent years there has been a large increase in the use of small screen devices. The technological advances of such devices mean that many can now be considered as information terminals, capable of supporting information access tasks normally only associated with the desktop PC. However, despite the advances in device technologies there remains difficulties in supporting user interaction on such devices. This is largely due to approaches designed for the large screen of desktop being applied directly migrated to the small screen. Conducted information retrieval tasks on small devices then proves to be difficult.

The work reported in this paper is a continuation of work assessing the effectiveness of using hierarchical query-biased summaries in the context of IR on non-traditional IR platforms. We propose the use of summaries as a means to improve interfaces for search results presentation on small screen devices. This experiment is aimed at measuring users' perception of relevance of hierarchical query-biased summaries, representing the full text of documents, viewed on a PDA device interface. The difference in users' perception of relevance relating to the judgment conditions and forms of response is compared.

Our results agree with the notion that users' perception of relevance is highly influenced by factors relating to the form of information presentation [21]. The results highlighted, for PDAs, a preference for concise summaries that are relatively brief, 7% of the document length (up to a maximum of 3 sentences) compared with other summary lengths used in the experiment. Questionnaires completed by the users suggest that hierarchical query-biased summaries are useful and assist users in making relevance judgments. The results are consistent with the findings of our previous study that found for small screen displays (WAP mobile phone interface) users showed both a preference and better performance with the shorter summary lengths (7% of the document length) [20]. Further support for presenting concise relatively brief summaries on small screen devices comes from the findings of a recent WAP usability study[7].

Limiting factors of our study include: the use of a small user sample that had similar experience of using current technologies thus representing only a small proportion of the user community, and using the TREC collection to simulate an information discovery task.

As future work, using the results we have presented as a basis for supporting the use of summarisation as a better means of representing the results of a IR search, we intend to investigate the generation and use of adaptive content-aware summarisation techniques that present content to a user on the basis of their means of access, the device being used. We envisage at that such a framework may provide better support for information access that is platform independent.

[7] Carried out my Nielsen Norman Group. WAP Usability Report, December 2000. Available at http://www.nngroup.com/reports/wap

Acknowledgements

This work is supported by the EU Commission under IST Project MIND (IST-2000-26061). More information about MIND can be found at http://www.mind-project.net/.

References

1. R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley, 1999.
2. R. Brandow, K. Mitze, and L. Rau. Automatic condensation of electronic publications by sentence selection. *Information Processing and Management*, 31(5): 675-685, 1995.
3. J. Burkhardt, H. Henn, S. Hepper, K. Rintdorff and T. Schack. *Pervasive Computing Technology and Architecture of Mobile Internet Applications*. Addison-Wesley, London, 2002.
4. O. Buyukkokten, H. Garcia-Molina, A. Paepcke, and T. Winograd, Power Browser: efficient web browsing for PDAs. *Proceedings CHI2000*, Amsterdam, 430-437, 2000.
5. H. Chen, A. Houston, R. Sewell, and B. Schatz. Internet Browsing and Searching: User evaluations of category map and concept space techniques. In:*Modern Information Retrieval*. R. Baeza-Yates, and B. Ribeiro-Neto. pp. 272–274, 1999.
6. H. Edmundson. New methods in automatic abstracting. *Communications of the ACM*, 16(2), 264–285 1969.
7. D. Harman. Relevance feedback and other query modification techniques. In: *Information retrieval: data structures & algorithms*. W. B. Frakes and R. Baeza-Yates, ed. pp. 241–263, 1992.
8. A. Jameson, R. Schfer, T. Weis, A. Berthold and T. Weyrath. Making Systems Sensitive to the User's Time and Working Memory Constraints. *Proceedings of the 4th International Conference on Intelligent User Interfaces*. Los Angeles, California. ACM Press: New York, 79–86, 1998.
9. B. Jansen, A. Spink, T. Saracevic. Real life, real users and real needs: a study and analysis of user queries on the web. *Information Processing and Management*, 36(2): 207-227, 2000.
10. M. Jones, G. Marsden, N. Mohd-Nasir and K. Boone. Improving Web Interaction on Small Displays. *Proceedings of 8th WWW Conference*, Toronto, Canada, May 1999.
11. M. Jones, G. Buchanan, N. Mohd-Nasir. Evaluation of WebTwig - a site outliner for handheld Web access. *Proceedings International Symposium on Handheld and Ubiquitous Computing*, Karlsrhue. Lecture Notes in Computer Science, vol. 1707. Springer, Berlin, pp. 343-345, 1999.
12. M. Jones, G. Buchanan, and H. Thimbleby. Sorting Out Searching on Small Screen Devices, *Proceedings of the 4th International Symposium on Mobile HCI*. (pp. 81–94): Springer, 2002.
13. G. Loudon, H. Sacher and L. Kew. Design Issues for Mobile Information Retrieval. *Proceedings of Workshop on Mobile Personal Information Retrieval* (ACM SIGIR 2002) Tampere, Finland, August 2002.
14. H. P. Luhn. The automatic creation of literature abstracts. *IBM Journal*, pp. 159-165, April 1958.
15. C. Paice. Constructing literature abstracts by computer: techniques and prospects. *Information Processing and Management*, 26(1): 171-186, 1990.
16. J. Rush, R. Salvador, and A. Zamora. Automatic abstracting and indexing. ii. Production of indicative abstracts by application of contextual inference and syntactic coherence criteria. *Journal of the American Society for Information Science*, 22(4):260-274, 1971.
17. T. Saracevic. Comparative effects of titles, abstracts and full texts on relevance judgements. *Proceedings of the American Society for Information Science*, pp. 293-299, 1969.

18. E. Schofield and G. Kubin. On Interfaces for Mobile Information Retrieval. *Proceedings of the 4th International Symposium on Human-Computer Interaction with Mobile Devices*, Pisa, Italy, September 2002.
19. S. Sweeney. Hierarchical query-biased summaries for WAP mobile phones. MSc. thesis, Department of Computer and Information Sciences, University of Strathclyde, Glasgow, UK, 2001.
20. S. Sweeney, F. Crestani, and A. Tombros. Mobile Delivery of News using Hierarchically Query-Biased Summaries. *Proceedings of ACM SAC 2002*, pp. 634–639, Madrid, Spain, March 2002.
21. A. Tombros and F. Crestani. Users's perception of relevance of spoken documents. *Journal of the American Society for Information Science*, 51(9): 929–939, 2000.
22. A. Tombros and M. Sanderson. Advantages of query biased summaries in Information Retrieval. *Proceedings of ACM SIGIR*, pp. 2-10, Melbourne, Australia, August 1998.
23. C. van Rijsbergen. *Information Retrieval*. Butterworths, London, second edition, 1979.
24. E. Voorhees. Overview of TREC 2001. em Proceedings of the 11th TREC Conference, Gaithersburg, MD, USA, November 2002.
25. J. Wise, J. Thomas, K. Pennock, D. Lantrip, M. Pottier, and A. Schur. Visualizing the non-visual: Spatial analysis and interaction with information from test documents. In: *Modern Information Retrieval*. R. Baeza-Yates and B. Ribeiro-Neto. pp. 272-274, 1999.

Towards the Wireless Ward: Evaluating a Trial of Networked PDAs in the National Health Service

Phil Turner[1], Garry Milne[1], Susan Turner[1], Manfred Kubitscheck[1], and Ian Penman[2]

[1] School of Computing, Napier University, Edinburgh, EH10 5DT, UK
{p.turner,g.milne,s.turner,m.kubitscheck}@napier.ac.uk
[2] Gastrointestinal Unit, Western General Hospital, Crewe Road, Edinburgh, UK
i.penman@ed.ac.uk

Abstract. In this paper, we describe a pilot study of the clinical use of a wireless network of personal digital assistants (PDAs). We describe how we are dealing with the concerns of the clinicians with respect to maintaining the security of patient records and the potential interference which wireless devices might cause critical medical systems. Beyond these technology-driven issues we also describe a framework based on activity theory which we will use to guide the evaluation of the PDAs.

1 Introduction

This paper aims to provide a snapshot of our work at the Gastrointestinal (GI) unit of the Western General Hospital Trust where we have initiated a pilot study of the clinical use of a wireless network of personal digital assistants (PDAs).

The use of PDAs in clinical settings is growing with anecdotal evidence that almost 50% of clinicians in the United States use a PDA in their work. We quote only two illustrative cases here. Lapinsky et al. [1] have reported the use of the infrared-enabled Palm III PDAs in a Canadian intensive care unit. Limited medical software had been pre-installed. All participants reacted favourably, irrespective of prior familiarity with the device. However, it was suggested that usability could be enhanced by improving data entry and providing drop-down menus and shortcuts. The need for wireless data transmission between staff and customised features was also highlighted. A similar trial of Palm VIIs has been conducted at the Cedars-Sinai hospital in California for wireless access to clinical information from patient records, replacing web browsers and desktop PCs. Information was also transferred between colleagues during ward rounds or at shift changes [2]. The hospital is researching the potential for closer integration between PDAs and Oracle databases. More generally, Shipman [3] reports popular uses of PDAs to include patient tracking, particularly laboratory and test results, and access to treatment protocols and educational information. In the UK, a pilot study in Glasgow of pen-based PDAs for the capture of anaesthetic clinical data suggested that the device presents a viable alternative to paper [4], while a feasibility study investigating the potential for PDAs in an Edinburgh intensive care unit [5] indicated benefits would be realised in both patient handover and the processing of

F. Crestani et al. (Eds.): Mobile and Ubiquitous Info. Access Ws 2003, LNCS 2954, pp. 202–214, 2004.

vital signs data. However, it was suggested that the benefits of mobile technology would be optimised through a combination of PDAs and larger tablet handlheld devices.

It is recognised, of course, that the concept of a personal digital assistant is necessarily at odds with the highly collective / cooperative nature of the work involved. To address this problem is, in principle, very simple namely linking the PDAs by the use of a wireless network. In practice, of course, the NHS (British National Health Service) has a number of major concerns regarding wireless networking. Firstly, it has an understandably deep reluctance in having confidential patient records broadcast across the ether. There is a partially voiced fear that unauthorised people lurking in hospital car parks could in some sense 'pick up' such transmissions and compromise patients' rights to confidentiality. The second major concern is that wireless devices on and about the wards and consulting rooms might interfere with critical medical systems. While custom and practice might witness a surgeon taking calls of her cell phone during a medical procedure, this is generally perceived to be a 'bad thing' and not to be encouraged. A third concern is what we have termed 'Lenin's argument'. Lenin famously observed that everything is connected to everything else. This is also true of the networks of the National Health service. The Western General Hospital, Edinburgh (WGH), is part of the Lothian University Hospitals NHS Trust which comprises a number of other hospitals including the Edinburgh Sick Children's NHS Trust and the Royal Infirmary of Edinburgh NHS Trust. And all of this is part of the UK-wide NHSnet. Everything is connected to everything else. This inter-connectivity is another source of anxiety for network security. A breach in security anywhere is a breach in security everywhere (or at least this is the perception / fear).

These concerns must be seen against the background of potential advantages and opportunities for the clinician, which for this pilot study are seen to be (we do not expect this list to be in any sense definitive):

1. Being able to view patient records on demand on a mobile device;
2. The voice dictation of letters, notes during consultations with patients. These notes would then ideally be automatically transcribed using a voice-to-text system.
3. The on-line ordering of medical tests.
4. The on-line viewing of medical test results. This may prove to be the 'killer application' for the clinicians in the unit. Blood test results are an essential diagnostic tool and retrieving them a major focus of a clinician's use of desktop Pcs. If these results could be made available, it is likely that using a PDA may become a *sine qua non*.
5. Email to primary carers (i.e. the patient's doctors).

We now provide a description of the context of this work.

2 The Work of the GI Unit

The Western General Hospital cares for more than 150,000 patients every year. The hospital's policy is to ensure that each patient receives the highest possible standard of care and treatment in the most appropriate environment. It provides district hospital

services for North Edinburgh and surrounding areas, including some services for the whole of Lothian, with its population of over 750,000 people. The hospital also provides specialist acute health care, locally, nationally, and internationally in specialities including Neurosciences, Oncology and Gastrointestinal Medicine.

The GI Unit is a busy department providing care and a wide range of treatments for patients from a large area of Scotland. It specialises in the investigation and management of patients with conditions involving the stomach, intestines, liver, pancreas and bowel. The unit comprises four consultants, registrars, house officers, junior and student doctors, nurses, research staff and laboratory staff. There are also four permanent secretaries and two office clerks. The physicians look after emergency admissions, carry out several patient clinics, see patients in the ward, perform specialist procedures, and interact with many other specialists in the care of these patients. The secretarial and clerical duties include typing up clinic and other patient letters and discharge summaries, result gathering and information dissemination, tracking patient notes, and making patient appointments.

A major problem in the GI unit, and all busy hospital departments, is managing the flow of information regarding patient management. Tackling this problem has been the focus of two years of work in collaboration with the GI Unit. This project aims to continue that work, with further improvements to the information system, by evaluating the usefulness and technical viability of a network of PDAs. The Unit was chosen to be the focus of the project, as it typifies a busy hospital department, and reflects the work patterns, goals and constraints regarding the information system of most similar departments within the hospital.

3 Answering the Challenges

3.1 Security

Current security methods employed by the IEEE 802.11b standard for wireless networks use the WEP (Wired Equivalent Privacy) protocol are designed to provide the same level of security as on a wired LAN. This includes the encryption of data transmitted over radio waves. Although widely used in corporate, education, healthcare and other contexts, there are still valid concerns over the vulnerabilities of wireless networks to eavesdropping and general hacking, and serious flaws have been exposed in the WEP encryption algorithms.

To enable significantly improved secure end-to-end transmission of data over a wireless network, the solution we have adopted is to overlay the 802.11b wireless network security with a further layer of security in the form of a Virtual Private Network (VPN). The VPN provides very strong authentication and encryption, using a secure tunnel end-to-end connection. IP packets are encapsulated within packets, which are encrypted before being transmitted through the secure tunnel. VPNs are based on the IPSec protocol and well-established authentication and encryption algorithms, such as AES and MD5, making them the gold standard for security.

In the proposed set up for the GI Unit, PDA client devices communicate over a wireless network, with a database server held on the existing trusted wired hospital

network. Security of data communicated over the wireless network is be achieved by the use of a VPN. A VPN router (the CISCO VPN concentrator 3005), is the connection point to the hospital network, and routes all wireless communications through a secure tunnel connection to the PDA devices. The PDA devices are equipped with the software client necessary to establish these end-to-end tunnel connections. Figure 1 is a diagram of the implementation.

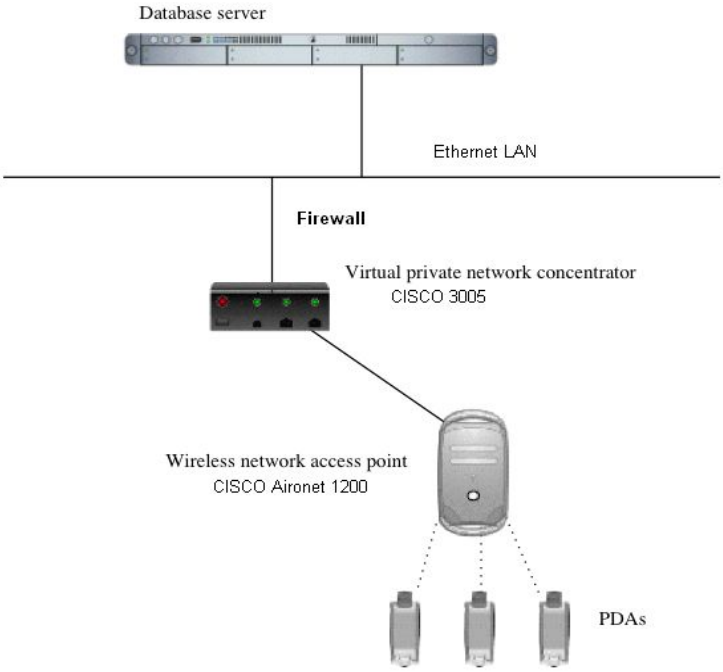


Fig. 1. A schematic of the implementation architecture

The concentrator negotiates the security parameters, authenticates users, creates and manages tunnels, encapsulates packets, transmits and receives them through the tunnel, and un-encapsulates them. Using this method, users are authenticated, and data is secured and encrypted while on the wireless network. The concentrator also provides firewall functionality to the wired network, allowing strict restrictions to be placed on which devices on the wireless network are allowed wired network access, and restrictions on the devices and services on the wired network that they can access. A further authentication stage is required in the form of a username and password to gain access to the data on the database server on the Ethernet hospital network.

This implementation satisfies all of the NHS security requirements as set out in the NHSnet SyOP document Wireless LANs in an NHSnet Environment, and the Wireless LAN's Guidelines for Implementation, Security and Safety (Rev 03 Feb 2003).

3.2 Interference

In addition to these wireless network security measures, another major concern is the potential interference which the wireless network may cause medical or other equipment in the hospital. Interference testing at the Medical Physics department of the Edinburgh Royal Infirmary is being conducted to ensure that no conflict is caused with existing wireless telemetry and monitoring systems in the hospital.

802.11b wireless devices work at a frequency in the 2.4 GHz spectrum, and it is known that other devices working on the same frequency spectrum may cause mutual interference. The field tests ensure that electromagnetic compatibility (EMC) guidelines suggested in NHS Policy Document 631 (April 2002) are adhered to, and that no interference is caused by the wireless devices, or to the wireless network by other devices. Tests are being carried out with PDAs and Access points at distances of 0.5, 1, 5 and 10 metres from any clinical, IT and telephony hardware. Care will be taken in locating base stations and any antennae at a safe distance from wiring and cabling. The cardiac unit telemetry system uses a wireless network by Symbol Technologies. The company claims this Spectrum24 system has high immunity to electronic interference.

The chosen wireless Access Points for the pilot are Cisco products which are currently deployed in other medical environments. These products use Direct Sequence Spread Spectrum radio technology (DSSS), which can be programmed to operate on select dedicated channels to reduce interference. Radio power management allows DSSS systems to be configured to work at lower power levels, which also reduces the likelihood of interference to installed medical equipment. To date, there have been no reported cases of EMC interference to medical devices from Cisco wireless LAN equipment deployed in hospitals.

4 Supporting the Opportunities

One of the reasons this pilot project came about lies with a tranche of preliminary work which two of us had been pursuing for some time (Milne & Penman). This new GI Patient System (GIPSY) has been used successfully as a working system for over a year. The GIPSY system largely replaced a paper-based system and contains patient details, clinical history, treatment records and other related material. This work began with the development of a simple standalone Access database designed to manage patient correspondence flowing between the patient's doctor and the GI unit and has now grown into an intranet-based implementation. This revised system comprises an SQL database with a layer of PHP programming to access it. As part of this work we were able to demonstrate both the practicality of accessing these data using a PDA and the restrictions of doing so without the use of a wireless network.

4.1 Choice of Mobile Device

The PDA chosen for this pilot is the HP iPAQ H5450 Pocket PC, selected because of the availability of the Pocket PC operating system and a wireless extension pack. In the first instance we have purchased 8 and have distributed them to GI unit clinicians

in advance of the trial of the wireless network proper so that they can become familiar with their operation.

4.2 The Intranet Application

A working 'proof of concept' intranet application has been created. This application, based on GYPSY, has four main functions which are:

1. Basic patient demographics (name, address, date of birth).
2. Access to existing clinic letters. These comprise the correspondence between the unit and the patient's own doctor and as such provide a clinical history.
3. Direct entry of diagnoses, test requests, drugs, follow up (i.e. "I'll see this patient again in 3 months time."). The creation of new out-patient records.
4. Access to GI guidelines. This is aimed at the junior doctors and provides clinical help and a guide to the GI unit's procedures.

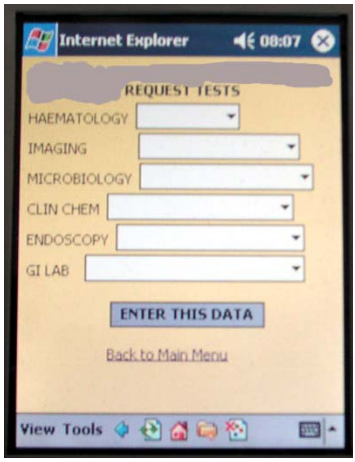


Fig. 2. A detail from a data entry screen for a patient (patient's details obscured)

Initial tests with this simple system have established its stability and a small number of real patient records have been entered into the database – see figure 2 – modified and retrieved.

5 Evaluating the PDAs in Use

The evaluation of the PDAs in use presents a non-trivial challenge. There are multiple potential foci. To take just a few examples, these include:

- issues of ergonomics such as the readability of the text on-screen;
- aspects of co-working such as the effectiveness of communication between general practitioners and hospital clinicians;
- matters arising from NHS policy, such as support for clinical governance.

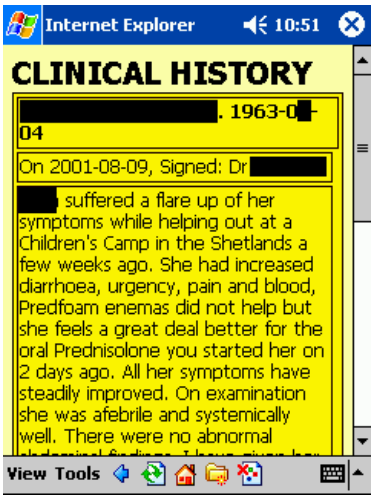


Fig. 3. Part of a clinical history screen

There are also multiple stakeholders in the process: to identify just a few, hospital clinicians, primary care practitioners, NHS IT personnel, the patients themselves, administrative staff, and the team developing and evaluating the technology. Each of these groups have their own concerns and critical success factors. Taking two examples, for clinicians, as we have already noted, better access to test results will be the core element in judging whether the initial application is worthwhile. For their colleagues in IT, concerns focus on the trouble-free co-existence of the PDA applications with other technologies, and the integrity and security of patient data. These and many other aspects need to be investigated and reported in a co-ordinated manner if the evaluation project is not to become impossibly unwieldy. Clearly some form of organising structure is required. Once that is in place the identification of specific evaluation techniques is relatively straightforward in most areas, though the evaluation of cooperative tasks still lacks proven methods.

Support for the view that evaluation in real-life practice is difficult is offered by Smithson and Hirschheim [6] in their review of information systems evaluation methods. They note the existence of significant problems in deciding what to evaluate, at what level to evaluate (e.g. macro, sector, firm, application and/or stakeholder) as well as the sheer practical difficulties of the evaluation process itself. Evaluation is, therefore, both a highly problematic and politically-sensitive task.

Smithson and Hirschheim's review groups approaches to evaluation into three 'zones' of application: efficiency of the system in question, the effectiveness of the system and an understanding of the very issues of evaluation itself (see table 1). These are seen to be moving from objective/ rational criteria to increasingly subjective / political.

The first of these, efficiency, has a strong quality and quality-control flavour about. It is the most 'objective' and quantitative of the three. Next, the zone of effectiveness

is based upon the theme of cost-benefit analysis (ranging from measures of systems usage through to user satisfaction). Finally, the zone of understanding recognises there is no one best method for evaluation for all situations and contexts. An approach aimed at understanding "…regards evaluation as problematic and seeks to understand more about evaluation in the particular organisational context".

Table 1. Categorising evaluation approaches after Smithson and Hirschheim, p. 166

Zone	(Indicative) Evaluation methods
Efficiency	Code inspection
	Software metrics
	Quality assurance …
Effectiveness	System usage
	Cost-benefit analysis
	Critical success factors
	User satisfaction …
Understanding	Context, content, process
	Social action
	Organisational behaviour
	Formative evaluation

Our own approach utilises a similar tripartite structure reinforced by a theoretical underpinning. We have demonstrated the utility of this partitioning of the problem of evaluation elsewhere [7, 8, 9]. In the first of these studies we drew on activity theory to show how the classic hierarchical structure of an activity as developed from the work of Vygoski, Leontev and Engestrom could be adopted as a conceptual structure for evaluation. Then extending these ideas we showed how such a structure could be mapped onto different forms of affordance. In essence, of course, the two sets of mappings are functionally isomorphic. Given the similarities between these two sets of mappings we will focus on the activity theoretic approach for the purposes of this discussion and demonstrate how it can be applied to the PDA evaluation.

5.1 Activities, Actions and Operations

In this section we set out the basics of one variant of activity theory, that developed by Leont'ev [10]. Unlike traditional task analysis, Leont'ev proposed the study of human activity based on an understanding of the individuals' *object,* which is usually interpreted as *objectified motive* – motive made visible or tangible. This allows us to identify uniquely a unit of analysis – the activity – by distinguishing between motivations. Activities are realised by way of an aggregation of *mediated actions*, which, in turn, are achieved by a series of low-level *operations* which are not under conscious control and hence do not require attention. This structure, however, is flexible and may change as a consequence of learning, context or both.

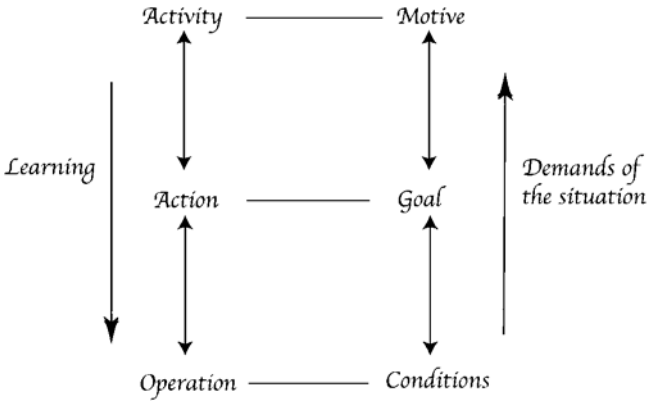


Fig. 4. An activity hierarchy

An activity, then, is the sum of the all of its constituent actions – and no more. To evaluate the actions is to evaluate the activity (at least this is a hypothesis we are happy to entertain).

By way of example, consider the process of learning to use a complex interactive device such as a PDA. The object of the activity is quite complex, probably including (among other things) the need to access and record information in a readily portable form, satisfying an interest in exploring new technology, improving the efficiency of day-to-day working life, perhaps even fulfilling a desire to be seen as someone ready to adopt new modes of working. The activity is realised by means of an aggregation of actions (e.g. setting up the device and its connection to the network, retrieving material, inputting one's schedule and so on). These individual actions in their turn are realised by a set of operations – (e.g checking relevant boxes with the stylus, handwriting items in a list). However, humans constantly learn with practice, so for instance when first presented with the handwriting recognition utility, the formation of characters recognisable by the device is the subject of conscious attention at the action level. With practice the action of writing on the PDA becomes an automatic operation. Over time the activity of using the PDA itself may be effectively demoted to that of an action – unless circumstances change. Such changes might include new procedures for communicating and recording patient data, or the acquisition of a radically upgraded device. In such circumstances consciousness becomes refocused at the level demanded by the context.

This formulation of an activity is of interest for a number of reasons: firstly, the essentially hierarchical structure, which allows us to look at different levels of task, from entering characters to the coordination of patient care. Secondly, it introduces the ideas of consciousness and motivation at the heart of the activity, supporting the identification and analysis of different activities belonging to different stakeholder groups. Finally, Leont'ev offers a mechanism by which the focus of consciousness moves up and down the hierarchy depending on the demands of the context, thus affording a consideration of changing device use over time.

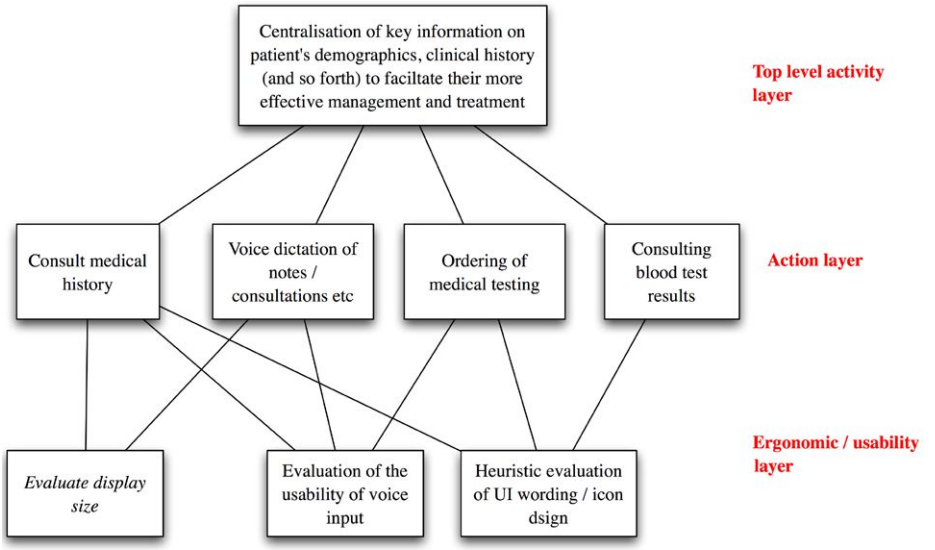


Fig. 5. An indicative hierarchical approach to PDA evaluation

There are two things of note in the above figure. Firstly, the operations layer has been re-badged the 'ergonomic /usability layer' to better reflect the nature of the evaluation. Secondly, the tasks do not neatly have a 1:1 mapping with the ergonomics layer which is not unexpected. (Figure 4 is intended to be indicative only and we expect to modify the mappings as the evaluation progresses.)

6 In Practice

Having established a theoretical and practical framework for the evaluation the next step is to map practical techniques onto each layer. Table 2 is (again) indicative of this mapping. The techniques themselves are standard in user-centre evaluation and are designed to elicit both 'snapshot' impressions and longer term experience.

Table 2. Ergonomic / usability layers

Issues	Techniques	Success factors
Physical ergonomics of input and output Stylus vs. voice input Visibility of text Comprehensibility of icons, menu labels, etc. Screen size Size and weight of device	Observation of sample tasks Heuristic evaluation Interviews	No significant usability difficulties after initial familiarisation Input and retrieval of data takes no longer than current methods. All relevant interface widgets are exploited. PDAs are carried routinely.

Table 3. Task level layer

Issues	Techniques	Success factors
Availability of specified functionality, e.g. ordering blood test.	Testing of specified functions with dummy and live data	Functionality performs as expected
Prompt retrieval of data	Interviews	Speed of retrieval is acceptable to all staff
Integrity of data input and output (including data from voice and handwriting input)	Interviews, data analysis	Degree of reliability is acceptable to all staff
Utility of device in performance of clinical and administrative tasks both single user and cooperative.	Shadowing of staff, interviews, automatic usage logs, unobtrusive user diaries	PDA is perceived by all staff to improve performance of relevant tasks. Continued usage of PDA by all staff.
Maintain security of patient data	Testing of data security	No access to any other data by unauthorised personnel

Table 4. Activity level layer

Issues	Techniques	Success factors
Enhancement of patient care Enhancement of clinical governance Demonstration of effective innovation to wider NHS, clinical, technological and academic communities	Observation, interviews, collection of statistical data.	To be established with the clinicians. Also acceptance of publications in recognised academic forums, attraction of funding.

7 Initial Evaluation Findings

As we write, we have a working wireless network of PDAs linked over the hospital network to the browser-based GIPSY system. Initially a trial group of 5 GI clinicians consisting of 2 consultants, 2 middle-grade registrars and 1 junior doctor have been issued with PDAs and spent some weeks familiarising themselves with their operation. They have been encouraged to use the PDAs for everyday tasks such as task, schedule and email organisation, note-taking and anything else they find useful. They have also been observed while interacting with the new GIPSY system with its initial basic functionality. Evaluation has begun at the ergonomic / usability-level layer, and at the task-level layer, using the group's reactions to this initial PDA usage.

At the *ergonomic/usability layer*, initial findings are very favourable, with the doctors experiencing no significant usability difficulties after the initial familiarisation period. They find using the virtual keyboard satisfactory, but slower to use than a conventional PC keyboard. This is reflected in the finding that the group enjoy being able to download their email messages to the PDA to be perused at leisure, but rarely

consider typing a reply on the PDA, preferring to do this from a PC keyboard. Most have adapted to using the handwriting recognition facility as their preferred text entry method, as it is quick and intuitive. The odd misinterpretation of a typed phrase is not seen as a big problem, as this method is mainly used for creating private notes on patients etc, for their own viewing. The majority of input for clinic detail entries, test requests and so forth is via drop-down selection boxes, which the doctors find very fast, easy to use and inherently accurate.

Fig. 6. Using the PDA

The voice dictation facility is seen as an extremely useful aid as it supersedes old, clumsy and unreliable tape machines, and the dictation is immediately available to the secretary for transcription. Interviews showed that the group find the way that text is presented on the high resolution PDA screen is clear and easy to read. All the User Interface forms on the system adapt well to the screen size and the group are happy regarding readability. The doctors find the PDA very light and convenient and easy to carry in their coat pocket, although concerns were voiced that a PDA bouncing around in a busy, fast-moving clinician's pocket might be liable to fall out due to its lightness! The junior doctor was particularly welcoming of the convenient size of the device stating that at present he is expected to carry with him at all times a large BNF (drug reference book), Guidelines manual, and Dictaphone, which could now all be replaced by the one PDA device.

At the *task-level layer,* the (initial limited) functionality performs as expected. There were many requests for the addition of access to test results, which is a functionality to be added very soon.

As regards the question of prompt retrieval of data, although the doctors noticed that the system was slightly slower than a wired network connection, this was perceived as still quite acceptable and there was no significant or irritating delay. The integrity of data on the system has been reliable, and it was remarked that data entry

via drop-down menu lists improves accuracy and legibility of data entry. The PDA functionality so far is perceived by all participants to improve performance of relevant tasks. For example, details of a patient's clinic visit including test requests can be entered by the doctor on average in 30 seconds – far faster than filling in the relevant paperwork. Requesting tests via the PDA was highlighted as being a vast improvement over the previous paper form filling method. Details were easy to enter and clear to read. Patient details such as name, Hospital Number, date of birth were automatically entered from the system, ridding doctors of the task of repeatedly writing these details on different forms.

Evaluation at the *high-level activity layer* will take place as the pilot evolves. The pilot project is still at an early stage, but initial interviews suggest that the introduction of PDAs in the hospital environment is a viable and useful proposition. Working closely with the clinician trial group, functionality will be added and improvements made as we understand more fully their mobile needs on a daily basis.

References

1. Lapinsky, S.E., Weshler, J., Sangeeta, M., Varkul, M., Hallett, D. and Stewart, T.F. (2001) Handheld computers in critical care, *Critical Care*, **5(5)**, 227-231.
2. Corman, R. (2000) Cedars-Sinai uses Palm VIIs to Access Clinical Information. A news item reported on http://www.handheldmed.com/
3. Shipman, J.P. and Morton, AC (2001) The new Black Bag, PDAs, *Health Care and Library Services*, **29(3)**, 229-237.
4. Gardner, M., Sage, M. and Gray, P. (2001) Data Capture for Clinical Anaesthesia on a Pen-based PDA: is it a Viable Alternative to Paper? In A. Blandford, J. Vanderdonckt and P. Gray (eds.) *People and Computers XV – Joint Proceedings of HCI 2001 and IHM 2001*, London: Springer. 439-456.
5. Swann, S. (2002) A feasibility study defining the potential utility of PDAs within a critical care environment. Unpublished MSc thesis, School of Computing, Napier University
6. Smithson, S. and Hirschheim. R. (1998). Analysing information systems evaluation: another look at an old problem. *European Journal of Information Systems*, **7**, 158-174.
7. Turner, P. and Turner, S. (2002) Surfacing issues using activity theory, Journal of Applied Systems Science, 3(1), 51-60.
8. Turner, P. and Turner, S. (2002) An affordance-based framework for CVE evaluation. *People and Computers XVII – The Proceedings of the Joint HCI-UPA Conference*, London: Springer, 89-104.
9. Turner, P. and McEwan, T. (2003) Activity Theory: Another Perspective on Task Analysis. In D. Diaper and N. Stanton (Eds.) *The Handbook of Task Analysis for Human-Computer Interaction*. London: Kluwer, 423-440
10. Leont'ev, A. N. (1978) *Activity, Consciousness and Personality*, (Eng. Tr. M.J. Hall) Prentice Hall Inc., Englewood Cliffs, NJ.

Aspect-Based Adaptation for Ubiquitous Software

Arturo Zambrano[1], Silvia Gordillo[1,2], and Ignacio Jaureguiberry[1]

[1] LIFIA, Universidad Nacional de La Plata
50 y 115 1er Piso
1900 La Plata, Argentina
{arturo,gordillo,jauregui}@lifia.info.unlp.edu.ar
[2] CIC, Provincia de Buenos Aires

Abstract. Information should be available everytime and everywhere in the ubiquitous computing world. Environment conditions such as bandwidth, server availability, physical resources, etc. are volatile and require sophisticated adaptive capabilities. Designing this kind of systems is a complex task, since a lot of concerns could get mixed with the application's core functionality. *Aspect-Oriented Programming* (AOP) [1] arises as a promising tool in order to design and develop ubiquitous applications, because of its ability to separate cross-cutting concerns. In this paper we propose an AOP-based architecture to decouple the several concerns that ubiquitous software comprises.

1 Introduction

An ubiquitous application should be highly adaptable, since it will be exposed to a world where runtime conditions change continuously. It must be able to face resource variability, user mobility, user's changing needs, heterogeneous networks and so on, by adapting itself as automatically as possible. As a consequence of the high number of concerns that must be modeled and the manner in which they interact, this kind of system is prone to mismatching designs.

By *adaptive capability* we mean the system's ability to adapt itself to new run-time scenarios, such capabilities which cope with specific issues (for instance: networking, system faults, etc.) should be applied in an automatic way, so that the user is not disturbed. Furthermore, *adaptive capabilities* should be incremental, that is, they should evolve in runtime, catch and store information regarding the system's context for further use.

It is desirable for the adaptive capabilities and the system's core functionality to be handled orthogonally, so that they can evolve individually and promote system's flexibility. Besides, adaptive capabilities should be isolated from each other as much as possible, in order to avoid conflicts among them and to promote the reuse of such capabilities across families of systems.

An aspect-oriented design could lead us to a better separation of concerns for self-adaptive ubiquitous applications, by isolating the different features composing them.

In this paper we present our approach to separate adaptive capabilities from the system main functionality. Section 2 and 3 present concepts related to ubiquitous computing and aspect-oriented programming. In section 4 we present our approach through

F. Crestani et al. (Eds.): Mobile and Ubiquitous Info. Access Ws 2003, LNCS 2954, pp. 215–226, 2004.

an example. The next section presents an analysis of advantages and disadvantages of this approach. After that, a comparison against an OO approach is presented. Section 7 presents implementation issues. Finally, we state our conclusions.

2 Adaptation in Ubiquitous and Mobile Computing

An ubiquitous computing system consists of a (possibly heterogeneous) set of computing devices; a set of supported tasks; and some optional infrastructure (e.g., network, GPS location service) the devices may rely on to carry out the supported tasks. [2]

Several approaches have been proposed to construct ubiquitous software artifacts. As expressed in [3] an architecture-based adaptation could be used to model adaptive systems, but in this approach most layers composing the system are aware of the existence of the others. In this way, changes in one layer could affect the others. It is desirable to use a transparent adaptation mechanism, where adapted components and components dealing with adaptation are independent.

In [11] a reflection-based approach is presented in order to modify application's behavior to adapt it to changing network conditions. It is called reflective architecture, and it allows to perform self-modifications of existing behavior. At the same time, it allows to separate system and mobility adaptation policies through a collaboration interface. We propose the use of AOP as way to enhance the independence between the system and adaptation mechanisms, and the use of the aspect-based adaptation to deal with all the concerns regarding context-awareness.

To adapt system's behaviours it is necessary to know the environment which surrounds the system. The set of properties characterizing the environment defines its *context*. A more formal definition of context is given in [4], where context is defined as: *"the reification of certain properties, describing the environment of the application and some aspects of the application itself"*. *Context* often comprises properties related to spatial and temporal positioning, networking, device constraints, user's needs and the application. A detailed study of *context* is given in [4] and [5].

Efficient execution of mobile systems requires adaptations in harmony with current context, for instance, as it is proposed in [12] *we might choose to have context feature that excludes content based on file-size, such a context feature should be active if the user is using a low bandwidth connection, but it should remain quiescent if there is a high bandwidth conneciton available*.

3 Aspect-Oriented Programming

In the application development process, it is common to find a set of concerns which are independent of any application domain and that affect many objects beyond their classes which constitute (in object-oriented programming) the natural units to define functionalities. They are called cross-cutting concerns.

A *cross-cutting concern* is a concern that is spread along most of the modules of a system. Typical cross-cutting concerns are *persistence, synchronization, error handling. etc.* As it is said in [6]: *"...existing software formalisms support separation of*

*concerns only along a **predominant dimension** neglecting other dimensions... with negative effects on re-usability, locality of changes, understandability...*". These secondary dimensions correspond to cross-cutting concerns. This idea is specially applicable to ubiquitous software, where a lot of dimensions are present.

Aspect-Oriented Programming (AOP for short) [7] is one of many technologies resulting from the effort to modularize cross-cutting concerns.

The intuitive notion of AOP comes from the idea of separating the several concerns that are present in any system. For instance, imagine a system where many *logging* operations are performed in order to track system flow control. In such a case, logging sentences are scattered along the modules of this system (e.g. `printf` for a C implementation). The *logging concern* does not have a materialization in this system, making its maintenance difficult (just imagine if it is necessary to change a parameter passed to the `printf` function, due to a change in the form that logging must be done).

The goal of AOP is to decouple those concerns, so that the system's modules can be easily maintained. AOP introduces a set of concepts:

Join Point A join point is a well-defined point in the program flow (for instance a method call, an access to a variable, etc)
Point-Cut A point-cut selects certain join points and values at those points.
Advice Advises define code that is executed when a point-cut is reached.

The program whose behaviour is affected by aspects is usually called *base program*. A join point is a concept which allows specifying points in the execution of the base program that will be affected by an aspect. One or more of these join points (from one or different classes) are identified by a point cut in the aspect layer, associating it with an advice. In this way, when one join point, defined in a point cut, is reached in the program execution, the additional code, defined in the correspondent advice is executed, adapting the original behaviour according to the current aspect. The aspect's code is composed of advises and the point-cuts where those advises must be applied. Advises could be compared to methods (in the *object-oriented* paradigm) defined within the aspects. When using aspects, the idea is to modularize cross-cutting concerns as aspects, which contain the code to handle the concerns. Since the concern is a cross-cutting one, it is necessary to apply the behaviour defined in the aspect in several places of the base program. This is done by defining the join-points and point-cuts that refer to the base program, and linking the code of the advises to the proper point-cuts.

As it will be shown in the next sections, we have used these AOP concepts to adapt the behaviour of an ubiquitous system to different runtime environments.

4 Decomposing Ubiquitous Software Using Aspects

We propose the use of AOP to separate the core functionality concern from the *context-awareness* concerns during design and implementation time, so that the corresponding software structure can evolve independently. By using AOP, the core application can be adapted in a transparent way, since it is not aware of context constraints. At the same time, the abstraction from those details makes the core application easier to design and implement. By encapsulating the adaptation mechanism and separating it from the base

application, a more reusable context representation and adaptation mechanism can be obtained.

4.1 Exemplary Application

To illustrate our approach we will use the following example:

> We must face the design of a personal assistant application for tourism. The aim of our application is to provide the user with relevant information about the place where he is in, for instance, accommodation locations, restaurants, museums, etc. Furthermore, it must report the user's current location.
>
> Implementations of the application must be able to run on a desktop computer, a laptop and PDAs, using wired or wireless connections to the servers. There might be a lot of servers which provide tourism information to the mobile client. It is supposed that a client application can connect to a different server according to the client's geographic location. Since there are different resource availability for each type of client (screen resolution, processing power, memory, etc), and there are other runtime changing issues such as bandwidth, location, etc., the whole system should be able to adapt itself to provide information in the proper way.

The natural architecture is a client-server one, where constraints associated with ubiquity make it more complex. From the client application's point of view, the designer must be conscious of:

- User's mobility: this affects the information that must be requested to the server and displayed. For instance, as the user goes on his trip, the system should report different accommodation vacancies for different cities.
- Variability of resources: the client application running on different devices is capable of using different resolutions to show graphics, variable available memory, etc.
- Variability of available bandwidth: the information should be available on time, therefore the client application should request information sized according to the connection's throughput.

We will analyze the impact of an AO design to reach a better separation of the concerns involved.

Identifying System's Concerns. The system's functionality can be summarize as *to provide the user assistance during a trip, according to some quality attributes: performance and reliability, across changing computational environments. The application relies on several servers that provide requested information*.

To cope with this general requirement, we must analyze which concerns are present. As a preliminary list of concerns of this application, we find the following:

1. System's core functionality: tourism assistant.
2. Visualization Concern: it means that information should be obtained in a format (textual, high or low resolution graphics) that can be displayed by the device.
3. Communication Concern: it means that communication should be optimized according current networking connection.
4. Memory Consumption Concern: this concern refers to the fact that requested information can be stored by the device.
5. Spatio-Temporal Concern: this concern affects the information requested since the system handles spatio-temporal positioned information.

Assuming that the object-oriented paradigm was chosen to model the application we must answer the following questions: *Which of these concerns will be modeled as aspects? Which of them as objects? How is their behaviour related?*

Most activities will be handled as requests made to the nearest server, whose results are presented to the user. It seems to be clear that the last four concerns affect the behaviour of the system's core (which is represented by the first concern), by modifying the way in which information is required. For instance:

- Spatio-Temporal Concern: affects the system by modifying its requests to reflect the current location, so that the server can return accurate information for this location. Geographic positioning can also be used to select the proper server.
- Communication Concern: this concern must deal with available connectivity and users' needs. This concern must modify requests according to current network throughput. For instance, if the user asks for a map, this concern could change the requested resolution for the map, in order to keep the network use within certain bounds.
- Visualization and Memory Consumption Concerns: these are similar to the previous case; here the concerns should modify the request in order to fit current device capabilities.

It would seem that there is a predominant dimension [6][8] where the system's core is located. Other dimensions correspond to those concerns that have some effect on the predominant one. Since these concerns modify system's behaviour for each request (see Figure 1), and they represent different topics of system's adaptation, we have decided to model them as *aspects*, leaving the core system's model as an object model. In fact, the *context model* is an object-oriented one, and the aspects (joint points, point-cuts and advises) are used as *glue* to attach the adaptive behaviour in a seamless way.

Modular Division of System's Functionality. We will focus on the client-side which has to provide pervasive features. As far as this example is concerned, the server-side is composed of a net of servers providing the information that is requested by the clients. Figure 2 depicts a simplified version of the system's architecture (client-side), where the class `Tourism Assistant` represents the base application. The *base* application's interface consists of a set of messages that obtain information from some server. The actual request should be adapted to fit current runtime constraints and user's needs, so that it is affected by the aspectual layer, which takes runtime information from the

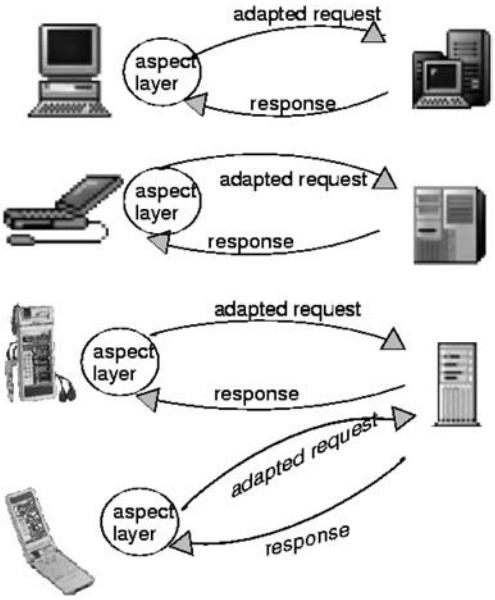


Fig. 1. The Aspectual Layer adapts client's requests

`Context` model. This is an standard object model which holds information about the current system's environment. In Figure 2 it is presented by a single class, but it is indeed a more complex representation of the reality. This model should be shared by all the aspects, so that they can *see* the same scenario.

The notation in Figure 2 has been taken from [9] with minor modification: the label `request*` indicates that the point-cut involves all the messages starting with the `request` word. Each invocation to those messages is intercepted and automatically adapted by the aspectual layer, since *requests* are defined as *point-cuts*. As it can be seen in Figure 2, the system's architecture is divided into three layers. The first layer corresponds to the base application, where no assumptions are made with respect to runtime environment constraints. The second layer is the *Aspectual Layer*, which contains the adaptive behaviour, ie. base application's behaviour is modified in a transparent way through the *point-cut* mechanism. The last layer is the context-aware one, which feeds the *aspectual layer* with runtime information.

We have analyzed how `requests` are affected by several concerns. This analysis can be extended to the remaining system's functionalities that should be adapted to the runtime scenario.

In this case, *aspects* have been used as a means of adapting the application's behaviour to the current context in runtime. They constitute a layer that provide a completely transparent instrument to obtain this adaptive behaviour. Therefore, the core application can be easily designed and implemented. Furthermore, the base application and the aspectual layer are integrated orthogonally, so that they can evolve independently.

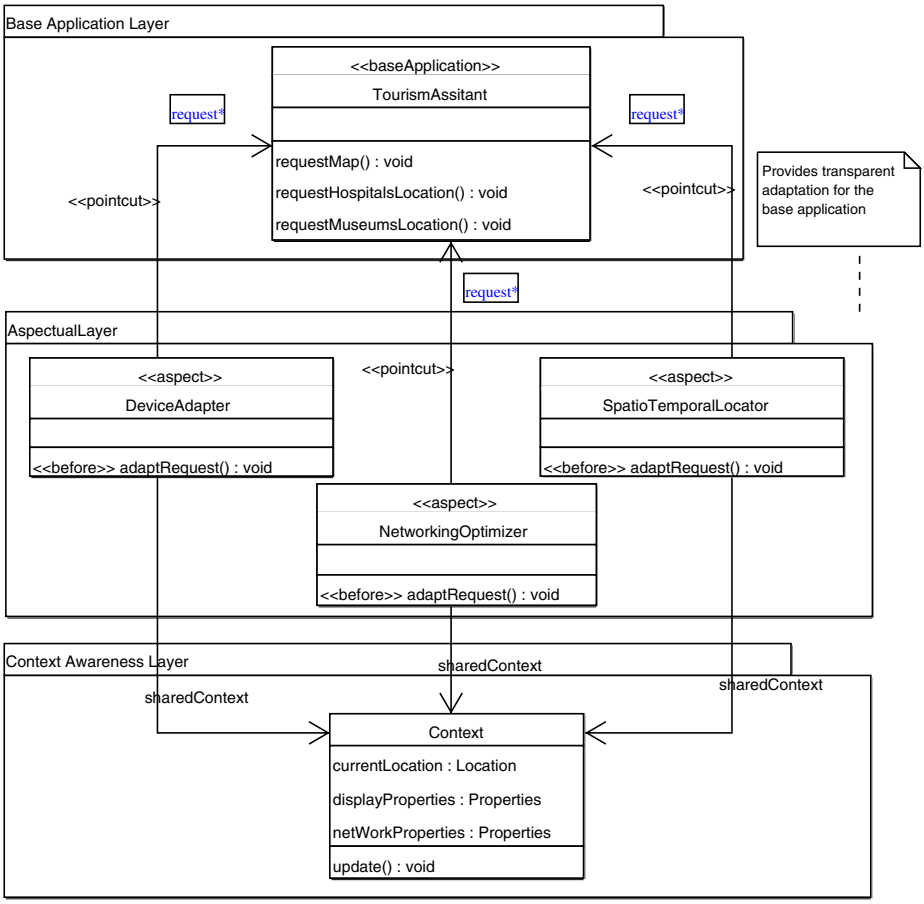


Fig. 2. Simplified Client-Side Architecture

Actual applications developed for desktop computers, laptops, handheld and PDAs may differ in implementation issues, but they can certainly follow this general schema.

Notice that this architecture corresponds to the client-side, where no data will be available at startup, instead, it will be downloaded on demand. Applications following this architecture are able to be deployed in mobile devices using current available technology, such as JVM (J2ME, SuperWaba, etc.) for mobile devices and AspectJ [10]. Since AspectJ generates pure Java code, implementations can run on any platform supporting J2ME.

5 Advantages and Drawbacks

In this section we present some advantages we have found in this approach and drawbacks that should be solved before getting a robust aspect model for ubiquitous applications. We will start by stating some advantages:

- Modifications to adapt the behaviour of base programs are included in the aspectual layer, which is invisible to them.
- Different concerns regarding ubiquity can evolve independently from one to another.
- Since the context representation is stored at the client side, the resulting application is more robust in relation to server failures.
- The separation between the core and adaptive capabilities allows us to reuse context representation and the adaptation strategies.

Some shortcomings have been found:

- Some concerns could require contradictory adaptation strategies and this could origin conflicts among them. For instance: if there is a fast network connection but a poor screen display, then the *network concern* would encourage heavy high resolution images downloads, whilst the *visualization concern* would require low resolution images download. There must be a mechanism to define which concerns take precedence or govern the others.
- In some cases, concern goals should be overridden by user defined goals, this could involve defining explicit interactions from base program toward the aspectual layer. This is not usual in the literature on aspect-orientation. Another approach could be treating user's preferences as a new *concern* modeled through aspects.

6 A Comparison against a Pure OO Approach

At this point it is interesting to discuss why a pure object-oriented is not powerful enough to correctly model this kind of systems.

Object oriented technology has proven to be a useful modelization technique. Having objects encapsulating internal state and behavior is specially useful to model abstractions in one dimension. But, as we stated in 3, abstractions belonging to several dimensions can not be easily integrated in a single unique model. Eventually, crosscutting concerns responsibilities will be spread along main dimension abstractions.

A good object oriented design for the given application might separate context awareness issues, but, in order to get applications behavior adapted by context awareness, some kind of explicit invokation will be needed. This need for an explicit invokation (the unique invokation mechanism available in object orientation) will couple at some point both, application and adaptation models. Finally, this binding between the two models will result in a dependence, making necessary to have both models in order to have a functional application. In the presented approach, the application model is completly unbound from context awareness model, allowing us to design and implement the application, forgetting non-functional constraints (such as context-awareness ones).

7 Implementation Issues

In this section we present some examples regarding how the proposed approach can be implemented using current available programming tools. For the following example

we have chosen AspectJ to provide aspects implementation. The base application can be developed using J2ME or Superwaba (a free open source Java dialect, that runs on palms and handhelds). XML has been chosen as the language used to express client's request.

For the example presented in 4.1 we will analyze the default system's behaviour and then how it is modified by aspects attending to specific concerns.

Let us analyze the request that asks the server for information about the location of the device.

```
1. public String request(){
2.     return "<REQUEST>
3.      <USER_ID>232</USER_ID>
4.      <POSITION_INFO/>
5.      </REQUEST>";
6.   }
```

Fig. 3. Base application XML request

The method `request()` in Figure 3 generates the XML request that will be sent to the server. In this example, it asks for information regarding the position of the mobile device. The expected answer can be a textual description (street names) or a map indicating the current device position in a user-friendly way. The choice between the different representations depends on the display capabilities of the device. Furthermore, in the case of a map, the picture resolution should be chosen according to the device resolution, free memory space and network bandwidth, as we discussed in 4.1.

More specifically, an aspect working on the bandwidth concern can add extra information indicating the current available bandwidth, so that the server can perform the necessary adjustments on the information to be sent as response.

A simplified implementation for this aspect is shown in Figure 4.

```
1. public aspect BandwidthAspect {
2.  String around(): call( BaseApplication.request(..)){
3.   String request = proceed();
4.    request+= "<BANDWIDTH_CONSTRAINTS>
5.     <MAX_SIZE>" + context.currentBandwidth() + "</MAXSIZE>
6.     </BANDWIDTH_CONSTRAINTS>";
7.    return request;
8.   }
9.  }
```

Fig. 4. Bandwidth aspect implementation

Notice that the bandwidth aspect defines a point-cut (Figure 4 line 2) for the method `request()` in the base application, as was expressed in 4.1. When this method is invoked, the control pass to the aspect which modifies the original XML request by

adding parameters indicating the current bandwidth. This information is obtained from the `Context` object that holds runtime information. In Figure 4 line 3 the base application method is invoked and its result is modified in the following lines.

The outcome XML request presented in Figure 5 corresponds to several aspects working on other concerns, such as memory consume, geographical location (through GPS) and image resolution, performing their adaptations.

```
1.<REQUEST>
2.  <USER_ID> 3232 </USER_ID>
3.  <POSITION_INFO> <TYPE> MAP </TYPE>
4.  </POSITION_INFO>
5.   s<CURRENT_POS>
6.            <LAT>20 20' 21"</LAT>
7.            <LONG>24 21' 0"</LONG>
8.  </CURRENT_POS>
9.  <IMAGE_CONSTRAINTS>
10.    <WIDTH>320</WIDTH>
11.    <HEIGTH>200</HEIGTH>
12.  </IMAGE_CONSTRAINTS>
13.  <BANDWIDTH_CONSTRAINTS>
14.    <MAX_SIZE> 2KB </MAX_SIZE>
15.  </BANDWIDTH_CONSTRAINTS>
16.  <MEMORY_CONSTRAINTS>
17.    <MAX_SIZE> 6MB </MAX_SIZE>
18.  </MEMORY_CONSTRAINTS>
19. </REQUEST>
```

Fig. 5. XML request after aspect's modifications

As it can be seen, some aspects might perform opposite modifications on the request. This kind of problem could arise in those base applications actions that are adapted by several aspects. For those cases where adaptation is done through requests modifications, our approach is to let all the modifications to be made, and solve the conflicts at the server side. That is to say, the server should overcome conflicting requests by stating some priority order among them. Then, there should be some strategy (at server side) that decides which modifications are the most important ones. For those base application actions that should be adapted and where several aspects are working on, it is possible to define a precedence order for the aspects (AspectJ supports such feature).

8 Conclusions

In this work we have analyzed how application behaviour can be affected and adapted by the runtime context in ubiquitous software mobile devices. Such an adaptation is necessary to optimize the use of the scarce device resources. This optimization concern

comes at a price: it can make application's development more complex. We have also addressed this problem, by providing a transparent way to modularize and decouple these optimization issues from the main application. We propose a possible decomposition of an ubiquitous system into aspects, and we analyze the consequences of the AO design.

We think that ubiquitous applications present high complexity which can be successfully targeted by the *aspect-oriented* paradigm. To conclude, we claim that aspect orientation is a fundamental tool that should be fully exploited to modularize intrinsic concerns in ubiquitous systems.

Acknowledgments

The authors thank Dr. Gustavo Rossi for his useful comments, and LIFIA for its support.

References

1. Gregor Kiczales, Erik Hilsdale, Jim Hugunin, Mik Kersten, Jeffrey Palm, William G. Griswold: Aspect Oriented Programming: Introduction. Communications of the ACM, Vol. 44. (2001) 29–32
2. D. Salber, A.K. Dey, G.D. Abowd: Ubiquitous Computing: Defining an HCI Research Agenda for an Emerging Interaction Paradigm: Tech. Report GIT-GVU-98-01. IFIP Working Conference on Engineering for Human-Computer Interaction. Georgia Tech. (1998)
3. Shang-Web Cheng, David Garlan, Bradley Schmerl, Joao Sousa, Bridget Spitznagel, Peter Steenkiste and Ningning Hu: Software Architecture-based Adaptation for Pervasive Systems. Lecture Notes in Computer Science Vol. 2299. Springer-Verlag (2002) 67–82
4. Gerti Kappell, Birgit Prll, E Kimmerstorfer, Wieland Schwinger and T.H. Hofer: Towards a Generic Customisation Model for Ubiquitous Web Applications. 2nd International Workshop on Web Oriented Software Technology in conjunction with the 16th European conference on Object-Oriented Programming ECOOP. (2002)
5. Gerti Kappell, Birgit Prll, Werner Retschitzegger and Wieland Schwinger: Customisation for Ubiquitous Web Applications. Int. Journal of Web Engineering and Technology (IJWET), Inaugural Volume, Inderscience, Volume 1, No. 1, (2003) 79–111
6. Stephan Herrmann and Mira Mezini: PIROL: A Case Study for Multidimensional Separation of Concerns in Software Engineering Environments, ACM OOPSLA 2000 Proceedings. Vol. 26, Issue 1. ACM Press New York (2001) 188–207
7. Gregor Kiczales, John Lamping, Anurag Mendhekar, Chris Maeda, Cristina Lopes, Jean-Marc Loingtier and John Irwin:Aspect-Oriented Programming. 11th Europeen Conf. Object-Oriented Programming. Lecture Notes in Computer Science, Vol. 1241. Springer-Verlag (1997) 220–242
8. Stephan Herrmann and Mira Mezini: On the Need for a Unified MDSOC Model: Experiences from Constructing a Modular Software Engineering Environment. OOPSLA 2000 Proceedings. ACM Press New York (2000)
9. R. Pawlak, L. Duchien, G. Florin, F. Legond-Aubry, L. Seinturier and L. Martelli: A UML Notation for Aspect-Oriented Software Design. Proceedings of the 1st international conference on Aspect-oriented software development. ACM Press New York (2002) 106–112
10. G. Kiczales, E. Hilsdale, J. Hugunin, M. Kersten, J. Palm, and W. G. Griswold:, An overview of AspectJ.Proceedings ECOOP 2001. Lecture Notes in Computer Science, Vol. 2072. Springer-Verlag (2001) 327–353

11. A. I. Periquet and E. Lin, "Mobility Reflection: Exploiting Mobility-Awareness in Applications by Reflecting on Distributed Object Collaborations," Technical Report 97-CSE-6, Southern Methodist University, 1997.
12. Lonsdale, P., Baber, C., Sharples, M. and Arvanitis, T. (2003) A context awareness architecture for facilitating mobile learning. In Proceedings of MLEARN 2003, London: LSDA.

Context-Aware Retrieval for Ubiquitous Computing Environments

Gareth J.F. Jones[1] and Peter J. Brown[2]

[1] School of Computing, Dublin City University
Dublin 9, Ireland
Gareth.Jones@computing.dcu.ie
[2] Department of Computer Science, University of Exeter
Exeter EX4 4PT, United Kingdom
P.J.Brown@exeter.ac.uk

Abstract. Mobile and ubiquitous computing environments provide a challenging and exciting new domain for information retrieval. One of the challenges is to provide relevant and reliable information to users often engaged in other activities or to agents acting on behalf of the user. We believe that identification of relevant information can be achieved by integration of existing methods from information retrieval and context-aware technologies. Making use of this retrieved information may be facilitated by contributions from human-computer interaction studies and agent technology to determine how and when to deliver the information to the user or how best to act on the user's behalf.

1 Introduction

A key feature of the growth in computing networks is the accompanying rapid expansion in the availability of online information. This is currently seen most obviously in the World Wide Web (WWW), which gives users the ability to access online information from anywhere in the world very rapidly. However, like networked computing itself, the current availability and utilization of online information is far from realising its full potential. The rapid developments in networked computing have the potential to make availability and exploitation of information a fundamental component of ubiquitous computing environments. Information may appear in many forms: within natural language documents in various media; from databases of facts; or aggregated from low-level sensors. Increased bandwidth and developments of mobile computing mean that it will soon be possible to access online material available from the WWW from almost anywhere on earth whatever media it originates in. Whatever source the information is contained within, wherever it is to be delivered, a fundamental issue is how to identify information relevant to an individual user, and, we argue, for ubiquitous computing how best to make use of *this* information for *this* user in their current *context*.

This paper explores issues in context-aware retrieval for mobile and ubiquitous computing. Our analysis sets out the modes of information delivery in networked model computing and the factors affecting the users of these devices and their interaction with the information and the device delivering it. We also describe our proposed methods to improve retrieval effectiveness in this environment in terms of retrieval accuracy and

F. Crestani et al. (Eds.): Mobile and Ubiquitous Info. Access Ws 2003, LNCS 2954, pp. 227–243, 2004.

user satisfaction. The primary focus of our current work is the integration of technologies from information retrieval and context-awareness to create systems for reliable and efficient delivery of information in networked context-aware environments.

At present the standard view of accessing documents from the WWW is to download them from their server onto a desktop computer. This picture is currently developing to incorporate download to networked PDAs and mobile phones. Emerging display technologies such as advanced headup displays embedded in eye glasses enable information to be displayed to a user in an augmented view of the world. In order to locate documents of interest users frequently make use of search engines such as *Google*. However, current search engines take no account of the individual user and their personal interests or their physical context. The development of personal networked mobile computing devices and environmental sensors means that personal and context information is potentially available for the retrieval process. We refer to this extension of established information retrieval (IR) as *context-aware retrieval (CAR)* [2] [13]. The objective of incorporating contextual information into the retrieval process is to attempt to deliver information most relevant to the user within their current context. As such we can see the retrieval process as embedded in a context-aware environment, and if it can be made sufficiently "intelligent", as pervasive to the user's world experience. Our interest is mainly in the incorporation of physical context data into the retrieval process. This process may involve personalization of the retrieval process in combination with context-awareness, but this need not be the case.

Retrieval to traditional visible computing devices is only one possibility: in a ubiquitous computing environment the information might be delivered to an *agent* [16] acting on behalf of the user or even on behalf of the institution that owns the environment. The ideal agent could (a) perform actions automatically if there is no need to consult the user; (b) summarise or coalesce documents before presentation; (c) decide when to deliver the retrieved information to the user, e.g. if it is marked as highly relevant, it should be delivered immediately, interrupting whatever they are currently doing; and (d) learn from users how its performance can be improved. All of this puts added demands on a retrieval engine delivering information.

The introduction of information delivery not directly controlled by the user introduces the ideas of *proactive* information retrieval, where a device may automatically initiate a request to a search engine, or may trigger information when the user enters a certain context.

Another important use of context for retrieval within ubiquitous computing is to determine the manner and timing of any information passed to the user. Sensors connected to the user and their environment can enable the user's current activities to be determined[1], thus allowing the retrieval devices to assess whether it is appropriate or safe to disturb the user at a given time. A further limitation of course is that, since the information is based on the user's context, it should be delivered in a timely fashion, since the context may change — it will often be useless to deliver information about a situation the user has just left.

[1] This is of course difficult and possible granularity may vary from determining exactly what the user is doing to merely assessing that they are busy or unoccupied.

Successful and effective CAR for ubiquitous computing environments may potentially incorporate work from a number of other established and emerging fields, including: human-computer interaction, wearables, agents and wireless networks. This paper introduces the relevant technologies from these areas, explores their integration, and outlines our current research prototype for the investigation of context-aware retrieval. The first three-quarters of the paper represents an ideas paper, building on existing work. However at Section 7 the paper changes gear, and the focus is largely on our own work in building a CAR system, and developing ideas of context-aware caching; a reader with knowledge of the field can read these later Sections independently of the rest, though it is best to read Section 4.1 too.

In detail the paper is organised as follows: Section 2 explores context and its application in context aware retrieval; Section 3 outlines the pertinence of established techniques from information retrieval and information filtering; Section 4 analyses the nature of change in context, and, in Section 4.1, introduces a piece of our relevant work; Section 5 outlines relevant methods for personalisation; Section 6 looks at the use of agents in ubiquitous retrieval application; Section 7, which starts the change of gear, describes some of our practical work that evaluates the effectiveness of these techniques, and Section 8 looks at ubiquitous and timely availability.

2 Context and Context-Aware Retrieval

A user's context consists of the their present state, their previous states (history) and their predicted future states (taken from extrapolating past states and/or from future events captured in a diary[9]); this can be enhanced by the contexts of other, similar or related, humans and other objects, or even by the context of information itself. We are primarily interested in physical elements of the user's context, although for information delivery these cannot be separated from the user's personal or cognitive context in a simple way, and we do not seek to make a sharp distinction between these in this discussion.

Mobile applications are the prime field for CAR. This is for three reasons. Firstly information is now being made available in situations it was not available in before. Secondly a mobile user is often in an unfamiliar environment and needs information about that environment. Thirdly, following on from the second point, this is an especially favourable case to use context to help select the information that is needed. Obviously in mobile applications location is a key part of the context. We believe, however, that retrieval is much more effective if the context is richer than just location, and includes fields such as temperature, objects nearby, user's current interests (and even emotional state), etc. A context used to aid retrieval can also usefully include fields that may be considered as aspects of the user model.

CAR is part of the infrastructure needed by a range of applications that detect and exploit context. Such applications are currently in their infancy, particularly if we only consider those that are products rather than research prototypes. This paper attempts to look beyond the needs of current applications, and identify the properties of CAR needed to support the potential applications of the future. We concentrate on issues of retrieval: we assume the existence of a communications infrastructure and of sensors

where we need them; we also assume there is an acceptable policy for the privacy of personal information.

Context can also be associated with each of the documents that are candidates for retrieval. Thus a document may have contextual fields representing an associated location or a temperature: an outdoor cafe, for example, is only suitable at certain temperatures. Sometimes these contextual fields are part of the explicit mark-up of a document, and sometimes they need to be derived from, for example, the textual content of the document. A central task of CAR is to match the context of the user with that of each available document.

Documents may also be associated with contextual matter of a different nature to the user's context. An example would be a contextual field that measured the authority of a document [7], in terms of the status of its author, the number of citations or links to it, its revision history, etc. This extra information, though not directly involved in the matching process, can still be used to improve the quality of the material delivered to the user; again we discuss these issues later.

2.1 Retrieval Paradigms

CAR is related to the well-established fields of information retrieval (IR) and information filtering (IF). IR and the related technology of IF are concerned with the finding of information, often in the form of text documents, which are in some sense *about* a topic that a user is interested in. Both are concerned with satisfying the user's underlying information need. The user typically expresses their information need as some form of search *request* (sometimes referred to as a *query* or as a *profile*, see later), which is then matched against the available documents. Information is conventionally retrieved from a *collection* of discrete documents. Each document may be sub-divided into *fields*. These fields may be textual, such as title, author, keywords, and the full text of the paper. Alternatively they may be of other data types which are part of the document or accompanying metadata, e.g. numbers, locations, dates, images. The retrieval task is to deliver the documents that best match the current query; each retrieved document may be accompanied with a score that gives a weighting of how well it matches. We distinguish two CAR paradigms as follows:

- *interactive*: A user initiated request is combined with the current context to derive a retrieval query, which is then applied to the document collection in the standard manner used in IR.
- *proactive*: Some or all of the documents in the document collection contain a triggering condition, and when this matches the user's current context the document is supplied to the user. This matching process may include textual fields in both the document and user search profile. This has parallels with IF; the triggering condition has the role of the profile, and the current context acts as the current document; when the current context changes, a new current document is derived and a new retrieval takes place. One difference with IF is that the triggering conditions (profiles) are specified by the provider of the document, not by the user, and apply to all users.

A crucial property of many context fields is that they are *continuous*: as the user's context changes new information may need to be retrieved. Such continuous applications normally require fast retrieval, so that the user has the illusion that new information arrives immediately there is any change in their context. This is absolutely different from the 'one-off' nature of traditional information retrieval requests, and presents many research challenges.

2.2 Retrieval Environment

Two key properties of retrieval are *precision* and *recall*. Precision is measured as the proportion of documents retrieved which are relevant, whereas recall is the proportion of the available relevant documents that have been retrieved. With context-aware applications the user is often mobile and frequently involved in other tasks. When a retrieved document is brought to their attention this is an intrusion in their activities. Rhodes and Maes [25] have observed that, for CAR, *precision is generally more important than recall*, a key observation that influences many of the ideas presented in this paper. This is especially true when the retrieval was not explicitly asked for by the user, i.e. in the proactive case. Thus, *we believe that improving precision is an increasing need for applications that deliver information ubiquitously*.

There is a further factor influencing the need for precision: there is often physically a narrower bandwidth in communicating with the user than would be the case with a conventional desktop computer. Current applications use PDAs with tiny screens or perhaps audio delivery. Even if in future products there is a larger display area, e.g. information projected onto a wall or into the user's enhanced reality, the user's attention will not be solely focussed on that information. This contrasts with a static user whose attention is often concentrated on the screen of their desktop computer.

This observation will be less true of interactive retrieval initiated directly by the user, though even here, if the user is constrained by a small screen, they cannot easily browse through reams of information. Overall a useful maxim for the design of CAR applications is: *assume each retrieval brought to the attention of the user is an intrusion; therefore try only to deliver items that are both relevant and cannot be handled automatically via some form of agent acting on the user's behalf*. Even if the information is relevant, it still needs to be presented to the user in an appropriate manner; we return to this topic in Section 6.

2.3 Context Attached to Documents

In an ideal world each document would be marked up with the context associated with it in a way that can be readily matched against the fields of the user's context. Thus if the user's context contains location, temperature and time fields, these same fields should be attached to each document to be matched, giving the location, temperature and time associated with the information in the document. For example the document for each tourist site should give its location, the temperatures at which a visit would be suitable, and the opening times. (If a field is not relevant it can be given the infinite value 'ANY'.) In reality, however, an application will need to work with documents prepared

by outside organisations, and with legacy documents: in neither case is it likely that context is attached to documents in the way the application wants.

In such cases the application needs to derive the context from the mark-up or the content of a document. If available, HTML markup of the document can be used to identify context information, e.g. the address of a restaurant; as more documents become available marked up using XML more meaningful context is likely to be easily identifiable. In addition *information extraction* techniques developed for natural language processing can be employed to identify entities and their relationships [11]. Currently information extraction techniques are often restricted to narrow domains, but development of more general and robust methods is an active area of research.

Particularly when derived automatically, the association of a document with particular context values need not be treated as exact. The association could be treated as a likelihood (e.g. from examining the text of a document, it might be inferred that there is a 60% likelihood that the document relates to a location in Canterbury, England); this likelihood can be incorporated in the overall matching score between the search topic and the document.

3 Information Selection and Delivery

Information can be selected for delivery in a number of ways. The traditional paradigms are information retrieval and information filtering. Those documents matching a search query are delivered to the user, who then inspects the documents to extract information relevant to their need. These can be extended to incorporate recommendations based on profiles derived from the user or groups of equivalent users. Such recommendation algorithms are an additional source of search topic contents, but we do not pursue them further here. The following sections define the basic features of IR and IF systems and highlight relevant differences.

3.1 Information Retrieval

Most people are now familiar with the use of IR systems in the form of web search engines. The retrieval engine responds to a search request by returning a set of potentially relevant documents to the user.

Each matching score gives a weighting of how well the document matches the query. In CAR systems these matching scores are even more important: as well as being useful for ranking, they can be used in deciding whether to deliver any documents at all. For example a proactive system may decide that, since the best matching document still has a rather low score, it is not sensible to distract the recipient with it; thus nothing is delivered. While highly desirable, it is important to note that thresholding criteria such as these are notoriously difficult to determine in current IR and IF systems [26].

Most real-world applications involve retrieval from a huge number of possible documents, and unless some optimisations are made, there will be performance problems. Thus a lot of research has been devoted to such optimisations: the basic strategy is normally to take those parts of the data which are relatively static and to preprocess these parts and place them into carefully designed data structures, so that the retrieval engine can do its matching more quickly.

3.2 Information Filtering

In IF systems the user's interests are again represented by queries which describe their information need, but here these queries are often referred to as *profiles*. IF systems are aimed at relatively stable, long-term information needs, although IF systems usually allow these interests to be modified gradually over time as conditions, goals and knowledge change. In this environment, rather than actively searching collections, users are often more passive, waiting for individual documents to be brought proactively to their attention. IF systems typically apply the same text preprocessing strategies as IR systems to improve efficiency and reliability of matching between profiles and documents. Documents with matching scores exceeding a threshold are passed to the user [26].

IF systems raise their own issues of efficiency. Many systems support thousands of simultaneous users, and thus a single document is compared in parallel to a potentially very large number of profiles. Efficiency can be achieved here by using an inverted file of the search profiles [1].

3.3 Document Structure

For both IR and IF, the simplest approach to matching of queries/profiles with documents is to treat the whole document as a single object. However, when the document is divided into distinct fields, it is straightforward to take these into account in the matching process if desired. The use of document structure is particularly pertinent to CAR where, as outlined earlier, the document structure will usually be extended to include its associated context fields, e.g. location, time, etc.

3.4 Developments in Context

The more an application knows about the user's context, the more likely it is that it can deliver documents the user wants. Increasingly sensors are available to record the user's physical context, and the values from these sensors translate into a rich array of contextual fields. The area of wearable computing accentuates this trend.

Contextual fields such as location are becoming straightforward to infer by extracting sensor information and relating this to sources such as maps. Much more difficult is the aggregation of sensor information to determine the user's current activity [10]. This is important to ensure that the presentation of information is appropriate to the user's activity and consistent with the sensed event. For example, the media of presentation can be varied according to the opportunity to devote attention to it. Aside from the theoretical development of such multi-sensor inference technologies, various practical requirements need to be taken into account. For example, context must be computed quickly and the hardware should be generally be cheap.

A current example of work in this area is the use of accelerometers to determine user activity [23]. A single accelerometer mounted in a user's jacket is used with a neural network classifier to determine whether the wearer is sitting, standing, walking or running. This information could obviously be combined with other information from the environment, e.g. information about whether the wearer is talking or listening could be

detected by a microphone mounted in their jacket. Such information can help determine whether the user should be interrupted at this time.

In addition applications need to cater for a rich variety of information sources. There are doubts [25] whether a retrieval engine should aim to cater for several *simultaneous* information sources, but there is no doubt that a retrieval engine should be able to use different sources at different times. A resource selection method might be used to determine the most appropriate source given the current context. Moreover there is the problem that the content of some sources, such as one concerned with traffic information, will be highly dynamic, and thus not amenable to those retrieval optimisations, highlighted in Section 3.1, that depend on static information that can be pre-processed in advance.

A consequence of all this richness is that retrieval will become slower. Even with current context-aware applications there are countless stories along the lines of 'the system delivered just the information the user needed about available trams, but by the time the information was delivered the best tram had departed'. IR has been immensely successful in delivering information fast, even when searching over a billion documents; CAR technologies, which involves different parameters, must be developed that do the same. This observation leads us to consider what additional features of context might be exploited to help achieve this.

4 The Nature of Context Change

For CAR we have the challenge of selecting documents with high precision in a short time from document collections that may be dynamic. To meet this challenge we need to find some retrieval advantages that apply to CAR. We believe that the most important advantage is that the current context is often changing gradually and semi-predictably. Based on this we have developed two tools [3], the Context-aware Cache, which we describe in Section 8, and the Context-of-Interest, which we describe below.

In order to capture change we have introduced a structure called a *Context Diary*, which is described in detail in [3]; the Context Diary maintains a record of previous contexts and expected future contexts.

4.1 The Context-of-Interest

In many CAR applications, particularly mobile ones, the user may often not in fact be interested in information relating to their current context: instead they are likely to be interested in a context 'just ahead'. This is an example of what we refer to from now on as the *context-of-interest*. For example the context-of-interest of a traveller or tourist might be set with the aim that they retrieve information just before they need it. The Context Diary can be used, together with the current context, to predict the context-of-interest. This predicted context-of-interest is then passed to the retrieval system in place of the true current context, with the aim of retrieving documents that are more relevant to the user's needs at the time of delivery.

As an example of how a field within the context-of-interest may be used, a Location field may be set to a point (or, more likely, a range of values) ahead of the user's current

location, taking into account their direction and rate[2] of travel, since the user is more likely to be interested in sites ahead of rather than behind them. Of course prediction can be wrong. However our experiments so far indicate that modest predictions — taking a small rather than a large leap into the future — are, on balance, winners, at least in tourist applications.

Finally the context-of-interest can be used to improve the setting of the current context. Some sensors give occasional totally wrong values, and others periodically fail to work (e.g. GPS in a tunnel). Prediction can be used for checking and to smooth over difficult periods; the result should be an improvement in the relevance of delivered documents, and an elimination of some irrelevant ones.

5 Personalisation

So far we have only considered the incorporation of environmental context into the retrieval process. Another important context component is the individual user for whom the retrieval operation is taking place. Ubiquitous applications do not *per se* distinguish between different users. Nevertheless modelling the interests of individual users or user groups (e.g. proximity services [27] and collaborative filtering [12]) is clearly an important issue for effective CAR.

There are currently many projects underway exploring methods of *personalisation* for information seeking. The general starting point is to represent the interests of a user by means of one or more keyword profiles expressing aspects of the user's interests. In a very simple approach incoming information is compared with the profiles and passed to the user if there is a sufficiently high match. This scenario is very similar to standard IF, where the profiles represent topics of interest to the user. It can be combined with IR by either using the information from the profiles to enrich a query prior to retrieval by adding additional words to it, and/or after retrieval by using the profiles to rescore retrieved documents in an attempt to bring those most of interest to the individual user nearer to the top of the list.

An important issue in personalisation for retrieval is how the interests of the user are captured. Various options are available; important ones are as follows: the user selects from a number of preset topics, the user enters sets of keywords which they believe represent topics that they are interested in, or the personalisation system monitors the user's behaviour and learns profiles from this. In all cases once initial profiles have been acquired there is scope for the system to continue to adapt over time as more information is gathered. For example, user web and email habits can be monitored and used to identify interests, which can then be clustered into themes [6] [8]. Ongoing monitoring enables changes of interest to be detected and profiles to be changed.

A practical example of ubiquitous computing which could be extended for personalised CAR is the "Shopping Jacket" [22]. Sensors in the wearer's jacket and in nearby shops communicate to inform the user if desired items are available in the shops, possibly to compare prices between competing retailers, and to attempt to entice the wearer into the shop with special offers on products that relate to their interests. This scenario could easily be extended to incorporate CAR for more general applications, matching

[2] And potentially expected rate of progress, e.g. as predicted by monitoring traffic reports.

user interest profiles against information relevant to the user's current context and activity.

6 Application of Agents for CAR

In Section 2 we noted that proactive CAR will often be an intrusion into the user's activities and in the light of this suggested that this intrusion should only take place if really necessary. One aspect of this is to seek to filter out unimportant or unreliable information, a topic to which we return in the next section; another complementary approach is to make use of automation to process information of behalf of the user so that action is taken without requiring their attention.

Applications able to fulfil this requirement are often referred to as *agents*. An agent application of this type would monitor the context and available information, compute the importance of the information and interpret it, determine the action to be taken and then act. The initial stages of this are proactive retrieval, but interpretation requires the application of *information extraction* techniques to identify data within the retrieved documents, and then some form of "intelligence" to determine the appropriate action. The issue of how an agent acquires *competence* is a critical question for agent applications: typical methods used are rule-based systems, expert systems or some form of machine learning [16].

The delegation of responsibility for the interpretation and use of information in a ubiquitous computing environment raises another key topic in agent technology, that of *trust*. The user must trust that the agent will act correctly when receiving new information. The user's trust of the agent will relate to several aspects; the user will tend to trust the agent if they were personally involved in determining its competence, e.g. by designing its rule set, or by observation if the agent performs well over time.

Note here that we are not suggesting that agents should sit between the information and the user, but rather that they should adopt the role of the *personal assistant* working alongside the user, as advocated in [16]. Thus the user is free to act themself if they wish, rather than let the agent take charge. They may, for example, ask their networked information appliance "what information is currently available which might be of interest to me?".

Agents, like human assistants, are typically good at repetitive tasks (a) for which they are directly programmed by the user, (b) which they have observed the user perform many times or (c) which other agents have informed them of.

6.1 Deciding When to Interrupt the User

Another important area for which agents can be used in ubiquitous CAR is the decision of when to deliver information to the user. Agents can monitor the user's current activities from the available sensor information, as outlined in Section 2, and use this to determine when and how to deliver information to the user. For example, if the user is currently moving forward while driving in heavy traffic it is probably not a good time to inform them of a suggested route half a mile further up the road: better to wait until later when the traffic has cleared. If the user is then stationary it may be quickest to

show the user the suggested route on a map. If the user is moving forward freely when the information needs to be delivered, showing the information on a map would be very dangerous and delivery via an audio description would be better. The concept of the context-of-interest introduced earlier is important here as well. By monitoring current activities, e.g. cars ahead of the driver moving forward, the agent can anticipate that the user, though currently stationary, will soon be occupied driving and that now is not a good time to begin delivery of information.

The area of driver disturbance from mobile phones and satellite navigation systems is currently contentious, with their role in causing accidents a high public concern. One aim of ubiquitous computing should be reduce such accidents, not to be the cause of them! Thus, we can also look to the agent in its role of personal assistant to decide whether the user needs to know this information at all. If the user typically ignores advice from the source of the information, or if a route to be suggested is one that the agent has observed the user to take before anyway, then the agent can decide not to pass it on to the user.

Again like a human assistant, the agent should only take action if it has a sufficient degree of confidence that this action will be in accordance with the user's wishes. Thus it should pass to the user information that is either novel, since the agent will not know what to do, or of unknown relevance or authority, since the agent will be uncertain as to whether an action is appropriate. (You would not want an agent to book a hotel for you, if (a) that hotel only had partial relevance to your needs, or (b) the only authority was that the hotel was recommended by its owners.)

6.2 Information Transformation

Another important role for agents in retrieval and management of information for ubiquitous computing is that of *intermediaries* that transform information as it flows from one computer to another in order to tailor it for the current circumstances [17]. For example, documents may be summarised to aid efficient delivery of key information (this may include the summarisation of multiple documents into a single summary); alternatively a document may be transformed from one media to another, e.g. a text document may be passed through a speech synthesizer for audio delivery, or a spoken document may be analysed by a recogniser to provide a text rendering.

Further examples include the possibilities of translating documents, useful for example for someone travelling in a foreign country, or of annotating documents for the individual user (an alternative form of information personalisation) [25].

6.3 Information Authority

One important factor in delegating responsibility for document and information management to an agent is the authority, importance and reliability of an information source. Some aspects of this can be inferred from the personalisation methods discussed in Section 5. For example, a source frequently acted on by a user can be regarded as reliable and important. An additional aspect in determining the action to be taken over some information relates to the *authority* conferred on it by its relationship to other available information [14].

Traditionally IR and IF techniques have focussed on the matching of document contents with query/profile expressions, and have handled documents as disjoint entities. More recently attention has been devoted to the topic of document authority. One application for these techniques is illustrated by the many broad topic queries, e.g. `find me details on PDAs`, entered by users of web search engines; often for these queries many thousands of pages can be identified as potentially relevant. Information retrieval algorithms give methods to rank the documents, but for short broad queries the resulting ranking owes more to chance word distribution statistics than meaningful selection of relevant documents. Much more useful, once a set of potentially relevant documents has been identified, is to make use of the authority assigned to each of them by users. One expression of document importance is the *conferred authority* expressed by latent human judgement of relevance indicated by the number of other documents which have hypertext links to it. This has the added feature that important documents related to a search topic can be identified even if they don't contain the search words, e.g. for the query `search engines` many relevant pages would fail to match (since search engine homepages rarely contain the words `search` and `engines`), but many documents which do discuss search engines explicitly will point to them. We can confer further authority by identifying groups of documents that point to the same pages, indicating that not only is the authoritative page pointed to by many other pages, but further that they are pointed to by pages that point to many relevant documents.

The concept of document authority may be particularly relevant for CAR where we are aiming for high precision. The authority of the document, computed either directly or inferred from its source, could be used to determine whether this information is likely to be important and/or reliable; this could be further incorporated with the matching score threshold in determining whether the document should be delivered. One example of an agent system is *Amalthaea* [19]. Amalthaea is a system for personalised news delivery; this uses a multi-level ecosystem with learning via genetic algorithms both to personalise profiles to user interests, and also to learn about the importance and reliability of individual sources in providing information of interest to a specific user.

7 Practical Investigation and Evaluation of CAR

In the last part of the paper we focus on our own practical work. In the first part of this paper, we explored a number of issues for information management in ubiquitous computing environments. In order to develop and test these ideas we have developed an experimental CAR platform. In a separate paper [3] we describe the software architecture of our CAR engine.

7.1 Experimental Testbed

Our experimental testbed consists of a set of exchangeable components. One central component is a retrieval engine: ideally this could be an existing IR engine, but, for our own purposes, given our need for experimentation, we have built our own engine. Other components are largely concerned with massaging data, for example (a) a context-of-interest component that pretends the user's context is slightly ahead of its true value

on the basis of future prediction or (b) a component to adjust scores or change the weights of fields to factor in past history. This architecture of exchangeable components, possibly including conventional IR engines, is also an aid to covering the spectrum between the two retrieval extremes of (a) conventional, entirely user driven, IR and (b) proactive retrieval with no direct user control.

7.2 Evaluation

An important aspect of research in information retrieval is the evaluation of precision and recall for the technology under investigation. We believe that this is no less true of CAR. We need to have a way of measuring precision, both to show whether our ideas really can deliver an improvement over existing IR and IF methods, and to help in tuning the various algorithms that the system uses.

One of the features of developing applications for ubiquitous computing environments is the difficulty of testing them prior to the widespread practical realisation of systems. An interesting approach to addressing this problem is introduced in [5]. Taking a graphical virtual world games engine, the authors explore ways in which a games engine can be used to simulate a user's activity in a virtual world into which monitoring of context and indeed elements of ubiquitous computing could be introduced. Such a scenario would enable the application of many different ubiquitous technologies to be explored in simulation, with the most promising investigated experimentally in the future.

7.3 Matching Algorithms

A focus of our recent work has been to find good matching algorithms that improve precision. As we have said, at the heart of a CAR system is the basic retrieval engine. At the heart of the retrieval engine is a matching algorithm that will take the user's context together with a potential document to be retrieved, and come up with a score on how well the two match. A context will usually consist of a set of different fields, and these fields will generally cover different data types. (As we have also said, if the user makes an explicit retrieval request, i.e. by specifying some search terms, it is convenient to treat this as a field of the context, along with the rest; in an extreme case it would be the only field.)

The matching algorithm typically works by computing a score for each field in turn, and then aggregating the results. (A more sophisticated algorithm might work with the fields in combination, but we will keep to the simple case here.) Matching of textual fields is a topic much studied in IR, and well-developed algorithms are available. Matching of numeric fields, such as locations given by pairs of co-ordinates, is a much more open field. An interesting approach to this problem is taken for a topic tracking application in [18]. In this work named locations are matched using a hierarchical taxonomy, for example Paris is in France, and this relationship is captured in the taxonomy. Temporal data is transformed into a standard form and overlap between topic and document recognized and rewarded. We have done some experiments following our earlier ideas in [13] and the 'fuzzy matching' techniques used by Rhodes [24], with a

focus on delivering high precision, and have made the following, albeit rather limited, conclusions:

- In practice, fields are often ranges rather than single values. This applies particularly to contexts attached to documents. For example a document about a town might apply to the area covered by that town, i.e. a range of locations, and a document about frost precautions might apply to all temperatures below 4 degrees Centigrade (here we have a potentially infinite range, as there is no lower limit specified). Ranges may also be used in the user's context, e.g. when the user is known to be somewhere within a given room, or when the user has specifically requested a range as their sphere of interest. (Ranges can also be used to represent uncertainty due to inaccurate sensors, but we prefer this uncertainty to be encompassed in the matching algorithms.)
- An ideal would be generic algorithms, one for each data type, that applied to all fields of that data type. Thus one generic algorithm would cover one-dimensional numeric fields. Such generic algorithms are indeed useful as a default, but they need to be overridden in many individual cases. For example a generic algorithm, when matching a point against a range, might give the highest score if the point were in the middle of a range, with lower scores towards the edges of the range. If, however, a field represents opening-hours of a building (a range of times), and this is matched against the current time (a point in time), the match should give a high score if the current time was near the start of the opening-hours, a less high score for the middle, and a low score if near the end.
- If two ranges are matched, the score should be higher if the ranges are roughly the same size. Thus delivering a document about the county of Devon is especially appropriate if the user has requested information for an area of about the same size – and of course if the area overlaps with Devon too.
- There must be smooth behaviour as a range gets smaller and smaller and eventually becomes a point, or when a range becomes wider and wider until it becomes infinite.
- Finally there is the negative conclusion that we have been unable to find good algorithms to aggregate individual field scores to get an overall score. Using arithmetic or geometric means has severe flaws, but more elaborate algorithms often have equally severe, though perhaps less obvious, flaws.

8 Performance and Ubiquitous Availability

Three performance challenges for CAR are: (1) delivering information of high precision; (2) delivering it fast; and (3) delivering it ubiquitously, even when the mobile user is periodically disconnected. We have discussed all these above, but will now concentrate on giving some detailed suggestions for (2) and (3).

Speed is an especial challenge. Many context-aware applications try to give the user the illusion of continuous retrieval, e.g. as a user moves round an exhibition, information on their screen continually changes to reflect nearby stands, nearby potential contacts, etc. Solutions to the speed problem have tended to assume that the content of

the information source is static, and we do that here. Indeed the case where all data is dynamic has been called 'the grand challenge' [20]. The possibilities for dynamic information repositories represented by networked mobile devices are described in [27], and handling such environments must be a long term research goal for CAR.

In traditional IR the approach to improving speed has been to build, from the content of the information source, surrogate structures that are such faster to search. Building these structures takes time, but if the information is relatively static it is time well spent.

In CAR we have proposed a surrogate structure called the *context-aware cache* [2, 3]. The context-aware cache tries to capture the documents the user is likely to need in future contexts that they are about to enter. In its current simple form it works as follows:

- The application sets a time span during which it thinks the cache will be useful; for our current data set of assuming tourist travelling in Devon, we set this to 20 minutes.
- On the basis of history and predicted future events (e.g. diary appointments in the immediate future) the application predicts what ranges each contextual field will cover in the next 20 minutes. Thus the predicted range for a location field might cover a range of a mile round the user's current point; if the user had been travelling in a fairly constant direction, this range might be biased towards locations ahead rather than locations behind.
- The application then does a retrieval where, within the user's context, the value of each field is replaced by its predicted range. The results of this retrieval are then treated as a cache. Generally the cache will be hugely smaller than the original source. (This retrieval may add a weighting factor based on past history, e.g. documents that have previously been retrieved by similar users within this context get added weight.)
- The cache is now used as the information source, rather than the original, retrieval should now be much faster.
- If the user's context strays out of its predicted range, the cache will become invalid. It will then be necessary to replace the cache or, in favourable circumstances, incrementally update it. If, contrary to our previous assumption, the content of the information source *is* dynamic the cache would need to be incrementally updated when changes occur. Strategies for incremental updating of the cache and, very importantly if rapid retrieval is paramount, for the updating of retrieval data structures, will need to be the subject of detailed further analysis and investigation.

Predicted contexts can sometimes be used independently of caches to improve performance [2]. For example if retrieval typically takes 10 seconds, then a retrieval request can use the user's predicted context in 10 seconds time.

The context-aware cache has some similarities to location-aware hoarding mechanisms [15], but the latter are concerned with explicit requests for documents, whereas our caches are concerned with anticipating the documents that are retrieved via a future query issued by the user — i.e. the user does not know the explicit documents they want. Our caches are designed to handle contexts that are much richer than just location, though location-aware hoarding mechanisms could be adapted to cover richer contexts too. Some initial results of our experiments with their use are found in [4].

We now move briefly on to issue (3): ubiquitous availability even when periodically disconnected. Caches stored on the user's personal device are, of course, a prime method of dealing with disconnected operation. The context-aware caches that we have proposed serve this purpose well. Their application ranges from short-term disconnectivity to cases in fieldwork [21] where the user is disconnected throughout a whole day, just docking with a base-station night and morning. Our implementation of context-aware cache algorithms is at an early stage, but our experiences thus far have proved useful in developing their specifications, and the issues which must be addressed in achieving these.

9 Summary

Context-aware retrieval needs to bring together a number of disparate technologies. In the case of the underlying technologies of IR and IF, CAR requires a new approach, taking features from each. Performance and method of delivery are crucial issues, and agent technology offers a means of tackling these. We have also discussed related issues in personalisation and information authority for CAR ubiquitous environments.

Towards the end of the paper, we discussed our own work. This work is built on what we hope is an apposite combination of existing IR and IF, extended to include new techniques to meet the challenges and opportunities of ubiquitous context-aware environments. These new methods include the *context-of-interest*, which seeks to deliver relevant information when the user needs it, and the *context-aware cache*. The latter addresses issues of the potentially very high number of search queries associated with rapidly and continually changing context, and also addresses problems arising from discontinuities in network connectivity.

References

1. Belkin, N.J. and Croft, W.B.: Information filtering and information retrieval: two sides of the same coin? *Communication of the ACM*, 35:29-38, 1992.
2. Brown, P.J. and Jones, G.J.F.: Context-aware retrieval: exploring a new environment for information retrieval and information filtering. *Personal and Ubiquitous Computing*, 5(4):253-263, 2001.
3. Brown, P.J. and Jones, G.J.F.: Exploiting contextual change in context-aware retrieval. In *Proceedings of the 17th ACM Symposium on Applied Computing (SAC 2002)*, Madrid, pp. 650-656, 2002.
4. Brown, P.J. and Jones, G.J.F.: Evaluation of straying outside a context-aware cache. `http://www.dcs.ex.ec.uk/$\sim$pjbrown/cgi-bin/stray\_evaluation`, 2002.
5. Bylund, M and Espinoza, F.: Testing and demonstrating context-aware services with Quake III Arena. *Communications of the ACM*, 45(1):46-48, 2002.
6. Crabtree, B. and Soltsiak, S.: Identifying and tracking changing interests. In *Proceedings of a IJCAI'97 Workshop on AI in Digital Libraries*, 1997.
7. Dourish, P., Bellotti, V., Mackay, W. and Chao-Ying Ma.: Information and context: lessons from a study of two shared information systems. In *Proceedings COOCS'93*, San Jose, pp. 42-51, Nov. 1993.

8. Fisher, M.J. and Everson R.M.: Representing interests as a hyperlinked document collection. In *Proceedings of the Twelfth International Conference on Information and Knowledge Management (CIKM'03)*, New Orleans, 2003.
9. Ford, D.A., Ruvolo, J., Edlund, S., Myllymaki, J., Kaufman, J., Jackson J. and Gerlach, M.: Tempus Fugit: a system for making semantic connections. In *Proceedings of Tenth International Conference on Information and Knowledge Management (CIKM 2001*, Atlanta, pp. 520-522, 2001.
10. Gellersen, H.W., Schmidt, A. and Beigl, M.: Multi-sensor context-awareness in mobile devices and smart artifacts. *Mobile Networks and Applications* 7:341-351, 2002.
11. Grishman, R.: Information Extraction: Techniques and Challenges. In *Information Extraction*, Pazienze, M. T. ed., pp. 10-27, 1997.
12. Gurrin, C., Smeaton, A.F., Hyowon Lee, McDonald, K., Murphy, N., O'Connor, N. and Marlow, S.: Mobile access to Fishlár-News Archive. In this volume.
13. Jones., G.J.F. and Brown, P.J.: Information access for context-aware applications. In *Proceedings of ACM SIGIR 2000*, Athens, pp. 382-4, July 2000.
14. Kleinberg, J.M.: Authoritative Sources in a Hyperlinked Environment. *Journal of the ACM*, 46(5):604-632, 1999.
15. Kubach, U. and Rothermel, K.: Exploiting Location Information for Infostation-Based Hoarding. In *Proceedings of the Seventh ACM SIGMOBILE Annual International Conference on Mobile Computing and Networking (MobiCom 2001)*, Rome, Italy, pp. 15-27, July 2001.
16. Maes, P.: Agents that Reduce Work and Information Overload. *Communications of the ACM*, 37(7), 1994.
17. Maglio, P. and Barrett, R.: Intermediaries Personalize Information Streams. *Communications of the ACM*, 43(8):96-101, 2000.
18. Makkonen J., Ahonen-Myka H., and Salmenkivi, M.: Topic Detection and Tracking with Spatio-Temporal Evidence. In *Proceedings of the 25th European Conference on IR Research (ECIR 2003)*, Pisa, Italy, pp. 251-265, 2003.
19. Moukos, A. and Maes, P.: Amalthaea: An evolving multi-agent information and discovery system for the WWW. *Autonomous Agents and Multi-Agent Systems*, 1(1):59-88, 1998.
20. Oard, D.W. and Marchionini, G.: A conceptual framework for text filtering. Report EE-TR-96-25, Univ. of Maryland, 1996.
21. Pascoe, J., Morse, D.R. and Ryan, N.S.: Developing personal technology for the field. *Personal Technologies* 2(1):28-36, 1998.
22. Randall, C. and Muller, H.: The Shopping Jacket: wearable computing for the consumer. *Personal Technologies* 4(4): 241-244, September 2000.
23. Randall, C. and Muller, H.: Context awareness by analysing accelerometer data. In *Proceedings of the Fourth International symposium on Wearable Computers*, pp. 175-176, October 2000.
24. Rhodes, B.J.: The wearable remembrance agent: a system for augmented memory. *Personal Technologies*, 1(1):218-224, 1997.
25. Rhodes, B.J. and Maes, P.: Just-in-time information retrieval agents. *IBM Systems Journal*, 39(4):685-704, 2000.
26. Robertson, S., and Walker, S.: Threshold setting in adaptive filtering. *Journal of Documentation*, 6(3):312-331, 2000.
27. Touzet, D., Weis, F. and Banâtre, M.: PERSEND: enabling continuous queries in proximate environments. In this volume.

Ubiquitous Awareness in an Academic Environment

Miguel Nussbaum[1], Roberto Aldunate[1], Farid Sfeid[1],
Sergio Oyarce[1], and Roberto Gonzalez[2]

[1] Computer Science Department, School of Engineering, Universidad Católica de Chile,
V. Mackena 4860, Santiago, Chile
{mn,farid,soyarce,raldunat}@ing.puc.cl
[2] School of Psychology, Universidad Católica de Chile,V. Mackena 4860, Santiago, Chile
rgonzale@puc.cl

Abstract. The aim of this work is to provide tools to facilitate the encounter of people that are physically close and are (usually) moving in a given setting. In this way unknown people, or people that do not know about others need or knowledge, could meet in a face to face scenario to establish a collaborative relation. The idea of Socialware is enhanced to a face to face Socialware. An Ad Hoc network was build with 40 wireless interconnected Pocket PCs (IEEE 802.11b) under a distributed agent architecture. This work studied how the proposed model behaved with freshmen engineering college students in their social and academic life.

1 Introduction

Information exchange occur when people establish some sort of trust relationship. Trust occurs in absence of time and space where power is granted in absence of information and related to a dependable relation with another person [1].

Some authors indicate that communities of people relate among them because they share a common aim, have similar needs, or are engaged by dependencies or roles [2,[3]. When one person sends a message to another, it is possible to understand it when a common language and a shared context exists [4],[5],[6]. MDs have their own language and each of the specialties have their own context (e.g., cardiology, neurology, etc.). Other authors indicate that depending on the type of relation, inter or intragroup, the probability of establishing and maintaining an interaction is determined by their similitude and their differences [7],[8]. Additionally, when two people meet, their relationship can be sporadical or one that maintains in time.

The aim of this work is to provide mechanisms to brake the social threshold and ignite a relationship among people. It is necessary to recognize and communicate the common patterns, i.e., objects and relations, between individualsthat are relevant in a given context. Context refers mainly to our history and the way we have constructed our experience [9]. What we observe from reality is what we can see from it. We will be able to share a view with others when we have a common hypothesis about the objects and relations that conform our world.

Socialware and Communityware are terms indistinctly used for supporting community work in a computer network. Groupware could fall inside this definition, but it is usually used to characterize collaborative work of already organized people. Com-

This work was partially funded by FONDECYT 1020734 and Microsoft Research

F. Crestani et al. (Eds.): Mobile and Ubiquitous Info. Access Ws 2003, LNCS 2954, pp. 244–255, 2004.

munityware relates to amorph and diverse groups [10]; it is a dynamic community where there is no fixed organization and clear aim [11]. How can people be organized in this dynamic milieu and what support is required to identify the relevant information for each of them and make it available in the adequate context? [12].

We can find in literature models of agents that facilitate the encounter of people in the Internet [13][14]. In Socialware and Communityware there is usually a fix network, and when mobility is present, there is a server that stores the user profiles and manages the users interactions [15],][16]. This model has the problem of matching people that are not necessarily close and can only communicate through the network.

Our aim is to provide tools to ease the encounter of people that have common needs, are physically close and are (usually) moving in a given setting. In this way unknown people, or people that do not know about other's needs or knowledge, could meet in a face to face scenario to establish a collaborative relation. Natural mobility of people is permitted in this model, under a distributed agent administration. These agents communicate among them to recognize common aims and obviousness of the different people in their vicinity.

In this work we propose an enhancement to the idea of Socialware, by the means of a face to face Socialware. Our objective is to study how mobile collaboration supports college student's social and academic life. In Ad Hoc networks, agents would be responsible of detecting other agents that share the profile and current needs of their owners.

Most research in Ad Hoc networks relates to technical aspects. There are some applications in education [22] and a number connected with social relations. Among these we find the Hummingbird [23] that detects the proximity of other machines displaying the name of the related person. Proem [24] defines users profiles for detecting affinity between user, similar to the work in [25] where people that answered similarly to a questionnaire where contacted by the wireless net.

2 Problem Definition

When new students arrive to campus, it takes a while for the students to build trust among them, i.e., to know each other, form groups, work and study together, etc. To enhance student relationships, we want to foster face to face contact with people that are close to each other. Pocket PCs using WI-FI (Wireless Fidelity; IEEE 802.11b) permit to form an Ad Hoc network between students that are within the network range (within 50 meters). Each student has a mobile device, where an agent on the machine supports the following functionality (Figure 1):

1. Ubiquitous Awareness.
   a. Who has? A given student that requires a specific object. For example if s/he missed a given lecture and is interested in the corresponding notes, his/her agent should search for those that went to the lecture and could trust him/her. Once somebody is found in the Ad Hoc network, we know that s/he is close, facilitating the encounter and therefore the transaction.
   b. This idea can be generalized to: What is of interest to me?, where the agent, through the students profile and current needs looks for matches with other agents, notifying when the match with the other agent is found.

Fig. 1. Main Screen

2. Constrain group configuration.

When a student wants to join a group for developing a project or going to a party, for example, the agent that stores his/her profile searches for those agents that are within the Ad Hoc network and have a similar aim and profile.

3. Communicator.
   a. Inform me when you see a specific person (that also has a mobile device). When we search for somebody on Campus, it can occur that even when we are close to a person, we do not find each other. In this case the user notifies his agent to find a specific person that once found, it is indicated to both, in order to facilitate the encounter.
   b. Send a message and inform me. When we want to say something to somebody on Campus and want to know when s/he received the message, the agent searches for the other person's agent and once the second agent receives the message, the first person is informed.

3 Modeling Interpersonal Attraction

One of the theories which has best contributed to understand interpersonal attraction is the similarity attraction hypothesis [17]. There are other factors besides similitude; attraction and attitude, as well as religious orientation [19] adhesion to traditional sexual roles [20] and preferred activities [18]. Factors that appear to be significant in the beginning of a relationship are affinity in basic values, interests and hobbies [18].

Considering the above, a study was performed in 180 first year students of Engineering and Psychology to measure their preferences evaluating the impact on the interpersonal attraction. Two questionnaires of more than 100 questions each, were given to the students at the beginning and at the end of their first semester in College. A factorial analysis was performed to identify the latent variables or subjacent constructs among the observed Intercorelations from the different measured variables. The results showed five factors of preference:

1. Shopping and or relax activities.
2. Sport activities
3. Intellectual related activities.
4. Social activities.
5. Information activities.

To identify common patterns in the different profiles, a hierarchical accumulative cluster analysis was performed. This technique, based on the quadratic Euclidean difference, as a measure of similitude, allows us to identify clusters that simultaneously present a high intra-group degree of similitude and a high inter-group degree of differentiation. Five clearly statistically significant ($p < 0{,}001$) differentiable clusters were found (Figure 2).

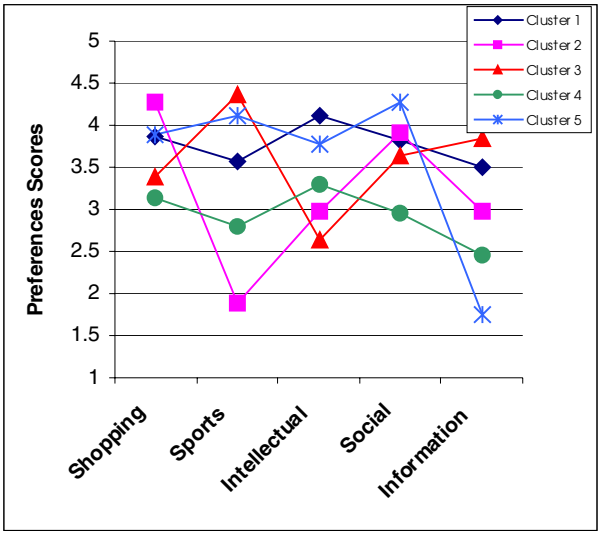


Fig. 2. Cluster Analysis

Cluster 1 represents a homogenous group with high ranking in each of the preferred categories. It identifies highly motivated students that like almost all. Cluster 2 shows a group that mainly like shopping and social activities. Cluster 3 is distinguished by people that mostly like sports, social activities and information related activities. Cluster 4 has a similar pattern to the one in cluster 1, but their difference is that their ranking in almost all is the middle of the scale. While Cluster 1 is highly engaged, Cluster 4 shows indifference. Finally Cluster 5 behaves similarly than Cluster 1, but they definitely do not like information related activities.

4 Agent Interaction

Agents don't match people together just because of their similarity, i.e., the cluster they belong. They negotiate interactions referring to their resources and needs. Resources are defined by personal information, while needs are predefined actions.

Search is structured in categories of activities that fit with the student's cluster. When a student is, for example, looking for someone to play football with, he chooses the category "Sports" and the subcategory "Football" defining the search by "play football". The agent then begins to search corresponding with all those agents he can find. The agents primary objective is to find a person who is generally interested to interact with people in reference to his cluster, scoring differently corresponding to their general interest in the activity. An agent will consider a person as interested from a certain score on and ignore it beneath this threshold.

The search sequence between different agents that are within a given vicinity, defined by the WiFi communication range, is the following one, Figure 3:

1. User (A) searches for a given (x)
2. The corresponding agent (A) sends a message indicating that user A requires x: (A,x).
3. Agents B and C, that are in the neighborhood of A receive the message (A,x).
4. B and C see if they are interested in x. If so, they send their corresponding public and private information back to A.
5. A receives the messages, if B and C send it. Calculates if there is interest with the received data and if so, sends back his public data with the corresponding private data.
6. If corresponds, B and C receives the messages and calculates with the new data if there is interest in making face to face contact. If so, sends a message to its own user and also one to A to inform the agent to send a message to user A.

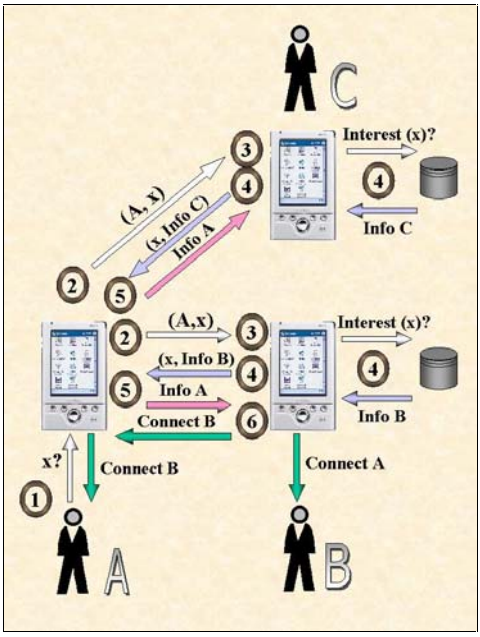


Fig. 3. Search Sequence

5 Architecture

Figure 4 illustrates the architecture of the proposed system. Four layers can be observed. At the bottom level lies the communication layer, i.e., the wireless Ethernet provided by Wi-Fi. Above the bottom level, lies the operating system, in our case Windows CE. Next, a Middleware is implemented by the means of two agents. One agent is focused to support the communication between the machines of the Ad Hoc network in a completely transparent way. The functionality provided by this agent is

to test the communication media reliability (UDP or TCP), to establish communication with other agents (machines), to hide messaging aspects (broadcast, multicast, unicast) and to maintain a list of active peers. The other agent, the profile manager, uses the services provided by the communication agent to connect with its peers in other machines. On one side it provides the services for the application, and on the other, manages the heuristics for implementing the functionality defined in Section 4. Finally the application is built on the middleware using both agents. The communication agent provides services for messaging while the profile manager agent administrates the user information and delivers the results of its heuristics.

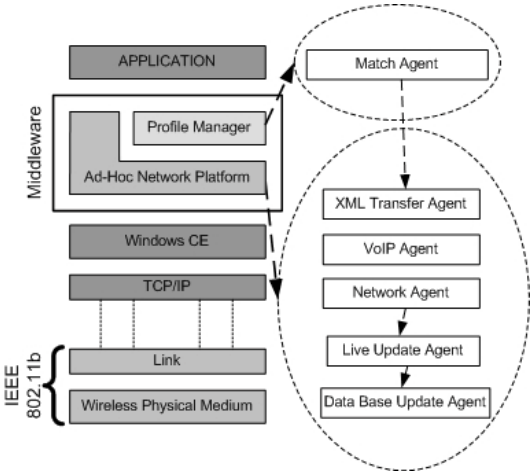


Fig. 4. Ad hoc network architecture.

The user profile is represented using XML. It provides a simple way to specify structure and allows independence of the semantics. Since it is a meta-language, oriented to describe grammars, it allows to build heuristics without knowing their implementation details before hand. Figure 5 shows an example of a user profile definition. Fig 5a graphically illustrates the user profile with a hierarchical tree. Fig 5b shows the DTD (Document Type Definition) specification, i.e., the grammar definition that allows us to implement the description of Fig 5.a. Finally Fig. 5.c, is the XML code for one instance of Fig. 5.a, using the grammar of Fig. 5.b.

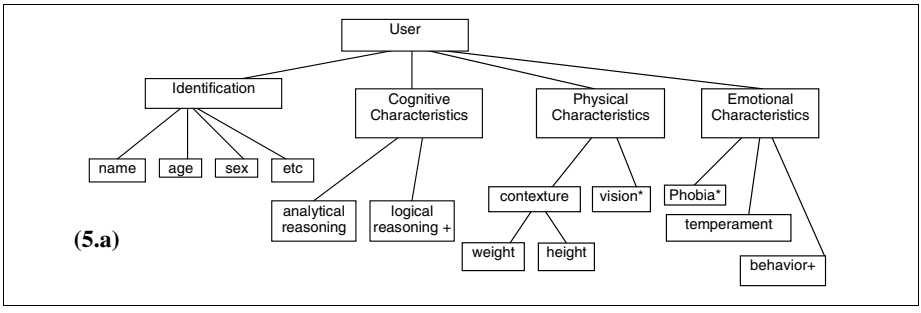


Fig. 5. Data representation: (a) hierarchical data tree

```
<?xml version="1.0" encoding="UTF-8"?>
<User UID="raldunat">
      <Identification>
         <name>Roberto Aldunate Vera</name>
         <age>34</age>
         <sex>male</sex>
         <etc>more attributes</etc>
      </Identification>
      <Cognitive>
         <analytical></analytical>
         <logical> </logical>
      </Cognitive>
      <Physical>
         <vision></vision>
         <contexture>
             <weight> 85</weight>
             <height>1.85</height>
      </Physical>
      <Emotions>
         <phobia></phobia>
         <temperament></temperament>
         <behavior></behavior>
      </Emotions>
</User>                                                    (5.b)
```

```
<!ELEMENT User (Identification, Cognitive, Physical, Emotions)>
<!ATTLIST User UID CDATA #REQUIRED>
<!ELEMENT Identification (name, age, sex, etc)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT age (#PCDATA)>
<!ELEMENT sex (#PCDATA)>
<!ELEMENT etc (#PCDATA)>
<!ELEMENT Cognitive (analytical*, logical*)>
<!ELEMENT analytical (#PCDATA)>
<!ELEMENT logical (#PCDATA)>
<!ELEMENT Physical (vision*, contextura)>
<!ELEMENT vision (#PCDATA)>
<!ELEMENT contexture (peso, estatura)>
<!ELEMENT weight (#PCDATA)>
<!ELEMENT height (#PCDATA)>
<!ELEMENT Emotions (phobia*, temperament, behavior+)>
<!ELEMENT phobia (#PCDATA)>
<!ELEMENT temperament (#PCDATA)>
<!ELEMENT behavior (#PCDATA)>                              (5.c)
```

Fig. 5. Data representation: (b) DTD associated to (a), (c) instance in XML that describes a user

Figure 6 shows a sequence diagram that summarizes the communication interactions among the agents and the applications. We can see a generic sequence of a message that flows on the network. First, a local application sends a search request of a specific type (music, pictures, profiles, etc), a unique key identifying the search, and a time interval that defines the time to end the search. Then the local search agent asks

the nearby peers if the message with the key is available by some of the peers. Finally, the network agent sends the request to the remote network agent which passes the request to the remote search engine. The petition is received by the remote application, which generates a response that travels all the way back to the local application.

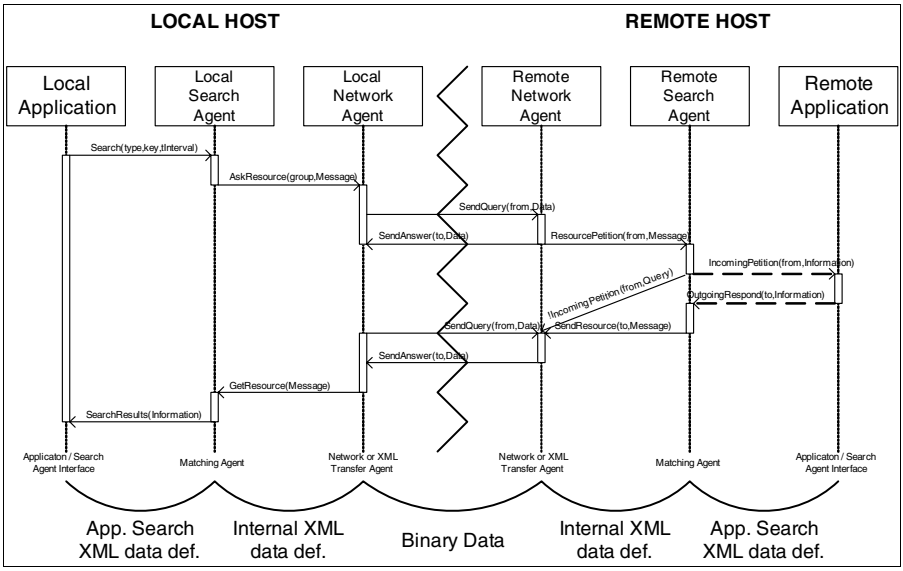


Fig. 6. Sequence Diagram of Agent Interaction

6 Results

A one semester experience started with the beginning of the Chilean academic year in March 2003, performed with university students. 40 out of 400 freshman of Engineering were randomly chosen to be part of the project. Each of them got for the semester a Wi-Fi enabled Pocket PC that was monitored by eight Wi-Fi enabled PCs, strategically located in campus. The PCs were part of the peer-to-peer network, working as gateways to the Internet. This permitted us to monitor the students machine usage while allowing them to have mobile wireless internet access.

Figure 7 illustrates the total transactions performed by the students Pocket PCs, registered by the eight PCs that monitored the networked during the semester. (It is interesting to mention that week six was an exam week). To understand this data, at the end of the experience and exhaustive questionnaire was applied to the participating students. Each of the question had to be graded from 1 to 7 (7 the best), being this the standard Chilean grading system that they were used to.

Figure 8 shows how the students rated the different tools they had available. All of them show to be below average, even the use of the Internet Tools (Explorer and Mail).

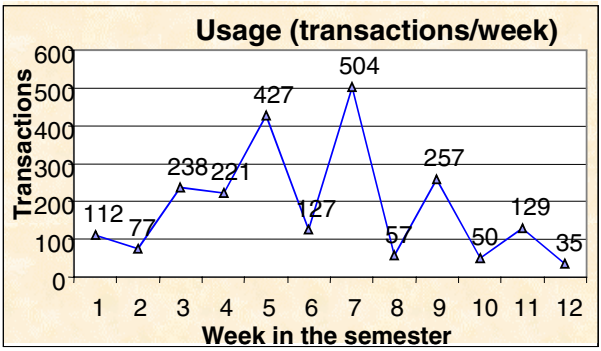


Fig. 7. Students Machine Usage during the Semester

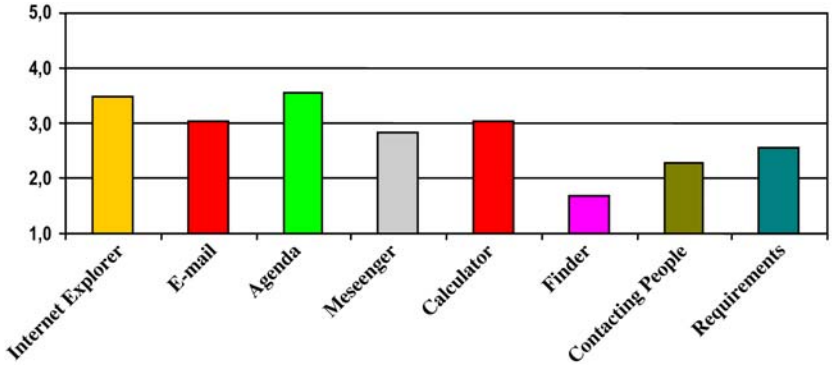


Fig. 8. Students Functionality Usage Evaluation (from 1 to 7)

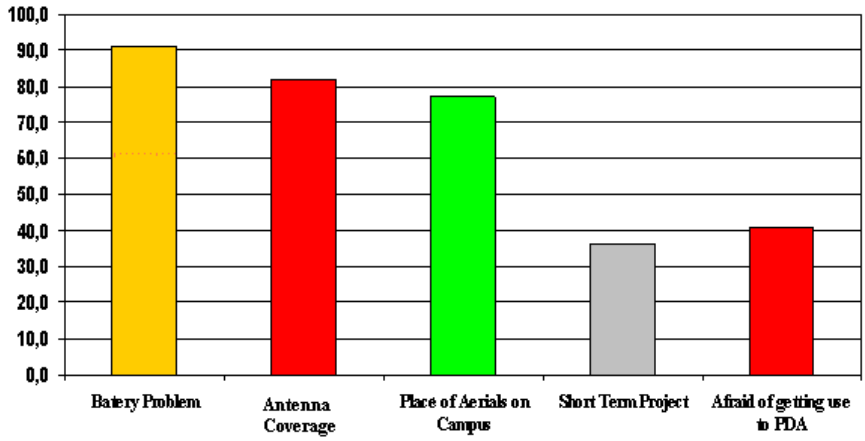


Fig. 9. Students Problems Usage Evaluation (from 1 to 7)

The reasons can be found on Figure 9 where their main claims were technological ones. Battery life is a key issue for the poor system usage. We were using Toshiba e740, that incorporates WiFi inside, which only lasted up to two hours on. For people

being more than 6 hours daily in campus having a machine available, that should be operative all the time, only less than one third of the time is a key issue. Battery life was also an issue for explaining the poor usage of the tools for supporting encounters. Since the machines have no stand by mode where the WiFi can be enabled for short periods just to monitor if somebody is requesting it, students machine intersection time is rather short (less than two hours) which makes the machine real usage time squalid. Additionally, students claimed that the network coverage was also a problem. We had only eight wireless Internet gateways available on campus, each with IEEE 802.11b limited distance coverage. This distance coverage restriction was also a problem for the peer to peer connection. It showed that in outdoors, as in an university campus, an operating range of up to 50 meters should be expected.

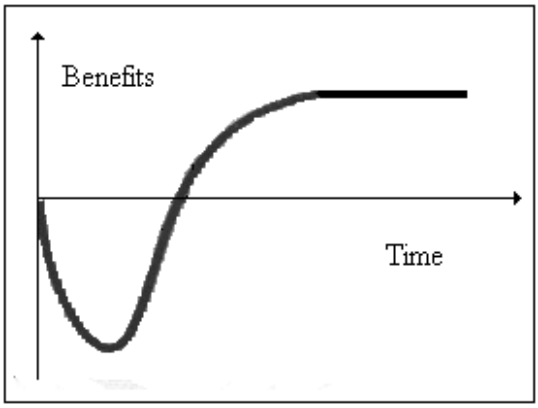


Fig. 10. Learning Curve of New Technologies [21]

Another issue the students claimed was the length of the project. We could experimentally prove the learning curve of new technologies, Figure 10, [21]. There is an initial time just after the technology introduction were productivity is increasingly negative until it reaches its worst. Then people begin to get used to it until it reaches the benefits zone, requiring some time until the users get all what they initially expected from the tool. What we saw is an initial drop out in the first month of around 25% of the users because they did not get used to the underlying mental model of the tool, plus the batteries short life time problem. Figure 6 shows that those that stood with the tool, increasingly augmented their usage after the first month until almost the second month. Then they were informed that only one month lasted for the project duration, to little for the effort still required to invest, and began aborting the project. This thesis was proved through the interviews we had with them at the end of the project.

7 Conclusions

What would we do different if we would do the project again? First, we would provide less functionality to the students in order to be productive much faster. Second,

we choose the students randomly for experimental results validity. From our experience, for testing new technology we would first select people really interested in using it and once the functionality is proved then we would assess it in a real environment. Finally, we would wait for better technology. We have not experienced with IEEE 802.11g which should improve the coverage. However the battery problem, which is a main issue, is still a big cloud in the sky.

References

1. Giddens. A. The consequences of Modernity. *Stanford University Press.* 1990.
2. Ishida T. Towards Community Ware, *PAAM'97* invited talk, 1997.
3. MacIver R. Community, *Macmillan Co*. 1917.
4. Gardner H. Perspectives of Mind and Brain. *In the disciplined mind: What all students should understand*, N.Y. pp 60-85.
5. Hetland L. Understanding Goals: Teaching the Humanities for Undersatnding in Middle School, *AERA Annual Meeting*, April 1996, NY.
6. Perkins D. What is Understanding? In Teaching for Understanding *(D. M.S. Wiske)*, San Francisco.pp 39-57.
7. Brown R.J. Group Processes. Second Edition. *Oxford: Blackwell*. 2000.
8. Sherif, M. & Sherif, C.W. Social Psycology. N.Y.: Harper & Row. 1969.
9. Mansilla V.B. Historical Understanding: Beyond the past and into the present. *Proc. Knowing, Teaching and Learning History*. Pittsburgh, November 1998.
10. Nishimura, T., Yamaki H., Komura T. & Ishida T. Community Viewer: Visualizing Community Formation on Personal Digital Assistants. *Proceedings of the 1998 ACM Symposium on Applied Computing*. 1998, pp 433 - 438.
11. Hattori F., Ohguro T., Yokoo M., Matsubara Sh & Yoshida S. SocialWare: Multiagent Systems for Supporting Network Communities. *Communications of the ACM* 42, Nro. 3, March. 1999. pp 55 - 61.
12. La Liberte, D. & Wooley., D. Presentation Features of Text-Based Conf. Systems on the WWW**.** *Comp. Mediated Com. Magazine***.** V 4, N 5. May 1997.
13. Okada, K., Maeda, F., Ichikawa, Y. & Matsushita, Y. Multiparty Videoconferencing at Virtual Social Distance: MAJIC Design. *Proceedings of CSCW'94*. 1994, pp 385 - 393.
14. Takemura, H. & Kishino, F. Cooperative Work Environment Using Virtual Workplace. *Proceedings of CSCW'92*, 1992, pp 226 - 232.
15. Matsuura, N. Fujino, G., Okada, K. & Matsushita, Y. VENUS: A Tele-Communication Environment to Support Awareness for Informal Interactions. *Proceedings 12th Scharding Int. Workshop-Design of Computer Supported Cooperative Work and Groupware Systems*. 1993.
16. Root R.W. Design of a Multi-Media Vehicle for Social Browsing. *Proceedings of the CSCW'88*. 1988. pp 25-38.
17. Byrne D. The Attraction Paradigm. *New York: Academic Press*. 1971.
18. Michinov, E., & Monteil, J. M. The similarity attraction relationships revisited: Divergence between the affective and behavioral facets of attraction. *European Journal of Social Psychology*, 2002. Nro.32., pp.485-500.
19. Kandel, D.B. Similarity in real life adolescent friendship pairs. *Journal of Personality and Social Psychology*, 1978. Nro. 65, pp. 282-292.
20. Smith, E. R., Byrne, D., Fielding, P. J. Interpersonal attraction as a function of extreme gender role adherence. *Personal Relationships*, 1995. #.2, pp.161-172.

21. Glass, Robert. The Reality of Software technologies Payoffs. Communications of the ACM February 1999, Vol. 42, No. 2.
22. Roschelle, J. Keynote paper: Unlocking the learning value of wireless mobile devices, Journal of Computer Assisted Learning (2003) 19, pp. 260-272
23. Weilenmann, A., Holmquist, L.E., (1999) Hummingbirds Go Skiing: Using Wearable Computers to Support Social Interaction., pp. 191-192 ISWC.
24. Kortuem, G., Segall, Z., Thaddeus G. Cowan Thompson (1999) Close Encounters, Supporting mobile collaboration, through Interchange of User Profiles. Lecture Notes in Computer Sc.; Vol 1707
25. Borovoy, M., Riesnick L. and Silverman T.(1998), Groupwear, Nametags that tell about a relationship. CHI 98, ACM Press, New York.

Accessing Location Data in Mobile Environments – The Nimbus Location Model

Jörg Roth

University of Hagen
Department for Computer Science
58084 Hagen, Germany
Joerg.Roth@Fernuni-hagen.de

Abstract. Location-based applications and services are getting increasingly important for mobile users. They take into account a mobile user's current location and provide a location-dependent output. Often, location-based applications still have to deal with raw location data and specific positioning systems such as GPS, which lead to inflexible designs. To support developers of location-based services, we designed the Nimbus framework, which hides specific details of positioning systems and provides uniform output containing physical as well as semantic information. In this paper, we focus on the location model, which takes into account the requirements of clients in mobile environments. A domain model contains logical links and allows the expression of semantic relations between locations. A decentralized and self-organizing runtime infrastructure offers operations to resolve the current location efficiently.

1 Introduction

Applications or services which take into account the current location will become increasingly popular in the future. Especially mobile phone providers expect a huge market for such services [23]. Typical applications answer questions like "Where is the nearest hotel?" or "Who of my friends is in proximity?". Further examples are city guides or navigation systems. Currently, the development of such services is cost-intensive due to the heterogeneity of positioning techniques, positioning systems and location data.

To support developers of location-based services we created the *Nimbus* framework. Nimbus provides a common interface to location data and hides the position capturing mechanisms. To achieve an optimal flexibility, it provides physical coordinates as well as semantic information about the current location. With Nimbus, mobile users can switch between satellite navigation systems such as GPS, positioning systems based on cell-phone infrastructures, or indoor positioning systems without affecting the location-based service. A developer can thus concentrate on the actual service function and has not to deal with positioning sensors or capturing protocols.

In this paper we present the Nimbus location model. After discussing related work we introduce the formal model. We strongly believe that a model has to consider the usage in a real scenario, thus our model supports the efficient access to location data in a network environment. In addition, Nimbus efficiently supports three-dimensional

F. Crestani et al. (Eds.): Mobile and Ubiquitous Info. Access Ws 2003, LNCS 2954, pp. 256–270, 2004.

locations with the help of a 2.5D approach. We conclude with the presentation of the underlying server infrastructure and discuss open issues.

2 Related Work

Many location-based applications and services have been developed in the last years. Tourist information systems are ideal examples for such applications. The systems CYBERGUIDE [1] and GUIDE [5] offer information to tourists, taking into account their current location. Usually, such systems come along with a general development framework, which allows a developer to create other location-aware applications. A second example for location-based applications is context-aware messaging. Such systems trigger actions according to a specific location [21]. ComMotion [12] is a system which links personal information to locations and generates events (e.g. sound or message boxes), when a user moves to a certain location. CybreMinder [6] allows the user to define conditions under which a reminder will be generated (e.g. time is "9:00" and location is "office"). Conditions are stored in a database and linked to users. Whenever a condition is fulfilled, the system generates a message box.

Several frameworks deal with location data and provide a platform for location-based application. In [11] Leonardt describes a conceptual approach to handle multi-sensor input from different positing systems. Cooltown [9] is a collection of location-aware applications, tools and development environments. As a sample application, the Cooltown museum offers a web page about a certain exhibit when a visitor is in front of it. The corresponding URLs are transported via infrared beacons. Nexus [8] introduces so-called augmented areas to formalize location information. Augmented areas represent spatially limited areas, which may contain real as well as virtual objects, where the latter can only be modified through the Nexus system. OpenLS [15] is an upcoming project and provides a high-level framework to build location-based services.

The first marketable service platforms come from the mobile phone providers. Services such as *Nightguide* or *Loco Guide* [25] serve as location-based information portals based on WAP technology. Such services reach a huge number of users, but they suffer from an insufficient location mechanism still based on the GSM cell information.

Geographic information systems (GIS) and spatial databases provide powerful mechanisms to store and retrieve location data [22]. Such systems primarily concentrate on accessing large amounts of spatial data. In our intended scenarios, however, we have to address issues such as connectivity across a network and mobility of clients, thus we have to use data distribution concepts, which are only rarely incorporated into existing GIS approaches.

3 The Nimbus Framework

Many existing frameworks either rely on a specific positioning system such as GPS or only provide a very high-level concept to integrate other positioning systems. We

designed the Nimbus framework to simplify the development of location-aware applications. Using this framework, developers can concentrate on the actual application function and can use location-dependent services of our platform. We distinguish three layers (fig. 1):

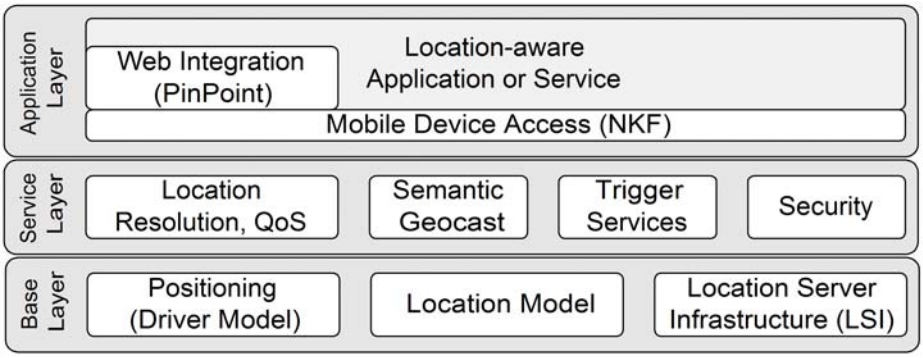


Fig. 1. The Nimbus framework

The *base layer* provides basic services related to positioning systems. The framework can use arbitrary positioning systems, ranging from satellite positioning systems, positioning with cell phone networks to indoor positioning systems, based on e.g. infrared or ultrasound. To achieve the required flexibility, we attach the positioning system via a driver interface. This interface allows the framework to switch between positioning systems at runtime. The actual focus of this paper, the *location model*, contains a formalism to describe locations and a set of rules to model the world. Finally, the *Location Server Infrastructure* (*LSI*) [19] stores the location data and provides services to access these data. It mainly consists of a federation of so-called *location servers*, each storing a piece of the entire location model.

The second layer, the *service layer,* provides higher-level location services. The most important service is called *location resolution*: an application uses this service to ask for the current location. In contrast to positioning systems, the location provided by this component contains globally unique physical as well as semantic locations. The application can specify requirements concerning precision and costs using quality of service parameters (QoS). If more than one positioning system is accessible at a certain location, the framework selects an appropriate system according to the specified parameters. An important service of this layer is the *semantic geocast* [20] which extends the original idea of geocasting by Imielinski and Navas [13]. Trigger services inform the application when a certain location was reached. A set of security functions protect the users and the framework against attacks.

The *application layer* contains the actual location-aware application or service. A communication middleware called *Network Kernel Framework* (*NKF*) [17] was especially designed for small mobile devices such as PDAs or cell phones and offers communication primitives to access the servers. To develop location-aware Web applications we offer a high-level component called *PinPoint* [18]. The World Wide Web is a powerful platform to develop location-based services, but currently makes

no use of the client's current position. PinPoint integrates location information into the HTTP data stream and still allows the usage of existing components such as Web browsers and Web servers without modifications. As an example application, we developed a Web-based tourist guide with PinPoint.

3.1 The Nimbus Location Model

The Nimbus location model contains

- a formal specification of sets which describe locations,
- a set of rules that define the relations between these sets, and
- a set of operations that process location data.

Even though we express the model independently of the later implementation, we strongly considered a decentralized storage. Especially the operations should efficiently be executed in a distributed federation of individual servers.

3.1.1 Structuring the Space with Domains and Hierarchies

The concept of semantic locations heavily influenced our model, thus we start with a brief introduction of this concept. The notion of semantic locations is not new (e.g., [11, 21]), but descriptions often tend to be very abstract. Pradhan distinguishes three types of locations [16]: *physical* locations such as GPS coordinates, *geographical* locations such as "City of Hagen" and *semantic* locations such as "Jörg's office at the university". In this paper, we do not distinguish geographical and semantic locations, but regard any location other than a physical one as a semantic location. In simplified terms: physical locations can be expressed by numbers, semantic locations by names.

Semantic locations are an ideal tool for a number of applications, sometimes in combination with physical locations. They have some important advantages:

- Semantic locations have a meaning to the user; in contrast, physical locations usually have no meaning at all to most peoples.
- Semantic locations can easily be used as a search key for traditional databases, tables or lists. In contrast, to look up physical locations, we need spatial databases with the ability to deal with geometric objects such as polygons.

In this section, we want to describe the concept of semantic locations more precisely. We especially want to relate semantic to physical locations. Let P denote the set of all physical locations. We call each coherent area $S \subseteq P$ a *semantic location* of P. We further call each set $C \subseteq 2^P$ of semantic locations, a *semantic coordinate system* of P. (2^P denotes the power set of P.) Note that we do not assume two semantic locations to be generally disjoint. A reasonable semantic coordinate system C contains semantic locations S with certain meanings, e.g.

- locations with a political meaning: countries, states, cities, districts;
- geographical locations: continents, oceans, mountains, rivers, lakes, forests;
- mobile locations: trains, planes, cars;
- temporary locations: construction zones, fairs;
- other locations: campus, malls, city centres.

We further introduce a *name* for a semantic location. Let N be the set of all possible names. We define a function $NAME$: $C \rightarrow N$, which maps a semantic location to a string. We require names to be unique, i.e. $NAME(c_1) \neq NAME(c_2)$ for $c_1 \neq c_2$. We call a semantic location with its corresponding name a *domain*. For a domain d, $d.name$ denotes the domain name, $d.c$ the semantic location.

In principle, a semantic coordinate system C could be an arbitrary subset of 2^P that contains coherent areas. Looking at real-world scenarios, however, we usually find hierarchical structures, e.g., a room is inside a building, a building is in a city, a city is in a country etc. Thus, we divide C in so-called *hierarchies*. A hierarchy contains domains with a similar meaning, e.g., domains of cities or domains of geographical items. Each hierarchy has a *root domain* and a number of *subdomains*; each of them can in turn be divided into subdomains. We call a top node of a subhierarchy a *master* of the corresponding subdomains. We denote $m \rhd s$ for master m of subdomain s. Further $\succ$ denotes the reflexive and transitive closure of $\rhd$, i.e. $d_1 \succ d_2$ if either $d_1 = d_2$ or d_1 is a top node of a subtree which contains d_2.

We call a link between a subdomain and its master a *relation*. Relations carry information about containment of domains. Hierarchies are built according to three rules:

- The area of a subdomain has to be completely inside the area of its master, i.e. if $d_1 \rhd d_2$ then $d_2.c \subset d_1.c$.
- The name of a subdomain d_2 extends the name of its master d_1 according to the rule $d_2.name$=<extension> + '.' + $d_1.name$, where <extension> can be an arbitrary string containing only letters and digits. With the help of this rule, we can effectively check if $d_1 \succ d_2$ or $d_1 \rhd d_2$ with the help of the names.
- Root domain names of two hierarchies must be different.

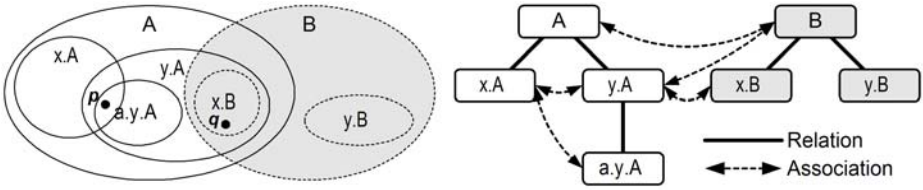


Fig. 2. Sample hierarchies

Fig. 2 shows an example with two hierarchies. Even though relations are directed, we use undirected lines in the figures as the directions are obvious.

3.1.2 Associations

In principle, the model is now expressive enough to specify realistic sets of semantic locations and their relationship among each other. One important question could be: "Given a physical location p, which semantic locations contain p?" E.g., in fig. 2 point p resides in the domains `A`, `x.A`, `y.A` and `a.y.A`. As a master fully encloses a subdomain, the results `A` and `y.A` do not carry useful information. A useful answer would be `x.A` and `a.y.A`.

This so-called *semantic resolution* could be performed by browsing through all hierarchies from the root down to the smallest domains covering p. This however would cause a large number of requests and in a real infrastructure a considerable amount of network traffic. Therefore, we introduce a second relationship between domains, the *association*:

Two domains d_1, d_2 are associated, denoted $d_1 \sim d_2$, iff

- they share an area, i.e. $d_1.c \cap d_2.c \neq \{\}$ (*condition 1*)
- and neither $d_1 \succ d_2$ nor $d_2 \succ d_1$. (*condition 2*)

Condition 2 prevents the superfluous linking of masters to their subdomains as they always share an area. Associated domains can be in different hierarchies or in the same hierarchy (see fig. 2). Using associations, we only need one domain d_0 that contains the position p. All domains $d \sim d_0$ are candidates to additionally contain p. In turn, no more domains have to be checked, thus we can avoid the time-consuming search through all hierarchies.

We can however reduce the number of candidates even more, because we are only interested in the most specific domains. If in the example above we want to know which domains contain the point q, we are only interested in the domains `y.A` and `x.B`, and not in `A` or `B`. Taking this into account, we can modify condition 1 as follows: associations only link two domains, if the shared area is not fully covered by their respective subdomains, i.e.

- $\left(d_1.c \setminus \bigcup_{e_1 \in C, d_1 \rhd e_1} e_1\right) \cap \left(d_2.c \setminus \bigcup_{e_2 \in C, d_2 \rhd e_2} e_2\right) \neq \{\}$. (*condition 3*)

Note that if condition 3 is true, condition 2 is true as well, thus we can use condition 3 as definition for associations. We introduce the abbreviation

$$\Delta(d) = \left(d.c \setminus \bigcup_{e \in C, d \rhd e} e\right)$$

for the domain's area without the subdomains' area and finally get the short definition

$d_1 \sim d_2$ iff $\Delta(d_1) \cap \Delta(d_2) \neq \{\}$.

In fig. 2, the shared area of `A` and `x.B` is fully covered by the domain `y.A`, thus `A` and `x.B` are not associated as this link would not carry additional information. Starting at `x.B` we only have to check `y.A`. Note that condition 3 does not always reduce the number of queries. E.g. starting at `y.A` we have to check `x.B` *and* `B` as there is an area of `y.A` ∩ `B` outside of `x.B`.

3.1.3 A Realistic Example

Fig. 3 shows a realistic semantic coordinate system. The figure shows a small part of a huge set of domains of two hierarchies: a `city` hierarchy contains the cities, districts etc. (white boxes) and a `geo` hierarchy contains geographical entities such as rivers and mountains (grey boxes). As an example `downtown.hagen.city` is associated to `volme.river.geo`, because Volme is a river which flows through the downtown of Hagen.

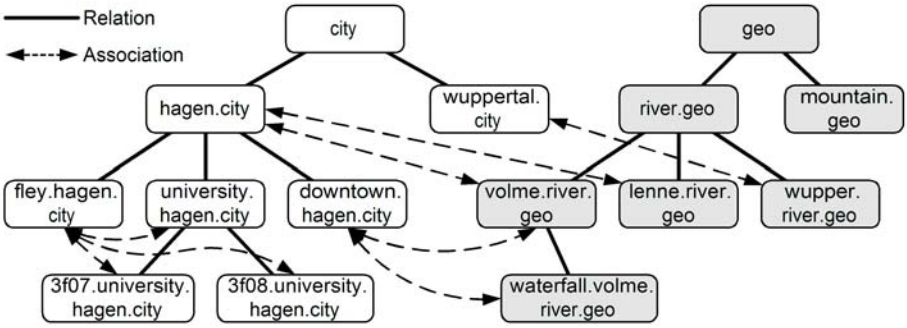


Fig. 3. A realistic semantic coordination system

Of course, there are more links conceivable between domains. We could, e.g. link two domains, if they have a common border. Bauer et al. proposed additional symbolic links to express topological aspects or to express proximity, which may be different from geometric distance [2]. We can store such links as meta data in a domain record, but they do not have any influence on the infrastructure described later. For the operations described in the next section, relations and associations are sufficient.

3.1.4 Operations on the Model

The primary goal of our approach is to provide uniform location information, which is independent from the actual positioning system. For each position, we want to provide both physical as well as semantic locations, even though typical positioning systems only offer one type. GPS e.g. offers physical locations, whereas some indoor positioning systems (e.g. [26]) directly produce semantic location output. Having both types, the application can choose the appropriate type (or even both types) for the specific operating condition.

To produce these location data, we have to perform one initial step: in our model, we currently can only express globally unique locations. Thus we have to perform a mapping, if the positioning system provides only local information. E.g., indoor radio systems (such as [4, 7, 14]) produce locally valid physical location output. They use, e.g., a special corner of the building as a reference point. So-called *mapping servers* set up for these positioning systems know the specific coordination system and transform from local to global locations.

Having a global valid location, we then can perform a resolution operation to get the respective other type. We distinguish two resolution operations:

- *Physical resolution*: Given a semantic location by its name *n*. What is the physical extension *d.c* of the domain *d* with this name?
- *Semantic resolution*: Given a physical location *p*. What are the names $\{n_i\}$ of the domains d_i which contain *p*?

The physical resolution is simple, as we only have to look up the appropriate domain and return *d.c*. The cost intensive part is the lookup mechanism which will lead to a network search function in the distributed implementation. We discuss this in a later

section. The more complex operation is the semantic resolution, as multiple hierarchies and domains may be involved. The algorithm to provide this resolution can be outlined as follows:

Look up an arbitrary domain d_0 with $p \in \Delta(d_0)$
$names \leftarrow \{d_0.name\}$
for all $d \sim d_0$ do
{
  if $p \in \Delta(d)$
    $names \leftarrow names \cup \{d.name\}$
}
return names

If we have an arbitrary domain which fulfils the first condition, we efficiently can loop through the associated domains. Again, the cost-intensive part is the lookup.

At this point, we want to outline a proof of the correctness of this algorithm. We want to show that $d.name \in names$ iff $p \in \Delta(d)$:

Step 1: we have to show that all names collected by the algorithm correspond to domains which actually contain p. This is obviously true, as this condition is part of the lookup and if statement.

Step 2: we have to show that there is not any solution that the algorithm does not collect. Assumption: there is such a solution domain h. Thus there must be no subdomain of h containing p (otherwise h would not be a solution, but this subdomain), i.e. $p \in \Delta(h)$. Further $p \in \Delta(d)$, which is ensured by the first statement of the algorithm. As a result $p \in (\Delta(h) \cap \Delta(d)) \neq \{\}$, therefore condition 3 (see above) is true and thus h and d are associated. As $h \sim d$ and $p \in \Delta(h)$, the algorithm would have collected h which is a contradiction to the assumption above.

3.2 Storing Domain Data – The Third Dimension

Until now, we make two demands on domain data:

- We can precisely specify an area $d.c$.
- There is an effective test, whether a point p is inside an area $d.c$ or not.

Since we only have a finite storage space, an area $d.c$ is usually approximated. Storing geographical data is a task of geographic information systems. Typical geo databases centrally store a large amount of geographical data. In our case however, we want to store only a small number of domains at a specific site. As a result, we can avoid heavyweight geo databases and use instead a lightweight toolkit to process polygonal data [24]. The toolkit handles all geometric operations in the runtime memory and especially can quickly check, if a point is inside or outside a polygon.

We store domain information using XML files in which the most important entry is the polygon specifying the area $d.c$. We conveniently can edit these XML files with the help of a graphical domain editor.

Two-dimensional polygons are sufficient for many domains. Unfortunately, our world is three-dimensional, thus for some domains it is necessary to take into account the third dimension. Some examples: Offices inside a building may have the same 2D coordinate; to map a physical location to the corresponding office, we have to consider the height. Another example is a street crossing another street via a bridge. On a bridge the 2D coordinates match both streets, thus we need the height to make a decision.

In principle, we could store a domain in three dimensions with the help of a volume model similar to those we find in CAD systems (fig. 4, left). With such a model we could specify arbitrary three-dimensional domains, but we have to consider two important disadvantages:

- As we cannot use the simple polygon inclusion test, it would be very cost-intensive to check if a point is inside a domain.
- It would be circumstantial to specify the three-dimensional domains, as most sources for domain data are basically two-dimensional, e.g. maps or ground plans from land registers.

As a solution, we avoid a full 3D representation and use a 2.5D representation as shown in fig. 4 (right). We request a domain to have a polygonal projection on a reference surface. As reference surface we use the WGS84 ellipsoid [27], which roughly can be viewed as an approximation of the earth's surface.

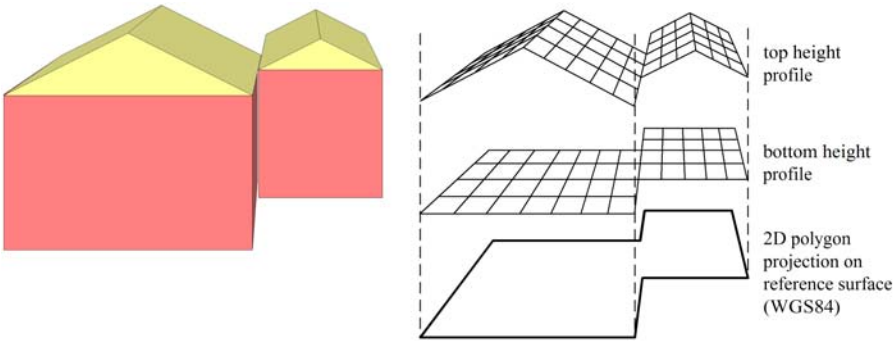


Fig. 4. Representing domains in three dimensions: full 3D (left), 2.5D (right)

We specify the third dimension of a domain with the help of two height profiles – a top and a bottom height profile. A height profile can be

- a surface specified by an array of reference points,
- a constant height, i.e. a surface parallel to the reference surface, or
- unspecified, i.e. the domain either extends to the sky or to the earth's centre.

Using the 2.5D representation, the check if a point is inside a domain is simple: We first check, if the 2D projection is inside the projected polygon of the domain with a simple polygon inclusion test. If not, the result is negative. Then we compute the domain's height values at the projected 2D position. If the height is inside the

height interval, the result is positive. Note that the height interval can be open at one or two sides, if the corresponding height profiles are unspecified.

As not all three-dimensional volume elements can be projected to a polygon and limited by a maximum of two height intervals, we cannot express some figures with our representation, e.g. some irregular polyhedrons. However, most conceivable realistic domains fulfil this requirement, thus we do not loose too much expressiveness.

The question is how to set the height profiles in reality. First, there is a huge class of domains, where the height is uncritical, e.g. countries or states. We either could leave these height profiles unspecified or set spacious constant heights such as [-500km...10000km].

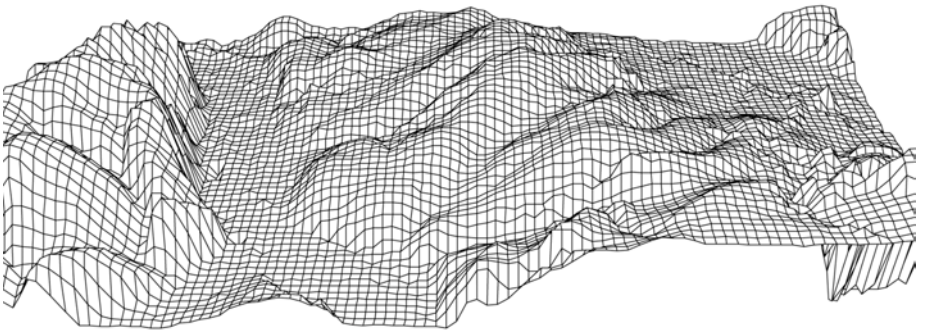


Fig. 5. Height profile of the city of Hagen

For buildings we can use constant heights, e.g., [178m...181m] for a specific floor. For the domains that require precise height profiles, e.g. streets, we can use height data as provided by land survey offices. Fig. 5, e.g., shows a height profile of the city of Hagen produced by a German land survey office [10]. Height profiles contain a number of reference points, in, e.g., a grid of 10m. We can easily compute height values between the reference points with the help of interpolation functions.

3.3 Performing Operations – The Location Server Infrastructure

We now switch from the abstract location model to an infrastructure storing model data. A location model is useless, unless we do not provide mechanisms to effectively run the required resolution operations. In principle, we could use one huge database and store hierarchies with the corresponding domains on a single server. A single database would be a bottleneck for a huge number of potential clients. In addition, information about local domains is usually available locally and difficult to administrate in a central database. As a solution, we use a distributed system of *location servers* each storing a number of domains.

3.3.1 The Infrastructure

Fig. 6 shows the distributed infrastructure which consists of three *segments*:

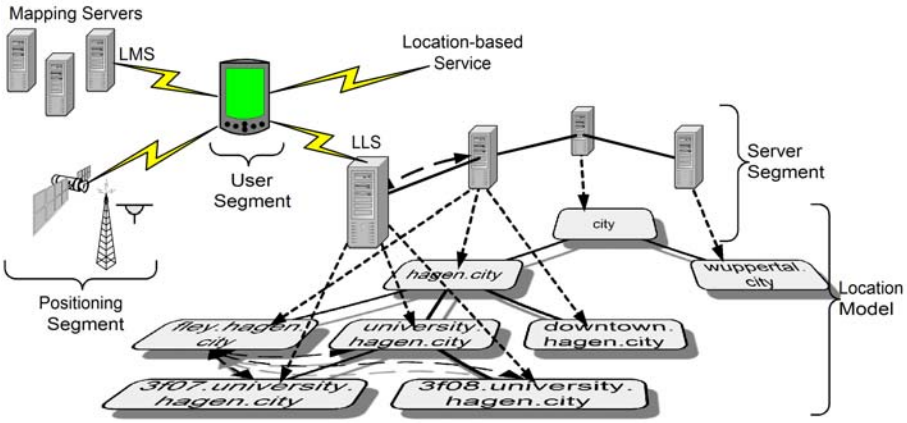


Fig. 6. The infrastructure

The *positioning segment* contains the positioning systems, e.g., indoor positioning systems, satellite navigation systems or systems based on cell phone networks. The runtime system accesses the positioning systems through *position drivers* which allow the change of positioning systems even at runtime. As many positioning systems provide local positioning data, we may need the help of *mapping servers* to transform local locations to global ones. Each mapping server is responsible for a specific positioning system, e.g. a mapping server inside a building may be responsible for the indoor positioning inside this building. A lookup procedure allows the mobile client to find the appropriate mapping server for a specific location, called the *local mapping server* (*LMS*).

The *user segment* contains the mobile nodes with a runtime system and the mobile part of the location-based service. Note that our infrastructure does not cover the network part of a location-based service. It depends on the mobile part to establish a connection to a specific server and to use the service. We developed a lightweight runtime system for the mobile nodes. We especially shift heavy duty tasks to the servers, thus the computational power of PDAs or mobile phones is sufficient.

The *server segment* contains the location servers that store the domain data. Each location server is responsible for a specific domain and all subdomains, for which no other location server exists. In our example, the location server for `hagen.city` covers `fley.hagen.city` and `downtown.hagen.city`, but not `university.hagen.city`, as this domain has its own location server. When a mobile node moves to a specific location, it automatically looks up an appropriate location server for the new domain, called the *local location server* (*LLS*). The LLS is the representative of the infrastructure for a mobile node. As mobile users are distributed among different location servers, this infrastructure is highly scalable. Especially, our system does not overload top-level servers.

The entire system is self-organizing. The location servers establish the links of relations and associations among each other automatically. Thus, a new location server

simply has to be configured using an XML file and turned on. A discovery procedure connects the server to its domain masters and looks up associated servers.

3.3.2 Looking Up Servers

Until now, we mentioned two different types of lookup: mobile clients looking up either an LMS or LLS, and a location server looking up other location servers (i.e. its master or associated servers). The latter lookup is called the *inter-server lookup*. As we do not have any central instance, the lookup has to run in a distributed manner. As location servers usually have a long lifetime, the links of relations and associations have a long lifetime as well (except for mobile domains, see open issues). As a result the inter-server lookup is uncritical and we can use well-established discovery mechanisms used in peer to peer networks, which may need a certain amount of time without significant drawbacks for the system.

Looking up the LMS and LLS is more critical: if a mobile node moves to a new location, the mobile user expects to use the service without interruption. Ideally, the user should not be aware, when the system performs a handover to a new LMS and LLS. For this, a mobile client automatically supervises its location and possibly discovers new servers. Our infrastructure supports the following lookup mechanisms:

- The mobile client can send lookup requests via broadcast messages using UDP multicast, or if available, Multicast IP.
- The mobile client can use service discovery protocols such as SLP or the network directories DHCP and DNS to ask for a server. For this, we defined new record types.
- The positioning system can distribute information about the LMS or LLS. Systems, which broadcast beacons, could e.g. distribute the corresponding network addresses in the beacon's payload.
- A mobile node can ask an arbitrary location server (e.g., the old LLS) for the new LLS.

The last point is very important: in principle, a mobile node has only once to know a location server to get the current LLS. A propagation mechanisms presented in [20] ensures that, after a certain (small) number of subsequent queries, an LLS will be found. The only prerequisite: there must be an uninterrupted sequence of relations and associations between the location servers, which in real environments is usually true.

We heavily can improve the lookup procedure using caches. When a mobile client looked up a server, it can store these data for a certain amount of time. Whenever it enters a specific area again, the lookup can thus be done without any network interaction.

3.3.3 Performing Resolution Operations

Having the lookup, we now can outline the distributed resolution operation, which is a distributed version of the algorithm presented in section 3.1.4:

- We first lookup an LMS which maps any local location data to global ones.
- We then lookup an LLS which returns one appropriate domain for a specific location and all associated servers.
- Subsequent queries to associated servers complete the resolution request.

This mechanism can entirely be controlled by the mobile client. We call this the *outbound mode*. In the outbound mode, the mobile node carries out a location resolution by subsequent queries to a number of location servers. This is efficient, as long as the mobile client is able to connect to every relevant location server. In some scenarios however, this is not possible: a mobile node may be separated from the global network by a firewall, which only allows pre-defined hosts to connect outside hosts. Or a mobile node using a cell phone network could have quick access to a server inside the phone network, but connections to servers outside are slow and cost intensive. In these cases, we use the so-called *inbound mode*: the mobile node only connects to one LLS, which in turn performs all subsequent queries to other location servers. Once an LLS queried the associated servers, the results are cached for future use.

3.4 Further Details

In this section, we summarize some further details concerning the location model:

Filtering: A specific location based-application may only be interested in a subset of all available domains. E.g., a bus schedule application may need semantic locations representing bus stations and not `geo` domains. If a mobile node only has to load specific domain information, we can drastically reduce the amount of network traffic. For this, we integrated so-called *domain filters*, which contain a description of subhierarchies included or excluded from the resolution process.

Compression: The number of associations can be very high for top-level servers. A request could lead to a large list of associated domains and cause heavy load on the server, especially in the inbound mode. We solve this problem with a compression mechanism: if the list of associations exceeds a certain limit, we connect a server to a top node of all associated domains. In fig. 2 we have an association between `A` and `B`. When compressing, we remove the association between `B` and `y.A` and just store one unidirectional association from `y.A` to `B`. To still get correct results, we need to modify the resolution algorithm: we now sometimes have to go down a hierarchy when checking associated candidates. As a benefit, we shift away processing load from top-level servers.

Proximity: The described model only resolves locations which are *inside* a certain area. We further could ask the system for domains in the nearer area. We call this operation the *proximity resolution*. We developed an algorithm, which collects all domains inside a certain circle with a minimum of network transactions.

3.5 Open Issues

Even though Nimbus reached a high level of completeness, we have some open issues:

Organizational Aspects: The technical platform is entirely decentralized. Nevertheless, for a specific hierarchy, we need a central organization to supervise the registration of subhierarchies. This problem is similar to the registration of Internet domain names. In addition to formal parameters such as the domain name, covered physical area etc., a domain has to satisfy informal conditions. E.g., if a city wants to register as a subdomain of `city`, one could require that it has a certain number of inhabitants. Our system currently does not support such issues and concentrates on the technical infrastructure. We could consider a second infrastructure to help organizations to control hierarchies and administrate additional information about domains.

Secret Domains: In the current implementation, our system stores domains with a public character. Every user can access all domains as domains such as rivers or cities have a certain meaning for the public. Some domains however should not be open for everyone, e.g. barracks in a military area. We are currently working on appropriate access control mechanisms.

Mobile Domains: Domains such as trains or ships permanently change their location. In principle, our system supports such domains, but they cause a high amount of updates messages. We currently work on a mechanism which avoids huge traffic and at the same time ensures consistency.

4 Conclusion

In this paper, we presented a location model especially designed for mobile users accessing location information. We introduced two resolution operations which provide a unique location data independently of the underlying positioning systems. We considered semantic locations and modelled three-dimensional locations with the help of a 2.5D approach. We took into account the distributed storage of location data in a decentralized federation of location servers.

Developers of location-based services and applications can use the Nimbus framework as a platform and do not have to deal with positioning capturing and resolution. As the corresponding infrastructure is self-organizing and decentralized, it is highly accessible and scalable.

References

1. Abowd, G. D.; Atkeson, C. G.; Hong, J.; Long, S.; Kooper, R.; Pinkerton, M, 1997: Cyberguide: A mobile context-aware tour guide. ACM Wireless Networks, 3: 421-433
2. Bauer, M.; Becker, C.; Rothermel, K.: Location Models from the Perspective of Context-Aware Applications and Mobile Ad Hoc Networks, Personal and Ubiquitous Computing, Vol. 6, No. 5, Dec. 2002, 322-328
3. Beigl, M.; Zimmer, T.; Decker, C.: A Location Model for Communicating and Processing of Context, Personal and Ubiquitous Computing, Vol. 6, No. 5, Dec. 2002, 341-357

4. Bahl, P.; Padmanabhan, V., N.: User Location and Tracking in an In-Building Radio Network, Microsoft Research Technical Report MSR-TR-99-12, Febr. 1999
5. Cheverst, K.; Davies, N.; Mitchell, K.; Friday, A.; Efstratiou, C., 2000: Developing a Context-aware Electronic Tourist Guide, in Proc. of CHI'00, ACM Press
6. Dey, A., K.; Abowd, G., D., 2000: CybreMinder: A Context-aware System for Supporting Reminders, Second International Symposion on Handheld and Ubiquitous Computing 2000 (HUC2K), Bristol (UK), Sept. 25-27, 2000, LNCS 1927, Springer-Verlag, 187-199
7. Hightower, J.; Boriello, G.; Want, R.: SpotON: An Indoor 3D Location Sensing Technology based on RF Signal Strength, Technical Report #2000-02-02, University of Washington, Febr. 2000
8. Hohl, F; Kubach, U.; Leonhardi, A.; Schwehm, M.; Rothermel, K.: Nexus - an open global infrastructure for spatial-aware applications. In Proc. of the 5th Intern. Conference on Mobile Computing and Networking (MobiCom '99), Seattle, WA, USA, 1999. ACM Press
9. Kindberg, T.; Barton, J.; Morgan, J.; Becker G.; Caswell, D.; Debaty, P.; Gopal, G.; Frid, M.; Krishnan, V.; Morris, H.; Schettino, J.; Serra, B.; Spasojevic, M., 2000: People, Places, Things: Web Presence for the Real World, Proc. 3rd Annual Wireless and Mobile Computer Systems and Applications, Monterey CA, USA, Dec. 2000
10. Land Survey Office North Rhine-Westphalia, http://www.lverma.nrw.de (in German)
11. Leonhardt, U.: Supporting Location-Awareness in Open Distributed Systems, PhD Thesis, University of London, 1998
12. Marmasse, N.; Schmandt, C., 2000: Location-aware Information Delivery with ComMotion, Second International Symposion on Handheld and Ubiquitous Computing 2000 (HUC2K), Bristol (UK), Sept. 25-27, 2000, LNCS 1927, Springer, 157-171
13. Navas, J.; Imielinski, T.: GeoCast – Geographic addressing and routing, Proc. of the 3rd ACM/IEEE inter. conf. on Mobile Computing and networking, Sept. 26-30, 1997, 66-76
14. Nibble Location System, http://mmsl.cs.ucla.edu/nibble
15. Open GIS Consortium, OpenLS Home Page, www.openls.org
16. Pradhan, S.: Semantic Locations, Personal Technologies, Vol. 4, No. 4, 2000, 213-216
17. Roth, J.: A Communication Middleware for Mobile and Ad-hoc Scenarios, Int. Conf. on Internet Computing (IC'02), June 24-27, 2002, Las Vegas, Vol. I, CSREA press, 77-84
18. Roth, J.: Context-aware Web Applications Using the PinPoint Infrastructure, IADIS Intern. Conference WWW/Internet 2002, Lisbon, Portugal, Nov. 13-15 2002, IADIS press, 3-10
19. Roth, J.: Flexible Positioning for Location-Based Services, IADIS International Conference e-Society 2003, Lisbon, Portugal, 3-6 June 2003, IADIS Press, 296-304
20. Roth, J.: Semantic Geocast Using a Self-organizing Infrastructure, Innovative Internet Community Systems (I2CS), Leipzig, June 19-21, 2003, Springer-Verlag
21. Schilit, B.; Adams, N.; Want, R., 1994: Context-Aware Computing Applications, Workshop on Mobile Computing Systems and Applications, Santa Cruz, CA, USA, 1994
22. Tomlin, C., D.: Geographic Information Systems and Cartrographic Modeling, Prentice Hall, 1990
23. UMTS Forum, Enabling UMTS/Third Generation Services and Applications, Report 11, http://www.umts-forum.org, Oct. 2000
24. Vivid Solutions, JTS Technical Specifications, http://www.vividsolutions.com, March 31, 2003
25. Vodafone Homepage, www.vodafone.com, 2003
26. Want, R.; Hopper, A.; Falcao, V.; Gibbson, J.: The Active Badge Location System, ACM Transactions on Information Systems, Vol. 10, No. 1, Jan. 1992, 91-102
27. WGS 84 - Implementation Manual, EUROCONTROL European Organization for the Safety of Air Navigation, Brussels, Belgium, Febr. 1998

A Localization Service for Mobile Users in Peer-to-Peer Environments

Marie Thilliez and Thierry Delot

Laboratory – CNRS UMR 8530
University of Valenciennes
Le Mont Houy
59313 Valenciennes Cedex 9 France
{Marie.Thilliez,Thierry.Delot}@univ-valenciennes.fr

Abstract. The recent emergence of handheld devices and wireless networks has implied an exponential increase of terminals users. So, today, service providers have to propose new applications adapted to mobile environments. In this article, we describe a new class of distributed M-services called proximity applications. In such applications, two or more handheld devices, physically close to each other, can communicate and exchange data in a same communication area. These applications need a high degree of flexibility, for an easy and rapid application development. Based on the Hybrid Peer-To-Peer (P2P) software architecture, different problems such as scalability, deployment, security, reliability and information retrieval in M-services, can be more easily resolved. In this article, we focus on the information localization problematic in proximity applications. Existing localization solutions (naming services, trading services, etc.) are not well adapted to the dynamicity and the heterogeneity imposed in this environment. So, we propose a decentralized localization service relying on the directory service model and adapted to the management of numerous distributed resources. This service allows users to locate and discover information, particularly location based services retrieved in function of users location.

1 Introduction

The emergence of both handheld devices and wireless networks [12] has implied an exponential increase of terminals users. Today, service providers have to offer new services adapted to mobile environments [1]. In this paper, we present a new distributed applications class: proximity applications [10]. This new class allows two or more handheld devices, close to each other, to communicate and exchange data in a secure way.

Due to the mobility of users, the information available in the communication area rapidly evolves and localization services are needed to provide a correct and up-to-date information to users. Without such mechanisms, users can not participate to the proximity services since they are unable to retrieve the information around them. For example, in a proximity electronic commerce application, the potential client has to

F. Crestani et al. (Eds.): Mobile and Ubiquitous Info. Access Ws 2003, LNCS 2954, pp. 271–282, 2004.

retrieve the different vendors and their interesting offers. As existing localization solutions do not support the constraints in term of distribution, dynamicity and heterogeneity of both terminals and networks imposed by proximity applications, new solutions have to be proposed. So, in this paper, we propose a localization service, relying on directory services technology, dedicated to mobile and dynamic environments. One of the main interests of this service is to provide location based services to users. Indeed, the service allows the evaluation of location based queries using location operators.
The paper is organized as follows: Section 2 describes the proximity applications, which are based on handheld devices and on mobile networks. Section 3 details the localization problematic. In section 4, we present our localization service. Then, we describe the prototype, and finally, we conclude and present the perspectives of our works in section 6.

2 Proximity Applications

2.1 Definition

Today, thanks to the evolutions of mobile and wireless networks, new services can be proposed to handheld devices users. Among these services, proximity applications, which are deployed in highly distributed environments and offer new devices use prospects to users. These applications are based on communication areas formed dynamically by juxtaposition of several wireless and mobile networks. They allow communications between different users physically close to each other. For example, a communication area can result from the association of a wireless LAN (Local Area Network) and a wireless PAN (Personal Area Network). Wireless communication areas are also highly dynamic since they evolve according to users mobility.

Proximity applications are relevant when users are close to each other. According to the location of these users, a set of services are proposed to them. Thus, through a proximity service, users can buy goods, exchange data or communicate with other users. In addition, the set of services can evolve when the user moves from an area to another one. In such a dynamic context, the life cycle of a proximity application is not predefined. First, a proximity service is created spontaneously when several users form a communication area and want to share information. Then this service evolves dynamically in function of the displacements of the participants and finally, the proximity application terminates when there are no more participants. To illustrate the concept of proximity applications, we detail an example in the next section.

2.2 Proximity Electronic Commerce (PEC) Application

Today, M-Commerce applications are more and more used by the cell phone users. However, based on mobile telephony networks, these applications do not evolve according to the location of users [4]. In the Proximity Electronic Commerce application, a user may choose and buy goods depending on his/her preferences and on his/her physical location. First, a potential client, fitted with an handheld device such

as cell phone, enters in the commerce zone and then, he/she can send queries in the wireless communication area dynamically formed by the juxtaposition of the different personal networks. These queries are evaluated by different peers and the client can retrieve several results such as merchants offers. If the client is interested in one or more specific offers, he/she goes to the merchant and buys the corresponding products.

2.3 Software Architecture

Due to the dynamicity and the heterogeneity of both networks and devices, a high degree of flexibility is required to deploy proximity services. In [10], we have shown the interest to base proximity applications on the hybrid Peer-To-Peer (P2P) architecture model [15]. Indeed, thanks to the partial centralization and the flexibility of this model, proximity applications developed using this architecture model are much more adapted to changing environments.

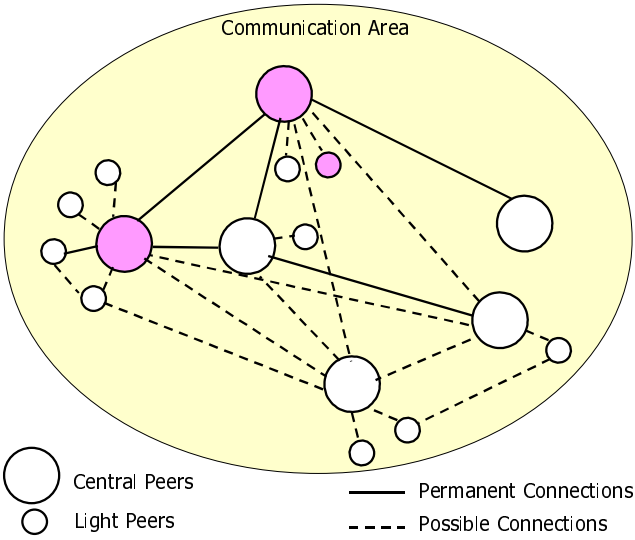


Fig. 1. Hybrid Peer-To-Peer Architecture

As shown in Fig. 1, two types of peers are distinguished in the hybrid P2P Model: the light peers and the central peers. Central peers centralize information and share it with the other peers. In fact, the type of a peer depends on its underlying hardware configuration and so central peers generally correspond to robust servers whereas light peers correspond to handled devices. Besides, the different peers communicate with the other peers either directly or using a central peer as relay. In the following, we present the requirements for a localization service adapted to mobile and dynamic environments and propose a decentralized solution based on the Hybrid P2P model.

3 Motivations

In a proximity application, each participant has to be able to easily discover and locate the other participants, the services, as well as the data available in its communication area. Today, many lookup solutions have been proposed to retrieve resources in highly distributed environments (naming services, trading services, directory services like LDAP [13] or UDDI[1]). These solutions generally provide a central service, which registers the available information that is not adapted to P2P environments. Moreover, existing services are based on a static approach and can not face to the dynamicity imposed by proximity applications. For instance, a trading service facilitates the offering and the discovery of services instances of particular types. It can be viewed as an object through which other objects can advertise (or export) their capabilities and match their needs against advertised capabilities (called import). Export and import facilitate dynamic discovery of, and late binding to, services. However, all the modifications brought to the services must be registered (i.e. exported) in the trader in an explicit way. This causes severe problems such as inconsistency problems when dealing with dynamic information [9].

As concern directory services, they provide very interesting features in the context of proximity services (scalability, querying facilities, authentication, security, etc.) but they also have severe limitations. For example, the support of distribution in directory services such as LDAP is crucial because a centralized management of a wide directory would cause important performance problems when accessing data. A solution to that problem is to partition directories, by country, by region or by organization in order to manage those different parts in separate servers. However, this distribution is managed with a big grain granularity as opposed to the requirements of proximity applications where the granularity of distribution has to be managed with a much more smaller granularity due to the use of many handheld devices. Moreover, existing directory services have not been designed to support mobility. To propose a localization service adapted to proximity applications, it is necessary to consider the very constrained resources of handheld devices and it is crucial to minimize the total number of transmissions over the network, in order to reduce battery consumption and the network bandwidth use.

4 A Localization Service for Proximity Applications

A participant of a proximity service has to be able to locate the data available in the communication area, that is naturally the information stored on his/her device but also the information managed by remote peers. Our localization solution does not rely on a centralized server but on the deployment of an extended directory service on each peer to support the dynamicity of the environment and to exploit the benefits of the underlying hybrid P2P architecture. Naturally, the functionalities of the services deployed

[1] Universal Description, Discovery and Integration: http://www.uddi.org

on the different peers have to be adapted to their underlying resources. Indeed, if a peer presents a lot of resources (as it is generally the case for a central peer), it can easily store and share information about the other connected peers. On the contrary, the localization service deployed on a light peer may only stores few information locally and provide to users a mean to retrieve information stored on remote peers. Therefore, when a light peer enters in a communication area, it is attached to at least one central peer what facilitates the information retrieval process.

4.1 Information Model

In this section, we present the information model of our localization service. As it is the case for directory services, this model relies on a tree structure, called the Directory Information Tree (DIT), used to represent hierarchically the information. The information model is centered around entries. Each entry contains information about one object, a person or a country for example. An entry is composed of a list of attribute/value pairs. Each attribute may be defined either mandatory or optional. It has a name and/or an alternative name, as for example ST for the building stages. These names may be used to generate the distinguished name (dn) of an entry which unambiguously identifies it. This information model provides flexibility and simplicity: attributes can be multi-valued and new attributes value can be added to entries dynamically at the execution time. An example of entry is presented in Fig. 2 for the PEC application. This entry is represented using Directory Services Markup Language (DSML)[2] which provides a means for representing directory information as an XML document.

```
<dsml:entry dn="tm=ApplicationData, b=ShoppingMall, st=First, sc=South">
 <dsml:objectclass>
  <dsml:oc-value>Vendor</dsml:oc-value>
 </dsml:objectclass>
 <dsml:attr name="name">
  <dsml:value>Virgin</dsml:value>
 </dsml:attr>
 <dsml:attr name="type">
  <dsml:value>Music Store</dsml:value>
 </dsml:attr>
</dsml:entry>
```

Fig. 2. Example of a vendor entry

In the DIT, two main parts are distinguished. First, the DIT contains System Metadata which describe the system characteristics of the underlying peer. For example, these metadata may detail software and hardware resources, network access properties, the degree of mobility, and so on. This entry of the directory service cannot be reached by the other peers. The second part of the DIT is used to store application data. These data represent hierarchically the information available and shared in the communication area. Different types of information may be stored such as information on the geographic location (for example the plan of the shopping mall in the PEC

[2] http://www.oasis-open.org/committees/dsml/

application), or the set of services available on each peer. The same directory service structure is deployed on each peer. The DIT is always formed of two parts (System Metadata and Application Data). Nevertheless, the amount of data stored in the directory service is adapted in function of the peer resources. Moreover, remote information may also be referenced in the DIT in order not to store it on the local peer. This aspect is very interesting for light peers which resources are strongly limited and relies on the use of referral entries which contain the address of the remote server. In our localization service, we extend the referral entry proposed in the LDAP standard to store metadata characterizing the referenced peer. Indeed, since the query evaluation may be constrained, it is very important during the evaluation process to retrieve information on the referenced peer to determine whether the query has to be forwarded to the remote peer or not. This is particularly important when dealing with constrained query evaluation since it is necessary to find as soon as possible the most interesting sites to compute the query result.

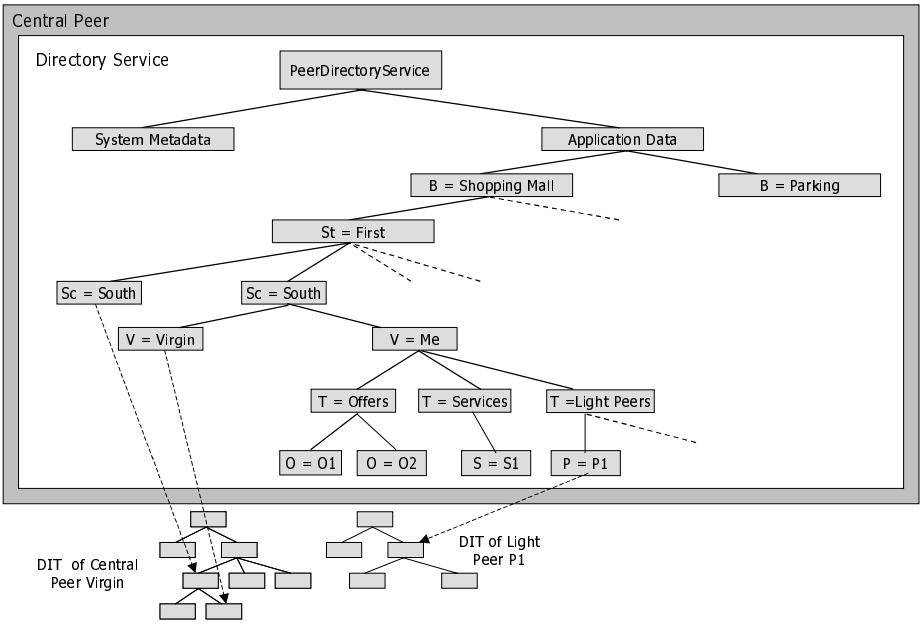


Fig. 3. Example of DIT deployed on a central peer

To illustrate the structure of the DIT, we propose in Fig. 3 the DIT created for the set of coloured peers represented in the Fig. 1. One of the interests of this information model resides in the standardization of the model on each peer that hides the underlying heterogeneity between the different peers. Another interest is the use of references (characterized with metadata) between the different peers which can be exploited by the query evaluator to limit the use of peers resources and a better use of the networks bandwidth when necessary. This information model also contains location information used to evaluate proximity queries introduced in the next section.

4.2 Types of Queries

One of main interests of directory services is the simplicity of proposed query languages. However, this is also very restrictive [2] and, here, we have chosen to express queries evaluated by the localization service in XQuery[3]. In the following, we present the different types of queries evaluated by the localization service: filter queries, path expression queries and location queries.

- Filter queries

This category contains relatively simple queries which are generally evaluated on existing directory services. The "filter" is composed of a conjunction and/or disjunction of predicates applied to a set of directory entries. The example of filter query presented in Fig. 4 retrieves all the music stores of the shopping mall.

```
<dsml:dsml xmlns:dsml="http://www.dsml.org/DSML">
  <MusicStore>
  {
   for $i in document("pec.xml")//dsml:entry[dsml:objectclass/dsml:oc-value =
"Vendor"]
   where $i/dsml:attr[@name = "type"]/dsml:value = "Music Store"
   return $i/dsml:attr[@name = "name"]/dsml:value
  }
  </MusicStore>
</dsml:dsml>
```

Fig. 4. Example of filter query

- Path expression queries

With filter queries, users can only get flat answers ; all containment relationships between entries are lost in the query result. Path expression queries propose to exploit the tree structure of the DIT to return structured query results. Fig 5. proposes an example of path expression query which retrieves the offers presented by merchant.

```
<dsml:dsml xmlns:dsml="http://www.dsml.org/DSML">
 <result>
   {for $a in document("pec.xml")//dsml:entry[dsml:objectclass/dsml:oc-value =
"Vendor"]
   return
    <Vendor>
    {$a/dsml:attr[@name = "name"]/dsml:value}
    {for $b in document("pec.xml")//dsml:entry[dsml:objectclass/dsml:oc-value =
"Offers"]
     where $b/dsml:attr[@name="vendor"]/dsml:value =
$a/dsml:attr[@name="name"]/dsml:value
     return
      <Offer>
      {$b/dsml:attr[@name = "description"]/dsml:value }
      </Offer>}
    </Vendor>
    }
 </result>
</dsml:dsml>
```

Fig. 5. Example of path expression query

[3] http://www.w3.org/TR/xquery/

- Location queries

In proximity applications, it is often very interesting for a participant to select an information according to its location or its proximity. Since the presentation of the concept [5], querying location dependent information in mobile environments has become an important research area. Today, proposed solutions mainly concern data management issues of mobile objects and their location information [3, 6, 8, 14]. Here, contrary to other approaches, our purpose is to retrieve approximate solutions using the location metadata stored in the DIT. Our solution is based on the use of several simple and user-friendly operators used to verify proximity constraints:

– ***op:inside($Srcval as item, $LocationType as item) as boolean***

The *inside* operator may be used to retrieve elements in a same area. The parameter "LocationType" is used to determine this research area. For example, this parameter may correspond to a particular stage of a building. The "Srcval" parameter appears in all the operators presented in this section. It is used to explore the set of possible solutions (computed thanks to the other clauses of the query). Thus, for the inside operator, this parameter is used to verify if an object is located in the $LocationType area.

– ***op:closest($Srcval as item, $Location as item?) as boolean***

The *closest* operator may be used to retrieve one particular element at the shortest distance from the issuer of the query or from the specified location parameter. The "?" symbol is used to precise an optional parameter. The "Location" parameter allows to describe the location from which the closest element should be retrieved. If this parameter is not defined, the location used to compute the query result is the one of the issuer of the query. To illustrate the use of this predicate, the query presented in Fig. 5 selects the closest TV repairer from the user who submitted this query.

```
<dsml:dsml xmlns:dsml="http://www.dsml.org/DSML">
   <TVRepairer>
    {for $i in document("pec.xml")//dsml:entry[dsml:objectclass/dsml:oc-value =
"Vendor"]
      for $j in $i//dsml:entry[dsml:objectclass/dsml:oc-value = "Services"]
      where $j/dsml:attr[@name = "description"]/dsml:value = "repair"
      and closest($i)
      return  $i/dsml:attr[@name = "name"]/dsml:value  }
 </TVRepairer>
</dsml:dsml>
```

Fig. 6. Location query illustrating the closest operator

– ***op:close($Srcval as item, $Location as item?, $Distance as integer?) as boolean***

The *close* operator is an evolution of the closest operator. It can be used to retrieve several elements close to the issuer or close to a specified location parameter. This operator may also be used with a distance parameter. This optional parameter is an integer representing the number of meters, which defines the maximal distance between the specified target and the issuer (or the specified location). In that case, the query result is true for each element, for which the distance between it and the issuer (or the location target) is inferior or equal to the specified distance parameter. For example, to retrieve "the Fast Foods in the fifty meters around Virgin", the operator close ($i, <dsml:value>Virgin</ dsml:value >, 50) is added to the query.

4.3 Query Evaluation

The information model of the localization service considered in this paper presents a fundamental difference with the other directory model. Indeed, numerous entries reference remote localization services. To widely exploit this distribution, and to avoid to users to write one query for each queried server as it would be done in traditional directory servers, distribution transparency must be assured. The query evaluator has to be able to retrieve in a single query all the information needed, even if this query concerns resources managed on several different sites. So, when a query concerns references to remote localization servers, query evaluation have to be continued on the different referenced servers.

Besides, since the localization service may be deployed on handheld devices with very constrained resources, the query evaluator has to provide the ability for the user to limit the resources according to the evaluation process. For instance, the user may want to limit the size of the query result or the time allowed for the query evaluation. In this last case, the query evaluator will only deliver to the user the partial result computed in the specified time.

One of the main difficulty in our environment concerns the evaluation of location based queries. First, the evaluator has to define whether the query is a location aware query (which does not depend on the issuer location) or a location dependent one [7]. Location aware queries are managed like standard filter queries whereas the position of participants have to be determined for location dependent queries. This localization process can be adapted depending on the resources of the underlying peer. For instance, it can be based on geographical localization technologies such as GPS but, as handheld devices do not often provide such features, the localization will be generally based on location metadata stored in the DITs.

5 Prototype

A prototype of the localization service presented in this paper, has been implemented and presented, in October 2003, at the French BDA Conference [11].

The Proximity Electronic Commerce application is selected to validate the prototype. In this prototype, the different peers are connected with Wifi technology in ad hoc mode. Pocket PCs Compaq Ipaq H5450 are used as light peers and represent the potential clients. Their data are stored in XML files. The central peers represent the vendors and their data are stored in the OpenLDAP directory server. The content of these directories is exported in DSML files in order to evaluate queries. To optimize the performances at the time of the updates, the different attributes of the neighbored peers are stored in indexes files.

Finally, in our prototype, users do not express directly their queries using DSML. In fact, those queries are automatically generated thanks to the choices performed by users using the localization service interface. The choices proposed to users in the query interface are dynamically parameterized thanks to an XML file broadcasted in the commerce zone. This file contains the available search criteria as well as parame-

ters used to establish the correspondences between the parameters of the query and the directory information. Fig. 7 illustrates the choices performed through the query interface to generate the DSML request presented in Fig. 8.

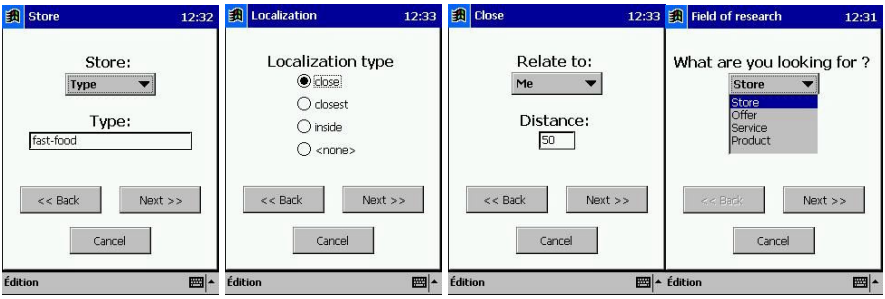


Fig. 7. Query interface

```
<?xml version="1.0" encoding="UTF-8"?>
<dsml:searchRequest dn="o=PeerDirectoryService"
xmlns:dsml="urn:oasis:names:tc:DSML:2:0:core">
  <dsml:filter>
    <dsml:and>
      <dsml:equalityMatch name="objectClass">
        <dsml:value>Store</dsml:value>
      </dsml:equalityMatch>
      <dsml:equalityMatch name="type">
        <dsml:value>fast-food</dsml:value>
      </dsml:equalityMatch>
    </dsml:and>
  </dsml:filter>
  <dsml:localization>
    <dsml:close>
      <dsml:location>me</dsml:location>
      <dsml:distance>50</dsml:distance>
    </dsml:close>
  </dsml:localization>
</dsml:searchRequest>
```

Fig. 8. Example of an extended DSML query

Our goal designing this version of the prototype was to validate our approach. Indeed, using this prototype, a potential client, fitted with an iPAQ provided with Wifi technology, can locate information (offers, products or vendors) in a shopping mall. The client can also query information in function of his/her location. However, thanks to this prototype, we have highlighted energy consumption problems using mobile devices. Thus, we are now considering optimization solutions to economize light peers resources.

6 Conclusion and Perspectives

In this paper, we have presented a localization service relying on the hybrid P2P software architecture and well suited to proximity applications. Our solution is fully decentralized since one localization service is deployed per terminal that provides sev-

eral advantages and highly facilitates the support of dynamicity. This service is based on directory services and also proposes an extended information model as well as a query evaluator providing distribution transparency and location queries. Even if we have focused on the light peer to central peer communication in this article, the communication is bi-directional. For instance, in the PEC application, vendors offers may be broadcasted from central peers to interested light peers.

As regards query optimization, several ways appear, several different search strategies can be applied in the query evaluator according to the type of considered applications. The more important aspects are the ones of location queries and referrals, which would allow to reference remote directories in the DIT. Distribution transparency completely changes the way query are evaluated and the generation of sub-queries towards the referenced servers according the resources of underlying terminals must be studied.

Acknowledgment

The authors wish to thank Sylvain Lecomte for his helpful comments on this paper.

References

1. M. Bechler, H. Ritter, J. H. Schiller, Quality of Service in Mobile and Wireless Networks: The Need for Proactive and Adaptive Applications, Proceedings of the 33rd Hawaï International Conference on System Sciences (HICSS), 2000.
2. T. Delot, B. Finance, Managing Corba Objects with Dynamic Behaviour in a Directory, Int. Symposium on Distributed Objects and Applications (DOA), 2001.
3. M. H. Dunham and V. Kumar, Location dependent data and its management in mobile databases, Int. Workshop on Mobility in Databases and Distributed Systems (MDDS), 1998.
4. C. Herault, N. Bennani, T. Delot, S. Lecomte, M. Thilliez, Adaptability of Non-Functional Services for Component Model, Application to the M-Commerce, Proceedings of Int. Symposium on Advanced Distributed Systems (ISADS), 2002.
5. T. Imielinski and B.R. Badrinath, Querying in Highly Mobile and Distributed Environments, Int. Conf. Very Large DataBases (VLDB), 1992.
6. D. L. Lee, J. Xu, B. Zheng, W-C. Lee, Data Management in Location-Dependent Information Services, IEEE Pervasive Computing, 2002.
7. A.Y. Seydim, M. H. Dunham, V. Kumar, Location Dependent Query Processing, Proceeding of MobiDE, 2001.
8. A.P. Sistla, O. Wolfson, S. Chamberlain, S. Dao, Modelling and Querying Moving Objects, Int. Conf. on Data Engineering (ICDE), 1997.
9. Z. Tari, G. Craske, A Query Propagation Approach to Improve Corba Trading Service Scalability, Int. Conf. on Distributed Computing Systems (ICDCS), 2000.
10. M. Thilliez, T. Delot, S. Lecomte, N. Bennani, Hybrid Peer-to-Peer Model in Proximity Applications, Int. Conf. on Advanced Information Networking and Applications (AINA), 2003.

11. M. Thilliez, Y. Colmant, T. Delot, Query Evaluation in ISLANDS, Les 19èmes journées de Bases de données Avancées (BDA), 2003.
12. U. Varshney, and R. Vetter, Emerging Mobile and Wireless Networks, Communications of the ACM, Volume 43, Issue 6, 2000.
13. M. Wahl, T. Howes, S. Kille, Lightweight Directory Access Protocol (v3), Internet RFC-2251, 1997.
14. O. Wolfson, B. Xu, S. Chamberlain, and L. Jiang, Moving objects databases: Issues and solutions, Int. Conf. on Scientific and Statistical Database Management (SSDBM), 1998.
15. B. Yang, H. Garcia-Molina, Comparing Hybrid Peer-To-Peer System, Int. Conf. On Very Large DataBases (VLDB) Conference, 2001.

Sensing and Filtering Surrounding Data: The PERSEND Approach

David Touzet, Frédéric Weis, and Michel Banâtre

IRISA, Campus de Beaulieu, 35042 Rennes, France
{dtouzet,fweis,banatre}@irisa.fr

Abstract. In the mobile computing area, short-range wireless communication technologies make it possible to envision direct interactions between mobile devices. In the scope of data access, devices can now be considered as both data providers and data consumers. Thus, each device can be provided with a remote access to data its neighbours agree to share. Such a service enables applications to consult a set of data providers which dynamically evolves according to the mobility of the neighbouring devices. The set of data sources an application may access by this way is therefore representative of its physical neighbourhood. In this context, we propose to design a tool making possible the continuous consultation of neighbouring shared data. We present, in this paper, the PERSEND system we develop in this scope. Based on relational databases systems, PERSEND enables applications to define continuous queries over neighbouring data.

1 Introduction

The recent development of powerful mobile devices has made mobile computing a more and more popular paradigm. Typical mobile environments are today composed of mobile devices accessing data by the mean of a fixed infrastructure (such as 802.11b cells). These environments are based on non-symetrical interactions since the infrastructure is the only data provider, the mobile devices being confined to a role of data consumers. These client/server exchanges mainly bear on structured data (such as visiting cards...) which are usually stored in database systems. Using wireless communication channels, mobile devices can download information as long as they are located in the network communication area. Disconnections from the fixed network occur as soon as mobile devices move away from the network communication area. Although, in such environments, they are supposed to be temporary events, disconnections raise many challenges in the data management domain. Some approaches, such as hoarding and optimistic replication [1], have been introduced in order to address data access issues.

Recently, in the area of pervasive environments, the rise of short-range communication technologies, such as Bluetooth [2], has made the emergence of a new type of mobile environments dealing with direct and proximate interactions between devices possible. Considering each device as a potential data provider, they aim to promote direct exchanges between physically close enough devices. Due to their limited communication range, considered devices can only directly communicate with their closest neighbours. Thus, two mobile devices are declared to be neighbours as soon as they are able to directly communicate one with the other. In the remaining of this paper, such

F. Crestani et al. (Eds.): Mobile and Ubiquitous Info. Access Ws 2003, LNCS 2954, pp. 283–297, 2004.

environments are called *proximate environments*. This approach enables us to envision new kinds of applications.

In proximate environments, each device sharing some local data has to be considered as a data provider. In such a context, the data set each device can access at a given time includes both local data and data shared by neighbouring devices. Mobility, which makes devices getting closer or going away from the others, breaks and establishes neighbourhood relationships. Due to this mobility, the set of neighbours of each device evolves in time. Therefore, the data set each entity can access evolves according to its set of neighbouring data providers. As devices may update their own shared data, the data set a device can access also has to evolve according to the modifications processed in its neighbourhood. Consequently, data which are available to a device in a proximate environment not only depend on the neighbouring devices but also on the updates these devices perform on the data they share.

In this paper, we propose a system enabling applications to access available neighbouring data in the scope of proximate environments. As a large part of data is today managed by databases, our system is developed using relational database systems (RDBMS). Rather than providing a complete view of the available data, our system is designed to enable users to query these data for some specified subsets. Just as the whole set of available data, the data subsets queried by users evolve according to the set of neighbouring data providers. However, they also have to reflect the conditions users specify. In order to enable applications to continually be aware of their currently available data, our system provides them with persitent data sets which match the users' conditions. For this purpose, it enlarges some works previously performed in the *continuous queries* area.

This paper is organized as follows: in the next Section, we present the concept of continuous query and highlight the main issues to be addressed in order to process such queries in proximate environments. Section 3 details the semantics we have considered to develop proximate continuous queries. In Section 4, we describe PERSEND (PERsistent SEnsing for Neighbouring Data), the continuous query system we designed for proximate environments. We discuss, in Section 5, some implementation issues. Section 6 deals with related works. Finally, Section 7 presents our conclusions and future works.

2 Continuous Querying Challenges in Proximate Environments

In this section, we first review some of the main existing continuous query systems. We briefly present the context of these studies and the architectures which have been developed. Then, we explain how proximate environments differ from these works and what kind of specific constraints they have to face.

2.1 Continuous Queries: Goals and Design

In the field of database systems, users querying continually changing databases may want to be notified of data updates which occured since a query has been submitted. The simplest way to provide this service is to process the query again each time the

database is updated and to return to the user the corresponding data set. This approach can prove to be extremely ineffective. Given a user's running query Q bearing on a single table, let us consider the insertion of a record r which is relevant to Q. The user can be provided with an up-to-date data set without requiring Q to be run again: it can simply be achieved by adding selected fields from r to the current result set.

Terry introduced the continuous query concept in order to manage such challenges [3]. They are defined as queries that continually run once issued. Considering a model limited to append-only tables (tables only accepting new insertions), Terry designed a system making the continuous querying of the Tapestry messaging system possible. By the mean of continuous queries specified using the SQL querying language, users can define some messages filters. Thus, they are notified as new messages matching their filters are received by the system and inserted in storage tables. Submitted continuous queries are recorded by the system and are processed so as to build corresponding incremental queries enabling to efficiently get up-to-date results.

The Tapestry continuous query model is highly centralized since a front-end server has to store all managed data and to perform the whole querying process. In the sensor database area, on-going works on *long-running queries* are extending this model [4]. Observing that interconnected sensors are now widely deployed [5], sensor database systems aim to make an efficient querying of these information providers possible. Centralized schemes proved to be unsuited to sensor database interrogations:

- sensors continually have to send captured data to the front-end server, thus overloading the communication network;
- answering a query on a single sensor is performed by searching through the entire database: this process includes data from non-relevant sensors.

Typical queries submitted to sensor database systems ask for values currently measured by some sensors. For example, a user can ask temperature sensors situated in a building for the current temperature every ten minutes. As each sensor is assumed to embed storage, computing and networking capabilities, distributed query processing schemes can be used. For this purpose, a front-end server is used to store a description of managed sensors. Each sensor is associated with an ID and some physical attributes (such as its location). Thus, queries executions can be distributed over sensors specified by users' conditions: non-relevant sensors are not included in the query execution plan. Moreover, only data which are relevant to the query are transmitted from concerned sensors to the front-end server. Otherwise, the concept of *virtual relation*, introduced by Bonnet, enables users to interrogate a sensor database using the SQL syntax [4]. Data scanned by sensors are indeed represented as append-only relational tables in which new measures are inserted associated with a time stamp.

Beyond this distributed model, the moving objects databases deal with continuous queries which involve mobile objects, such as cars. Usual storage schemes are not suited to manage such objects. As the value of their location continually evolves, keeping an up-to-date representation of mobile objects requires databases to be continually updated. Observing that the description of a mobile object motion is updated less frequently than its position, these systems chose to associate motion vectors to objects representations [6]. Thus, each mobile entity is associated with its last known location, its motion vector and the time stamp of its last update. Assuming a mobile object has

kept an unchanged trajectory since its last update, its actual position can be calculated at any time without requiring its stored position to be explicitly updated. Continuous queries submitted to moving objects databases may involve several mobile entities. For example, a user can ask for the devices which are less than one hundred meters away from him. The described storage scheme enables systems to compute at once the data set currently associated to a continuous query, and further ones. For this purpose, a set of tuples $(r, begin, end)$ is built, where the record r belongs to the data set between time $begin$ and time end.

2.2 Continuous Querying of Proximate Environments

The architecture of proximate environments fundamentaly differs from those of studied continuous queries systems. Since proximate environments are totally distributed, each mobile device potentially has to be considered as both a data provider and a query transmitter. As opposed to this model, continuous queries over Tapestry are based on a centralized scheme. In sensor database systems, the continuous querying process is distributed between two different kinds of entities: the sensors are the data providers and the front-end server is the query transmitter. Moving objects databases offer a more flexible architecture. Queries can be processed from a mobile device over multiple mobile objects which store each a subset of required data [6]. These systems however assume a global connectivity between all mobile objects by the mean of a wireless communication infrastructure. This assumption is not valid anymore in a proximate context.

Beyond the developed architectures, many differences can be observed between proximate environments and existing continuous query systems. Contrary to moving objects databases, we do not assume any knowledge about devices motion. This implies that the only computable data sets are those that currently satisfy the continuous queries. Moreover, the data model proximate environments have to consider is more flexible than those previously described. Consider users sharing information stored in their address book. Insertions, removals and updates should be allowed by a proximate continuous query system. Such handlings are not managed by the Tapestry continuous query system which is limited to append-only tables. Likewise, data scanned by sensor databases are modelized by virtual relations which are also append-only.

Proximate environment querying has to deal with more constraints than previously described querying systems. Its objectives are also different. Whereas most of the systems attempt to make seamless querying of data providers possible, whatever their physical location, data providers a device can query in proximate environments are restricted to the device's vicinity. Thus, a continuous query submitted in a proximate environment provides the user with a continuous view of available data matching the query in his physical neighbourhood. We call such a data set a *Continuous Result Set (CRS)*. This model implies that data stored by a device should be required to answer continuous queries issued by any device's neighbour. Consequently, devices involved in proximate environments have to watch for local data updates in order to notify interested neighbours of the occured modifications. Notifications have to enable interested neighbouring devices to keep continuous result sets up-to-date.

In order to make efficient continuous queries over proximate environments possible, we have identified three main challenges to address.

Table 1. The cd_to_sell table

nu_cd	cd_title	cd_price
1	War	8
2	Transformer	16
3	Animals	20
4	Kind Of Blue	32

Data Providers Management. In a proximate environment, each continuous query is submitted to the data providers located in the vicinity of the query transmitter. Each device has to know its neighbouring devices. Reminding that considered devices communicate by the mean of short-range wireless technologies, and that they are mobile, the neighbours set of a device may evolve. Consider two devices A and B. As B leaves the vicinity of A, continuous queries issued by A have no longer to take data from B into account. Conversely, as a new neighbour C gets closer to A, continuous queries submitted by A have to deal with data stored by C.

Assessment of the Data Updates Impact. In proximate environments, continuous queries have to return the data set both being stored in the device's vicinity and matching the user's expressed conditions. Let a *querying device* be a device which has issued a continuous query. Applications initiating such queries are called *querying applications*. Devices neighbouring a querying device are called *queried devices*. As data stored on a queried device are modified, the associated querying device has to reflect the processed modifications, assuming that they bear on data relevant to the continuous query. In order to highlight these problems, we study an example hereafter. Let A provide its neighbours with the list of audio CD its user sells (see Table 1). Now, consider a neighbouring device B having issued the continuous query CQ_B: *I'm looking for audio CD which price is between 10 and 20*. Let $CRS(CQ_B)$ be the continuous result set associated with this query. Data stored by the queried device can be modified in three different ways:

- *data removal*. Removed data which are relevant to CQ_B (*i.e.* their price is between 10 and 20) also have to be removed from $CRS(CQ_B)$.
- *data insertion*. Inserted rows matching CQ_B's conditions have to be included in $CRS(CQ_B)$.
- *data update*. Let us assume that CD prices are reduced by half. Such an update may have three kinds of consequences. First, some of the rows which were relevant to CQ_B may no longer match its conditions. Second, some of the non-relevant rows may now be relevant to the query's conditions. Third, data from $CRS(CQ_B)$ which are still relevant to the query have to reflect the executed update. In our example, the price of *Animals* has to be set to 10 in the continuous result set. Moreover, *Transformer* has to be removed from $CRS(CQ_B)$ whereas *Kind Of Blue* has to be added to (with a price of 16).

Notification of Data Modifications. A querying device has to be notified when remote data involved in a continuous query it has issued are modified. Queried devices are

responsible for this task: they have to notify querying devices in their neighbourhood of the modifications to perform on their continuous result sets. Let us consider the update operation of the previous example. Assuming that it knows what data are handled by CQ_B, device A is able to deduce what modifications B has to perform in order to keep its continuous result set up-to-date. For this purpose, A can send to B a message containing three *update commands*:

- remove $Row\ \#2$ from $CRS(CQ_B)$
- add $(4, Kind\,Of\ Blue, 16)$ to $CRS(CQ_B)$
- set $price$ to 10 at $Row\ \#3$

Now we have highlighted the challenges to be addressed, let us define some specific semantics for proximate environments.

3 Defining Semantics for Proximate Continuous Querying

In this section, we present some data querying semantics which are compatible with the specific constraints relative to proximate environments. We first study those related to the vicinity management. Then, we investigate the impact of the duration parameter introduced by continuous queries.

3.1 Vicinity Relative Issues

Querying neighbouring information supposes that involved devices share some of their local data. Two different data modes are considered: *private* and *shared*. Data in private mode only accept local accesses. Conversely, shared data can be read by any neighbouring device. The data mode is defined at the table level: data from a shared table can be queried by remote devices whereas those from private tables can not.

In a proximate environment, several devices can simultaneously store some data representing a same physical object (or person). These data can be concurrently updated in different ways according to the devices storing them. Due to the absence of a centralized control, the consistency of these co-existing copies can not be insured. Likewise, and for the same reasons, no global identification schemes are available: stored data are identified in an independent way on each device. Therefore, we consider that each database entry locally describes a unique object. In order to prevent any interference between remote identification schemes, objects are globally identified by a $(DeviceID, LocalID)$ pair.

In this context, join queries have to be carefully managed. The absence of a global identification scheme implies that database entries are not identified in the same way on every device. Moreover, a same $LocalID$ may be associated with distinct objects depending on the device. Thus, database objects having no relationship may be joined as a result of a join query involving remote tables. Meaningfull join queries have therefore to be processed on a single device. Consequently, distributed join queries have to be independently processed on each neighbouring device before merging the computed results.

3.2 Duration Relative Issues

Considering continuous queries introduces some temporal issues. Indeed, built continuous result sets evolve according to data which are available in the devices' vicinity. Common querying languages, such as SQL, have been designed to define and build static data sets. Some of the tools and functions they provide do not suit to manage data sets subject to variations. Thus, SQL enables users to call some aggregation functions (such as `max`, `sum`, `count`...) in the queries they define. These functions usually compute a single value from a static data set. Likewise, the use of the `distinct` keyword ensures that a returned static data set is composed of distinct rows only.

In order to manage changing data sets, dynamic semantics have been associated to these functions. The value these functions return has to mirror the current state of the continuous result set. For this purpose, this value has to be re-evaluated each time the continuous result set is updated. Fortunately, in many cases, the value to be returned can be computed without requiring the complete CRS to be scanned. For example, the value returned by a call to `count(*)` can be easily managed: it as to be incremented each time a row is inserted in the CRS and to be decremented for each removed row.

We have introduced theorical issues which are specific to proximate environments. We now present the PERSEND querying system.

4 Design of a Vicinity Continuous Querying System

Besides classical queries, the PERSEND querying system makes the continuous querying of proximate environments possible. In this section, we detail the architecture of this system. First, we introduce some necessary SQL extensions which make the definition of continuous and proximate queries possible. Then, we describe the different components composing PERSEND. Finally, we present the way the PERSEND system manages proximate continuous queries.

4.1 Vicinity Continuous Querying with SQL

SQL has been designed to handle data stored in relational databases. It enables users to issue instantaneous queries, that is queries which return data matching expressed conditions just as they are executed. However, its syntax provides no way to define continuous queries. We have therefore introduced the keyword `continuous` in order to distinguish continuous consultations from instantaneous ones. Positioned at the beginning of a consultation query, it indicates that the query has to be considered as continuous (see Query 1).

Query 1 *Continually querying local audio CD to sell which price is between 10 and 20.*

```
continuous select cd_title, cd_price
from cd_to_sell
where cd_price is between 10 and 20;
```

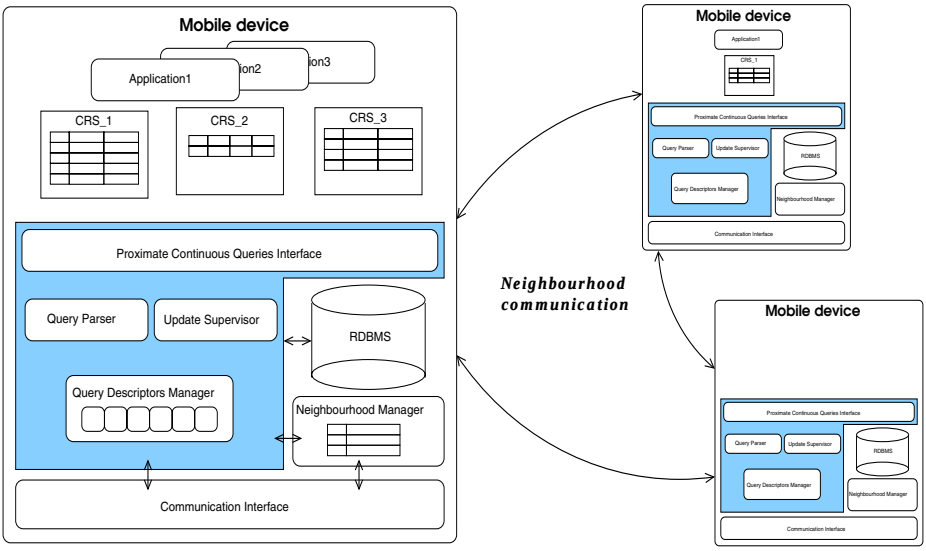


Fig. 1. Architecture of the PERSEND querying system

In SQL queries, data sources are identified by naming the involved tables and databases. In the querying model we consider, neighbouring tables can be involved in the queries a device processes. The system therefore has to know when to only consider local tables and when to distribute a query. For this purpose, we introduce the `vicinity` keyword. It has to be placed at the beginning of a consultation query, after the `continuous` keyword (if specified). It indicates that the following query, continuous or not, has to be distributed among all neighouring devices. We call such queries *proximate queries*. Query 2 extends the previous example by querying all neighbouring devices.

Query 2 *Continually querying proximate audio CD to sell which price is between 10 and 20.*

```
continuous vicinity select cd_title, cd_price
from cd_to_sell
where cd_price is between 10 and 20;
```

Now our SQL-based querying language is presented, let us study the architecture of the PERSEND system.

4.2 Overview of the PERSEND Architecture

Figure 1 presents the global architecture of the PERSEND querying system. PERSEND is based on a Relational DataBase Management System (RDBMS). Besides the neighbourhood manager, which provides applications with information on neighbouring devices, PERSEND is organized around four main components: the query interface, the

query parser, the update supervisor and the query descriptor manager. The remaining of this section is dedicated to their description.

The Proximate Continuous Query Interface. Queries, whatever their type, are transparently submitted by the way of the continuous query interface. Two primitives are currently provided. The first one, `executeQuery(Query- Text)`, is called to submit a query to the system. The type of submitted queries is determined by the query parser. Instantaneous local consultations are processed as usual. Once parsed, modification queries are transmitted to the update supervisor. Continuous queries are, as for them, inserted in the query descriptor list. According to the type of the submitted query, `executeQuery` returns a result set (instantaneous consultations of local data), a query status (data modifications) or a continuous query handler (continuous queries). Data currently matching a continuous query are stored in the CRS associated with the query. Handlers enable applications to access CRS associated with the continuous queries they have issued. They also provide a global identification of continuous queries with the *(DeviceID,CQueryID)* pair. The second primitive, `closeContinuousQuery (CQHandler)`, enables applications to terminate a given on-going continuous query and to close the associated CRS.

The Query Parser. The analysis of a query associates the query with a descriptor. Besides its type (`select`, `delete`, `insert` or `update`), a descriptor contains all available information on the query: its range (`local` or `proximate`), its duration (`continuous` or `instantaneous`) and the data it handles. Data handled by consultation queries are identified by *(db_name, table_name, field_name)* tuples. Descriptors associated with continuous queries are indexed, with their handler, in the descriptor list. Data handled by modification queries are coded in specific ways. Thus, an `insert` descriptor just provides the targeted table, identified by a *(db_name, table_name)* pair, and the row to be inserted. Descriptors associated with modification queries are transmitted to the update supervisor before the queries to be executed by the RDBMS.

The Update Supervisor. This component has to detect the repercussions that submitted modification queries (`insert`, `delete` and `update`) may have on on-going continuous queries. Given the descriptor of a modification query Q_m, it examines all continuous queries in the descriptor list in order to determine if the execution of Q_m interferes with the result of a continuous query CQ. This is achieved by computing the intersection between data handled by Q_m and local data which currently match CQ. If the computed set is not empty, the device having issued CQ has to be notified of the modifications to be performed on $CRS(CQ)$.

Remind the example presented in Table 1. We assume the user has sold the CD *Transformer*. He now wants to remove it from the table with Query 3.

Query 3 *Removing the 'Transformer' CD from the* `cd_to_sell` *table.*

```
delete from cd_to_sell
where cd_title = 'Transformer';
```

Consider that the continuous query CQ issued by a neighbouring device is still running (see Query 2). When Q_3 is submitted, the update supervisor computes the intersection between data that Q_3 handles ($Row\ \#2$) and those which locally match CQ ($Row\ \#2$, $Row\ \#3$). As the computed intersection ($Row\ \#2$) is not empty, and according to the type of the modification query (in this case, `delete`), the querying device has to be notified of the deletion of data locally identified by the (`cd_to_sell`, $Row\ \#2$) pair. This notification is performed by means of an update command.

The Query Descriptor Manager. This component manages the list of the continuous queries which are running in the device's vicinity. This list contains two kinds of descriptors: those associated with locally submitted continuous queries (proximate or not) and those associated with proximate continuous queries issued by neighbouring devices. A descriptor is removed from the list when an explicit call to `closeContinuousQuery` occurs. When a device leaves the neighbourhood, the system closes all the continuous queries this device has issued.

The Neighbourhood Manager. The aim of this component is to provide applications with an up-to-date list of neighbouring devices. This list is stored in the device's *neighbourhood table*. Each device is associated in the table with a unique handler and the time stamp of its insertion. Applications can access the neighbourhood table and read its content each time they require information about their physical vicinity. Those frequently requiring such information may issue redundant readings as the table remains unchanged between successive accesses. The neighbourhood manager therefore provides applications with a notification service. As they subscribe to the service, applications receive the current content of the table. Afterwards, they are notified each time a device leaves or enters the vicinity. The PERSEND querying system subscribes to the notification service to get information about its physical neighbourhood.

4.3 Managing Proximate Continuous Queries with PERSEND

We have presented the architecture of the PERSEND querying system. We now study how proximate continuous queries are managed: first, on the device having issued it and then on its neighbouring devices. Note that, save the communication issues, local continuous queries are managed in the same way than proximate ones are.

Locally Submitted Proximate Continuous Queries. A continuous query is submitted using the `executeQuery` function. As every query, it is transmitted to the query parser. If it is not syntaxically correct, `executeQuery` returns an error to the querying application. Otherwise, an empty CRS is associated with the continuous query and `executeQuery` returns a handler enabling the application to access the CRS.

The analysis of a query provides the system with its associated descriptor. Continuous queries' descriptors are indexed in the descriptor list. Then, PERSEND broadcasts to its neighbourhood the query associated with its descriptor. In the same time, it computes the set of local rows which matches the continuous query (by submitting the query

to the RDBMS) and inserts them in the associated CRS. As data sets associated with the query are received from neighbouring devices, they are merged to the CRS according to the semantics defined in Section 3.

When a device leaves the vicinity, the rows it has provided are removed from the CRS. A device entering the vicinity is sent all current proximate continuous queries which have been locally issued. When data modifications are performed, those interfering with the continuous query are detected by the update supervisor. The supervisor then generates the update commands to be performed on the CRS. Likewise, when update commands relevant to the query are received from a queried device, the corresponding CRS is updated according to the transmitted commands. Finally, the querying application can close the continuous query by calling `closeContinuousQuery`: the command is then broadcasted to the neighbourhood before the descriptor be removed from the descriptor list.

Remotely Submitted Proximate Continuous Queries. A device is involved in a remote proximate continuous query as soon as it is notified of the existence of such a query. Two cases have to be considered: a neighbouring device creates a new proximate continuous query, or a new device, currently running such a query, enters the vicinity. In both cases, the queried device receives the continuous query to be executed and its descriptor. Note that, in the second case, the queried device receives all the on-going proximate continuous queries issued by its new neighbour.

The descriptor of a received proximate continuous query is indexed in the descriptor list. The query is then executed on the local RDBMS and the result is returned to the querying device. In case the local execution of the query returns an empty set, no message is sent back.

When modifications are locally performed on data involved in, at least, one remote proximate continuous query, the update supervisor has to notify the querying device of it. For this purpose, it generates a message containing the suited update commands and broadcasts it to its neighbourhood. Finally, a remote proximate continuous query is stopped, and its descriptor removed from the descriptor list, when the querying device either leaves the vicinity or explicitly closes the continuous query by calling `closeContinuousQuery`.

5 Implementation Issues

A first prototype of the PERSEND querying system has been implemented. The experimentation platform we used is based on PocketPC PDAs running Windows CE 3.0 and equipped with 802.11b communication cards. Users' data are accessed by means of the Windows ADOCE 3.1 library.

The neighbourhood manager uses a simple discovery protocol, based on UDP sockets, in order to build and maintain the neighbourhood table. Devices announce their presence by periodically broadcasting a *Hello* message. When an announcement message is received, its sender is inserted in the local neighbourhood table associated with the current time stamp. If the sender is already in the table, the neighbourhood manager simply sets its associated time stamp to the current time stamp. When a fixed period

has elapsed since the last announcement of a neighbour, its entry is removed from the neighbourhood table. As devices constituting our platform are equipped with homogeneous communication facilities, we currenlty assume a symetrical discovery scheme (a device seeing a neighbour is also seen by this neighbour).

Each device runs a PERSEND server which uses the ADOCE interface to execute SQL queries. Communications between remote PERSEND servers are based on UDP sockets. As ADOCE internal features are not available, we have implemented our own query parser. Having no knowledge of the queried database structures, this parser is not able to associate fields defined in a query with their respective tables. Therefore, we assume users to prefix each declared field with either the name of the table it is issued or any defined alias (see Query 4).

Query 4 *Rewriting Query 2 to deal with the query parser's limitations.*

```
continuous vicinity select C.cd_title, C.cd_price
from cd_to_sell C
where C.cd_price is between 10 and 20;
```

Finally, we are implementing a basic continuous query viewer in order to experiment our system. The viewer enables users to submit all types of queries to the PERSEND server and displays the results of on-going continuous queries. Displayed data are periodically read from opened CRS.

6 Related Works

The PERSEND querying system considers the neighbouring devices as the only relevant data sources. So, the physical neighbourhood of a device can be seen as its current context. This notion of context is widely used in the pervasive computing area: pervasive systems aim to provide users with contextual services [7]. Since a few years, some of these studies have focused on neighbouring interactions in proximate environments. Some systems have been specifically designed in order to initiate casual meetings when mobile users meet. Thus, Proxy Lady triggers an alarm when a person within a predefined list is physically close enough [8]. When such a meeting occurs, Proxy Lady spontaneously provides the user with documents it has previously specified. Likewise, Proem performs some exchanges of users' profiles in order to initiate such encounters [9]. When a user-defined condition (such as mutual interests, common friends) is met, Proem triggers the action associated to the condition. The Side Surfer prototype was designed to enable spontaneous exchanges of relevant information between mobile users [10]. A user profile, based on the keywords used to describe personal documents stored on the mobile device, is automatically built by the system. During physical encounters, generated profiles make a fast discovery of mutual interests possible.

Some pervasive studies more particularly deal with data accesses in proximate environments. Thus, the SPREAD system defines a spatial programming model: data can only be accessed in the physical space associated with the device which manages them [11]. Data are published by means of tuples and are queried using some pattern tuples. By associating a physical space with each device, SPREAD provides a larger

model than the one PERSEND considers. However, compared to database systems, the tuple data model only makes it possible to publish basically structured information and to define very simple queries (conditions have to be expressed using equality operators only). Moreover, SPREAD does not deal with data storage issues. MoGATU is another system which aims to make proximate data accesses possible [12]. Managed data and submitted queries are defined by means of a semantic web language. The MoGATU system only makes it possible to run simple queries, that is, in a database model, queries involving data stored in a single table. Based on the profile of the user, and according to its current context, implicit queries can also be processed. However, and contrary to PERSEND, the MoGATU system allows queries to be routed to non-neighbouring devices. As devices can, by this means, access non-neighbouring data, the notion of physical neighbourhood is partially lost. Finally, MoGATU does not consider the storage issues.

The PeerWare system is designed to provide a middleware support for peer-to-peer interactions in mobile and ad hoc environments [13]. It enables mobile devices to share documents they store by means of a global data space. For this purpose, applications are provided with a set of basic primitives and a notification mechanism. Advanced features, such as continuous data access, have to be developed by application designers based on the provided primitives. Furthermore, since PeerWare is designed for both cellular and ad hoc networks, shared data spaces are not generated according to the physical neighbourhood of the involved peers.

In the database area, PERSEND is of course close to the studies on continuous queries. Besides works presented in Section 2.1, we can cite the Alert system [14]. Alert aims to build an active RDBMS based on a classical RDBMS. It defines the notion of *active table* which is an append-only table. *Active queries* can be run on active tables: they provide an append-only result set in which new relevant rows are added at the end. Such result sets are read using the *fetch-wait* primitive. This primitive is a blocking read: once the last row of the result set has been returned, the reading process is blocked until a new row is inserted in the result set.

Finally, the Microsoft ADOCE library enables users to open data sets which are dynamically linked to the queried tables [15]. When queried data are issued from a single table, the obtained data set behaves as a continuous result set by reflecting the updates performed on the data source. However, the library makes it possible neither to manage continuous result sets associated to join queries nor to define proximate result sets.

7 Conclusion

In this paper, we presented the design and the implementation of the PERSEND querying system. This system allows applications running in a proximate environment to define and access continuous result sets (CRS). These data sets can involve both local data and data stored by current neighbouring devices. The PERSEND system is based on a RDBMS and the continuous result sets are expressed using the SQL querying language. We defined, in this scope, new semantics for SQL agregation functions which are suited to proximate environments. We also introduced two new keywords making

the definition of continuous and proximate queries possible with SQL. The PERSEND system associates each continuous query with a CRS which can be read by the querying application. Managed CRS are kept up-to-date by supervising the data updates performed on the neighbouring data sources. In order to demonstrate our system, we have implemented a first prototype of the querying system and a continuous query viewer application is currently developed.

We are now investigating the design of additional features. Thus, for efficiency reasons, the PERSEND system may include a mechanism enabling applications to share a same continuous result set when they run the same continuous queries. Moreover, in order to make result sets easily readable, users consulting the query viewer application may want displayed rows to be sorted according to their time of presence in the data set. For this purpose, we consider associating a time stamp with each CRS row. Likewise, in order to be aware of the last modifications, applications currently have to periodically scan by themselves the content of the CRS they have opened. Just as for the neighbourhood table, this scheme is not satisfactory: applications can miss important updates and perform some unnecessary readings. We therefore plan to associate CRS with a notification mechanism enabling a querying application to be warned when its CRS is updated. This mechanism can be provided by means of a blocking event-based primitive.

References

1. J. Jing, A. Helal, and A. Elmagarmid. Client-Server Computing in Mobile Environments. *ACM Computing Surveys*, 31(2):117–157, June 1999.
2. J. Haartsen, M. Naghshineh, J. Inouye, O. Joeressen, and W. Allen. Bluetooth: Vision, Goals, and Architecture. *Mobile Computing and Communications Review*, 2(4):38–45, October 1998.
3. D. Terry, D. Goldberg, D. Nichols, and B. Oki. Continuous Queries over Append-Only Databases. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 321–330, June 1992.
4. P. Bonnet, J. Gehrke, and P. Seshadri. Querying the Physical World. *IEEE Personal Communications*, 7(5):10–15, October 2000.
5. Deborah Estrin, Ramesh Govindan, John S. Heidemann, and Satish Kumar. Next century challenges: Scalable coordination in sensor networks. In *Mobile Computing and Networking*, pages 263–270, 1999.
6. A. P. Sistla, O. Wolfson, S. Chamberlain, and S. Dao. Modeling and Quering Moving Objects. In *Proceedings of the 13th International Conference on Data Engineering (ICDE'97)*, pages 422–432, April 1997.
7. G. Chen and D. Kotz. A Survey of Context-Aware Mobile Computing Research. Technical Report TR2000-381, Department of Computer Science, Dartmouth College, 2000.
8. Per Dahlberg, Fredrik Ljungberg, and Johan Sanneblad. Proxy Lady: Mobile Support for Opportunistic Interaction. *Scandinavian Journal of Information Systems*, 15, 2000.
9. G. Kortuem, Z. Segall, and T. G. Cowan Thompson. Close Encounters: Supporting Mobile Collaboration through Interchange of User Profiles. In *Proceedings of the First International Symposium on Handheld and Ubiquitous Computing (HUC'99)*, pages 171–185, September 1999.

10. D. Touzet, J-M. Menaud, M. Banâtre, P. Couderc, and F. Weis. SIDE Surfer: Enriching Casual Meetings with Spontaneous Information Gathering. *ACM SigArch Computer Architecture Newsletter*, 29(5):76–83, December 2001.
11. P. Couderc and M. Banâtre. Ambient computing applications: an experience with the SPREAD approach. In *Proceedings of the 36th Annual Hawaii International Conference on System Sciences (HICSS'03)*, pages 291–299, January 2003.
12. F. Perich, S. Avancha, D. Chakraborty, A. Joshi, and Y. Yesha. Profile Driven Data Management for Pervasive Environments. In *Proceedings of the 13th International Conference on Database and Expert Systems Applications (DEXA'02)*, pages 361–370, September 2002.
13. G.Cugola and G. P. Picco. PeerWare: Core Middleware Support for Peer-to-Peer and Mobile Systems. Technical report, Dipartimento di Electronica e Informazione, Politecnico di Milano, May 2001.
14. U. Schreier, H. Pirahesh, R. Agrawal, and C. Mohan. Alert: An Architecture for Transforming a Passive DBMS into an Active DBMS. In *Proceedings of the 17th International Conference on Very Large Data Bases (VLDB)*, pages 469–478, September 1991.
15. Microsoft ADOCE 3.1 documentation. Available at http://msdn.microsoft.com/library/default.asp?url=/library/en-us/adoce31/html/adowlcm.asp.

Author Index