September 8-11 2003,
Udine (Italy)

MOBILE HCI 03

Mobile HCI 03 Workshop on Mobile and Ubiquitous Information Access

Draft Proceedings

Edited by
Fabio Crestani, Mark Dunlop and Stefano Mizzaro

Papers

Demonstrations & short talks

Papers

Mobile Access to the Físchlár-News Archive

Cathal Gurrin
Centre for Digital Video Processing
Dublin City University
Ireland
353 1 700 5234
cgurrin@computing.dcu.ie

Alan F. Smeaton
Centre for Digital Video Processing
Dublin City University
Ireland
asmeaton@computing.dcu.ie

H. Lee, K. McDonald, N. Murphy, N. O'Connor, S. Marlow
Centre for Digital Video Processing
Dublin City University
Ireland

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 the completion of the system.

Categories and Subject Descriptors
H.5.1[**Information Interfaces and Presentation**]: Multimedia Information Systems – *Audio input/output, Evaluation/ methodology, Video.*

General Terms
Video, Management, Human Factors, Mobile.

Keywords
Mobile Access, Information Retrieval, Digital Video, News Retrieval, Video Analysis, Collaborative Filtering, Interface Design.

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.

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:

1. Automatic news story segmentation.
2. Personalisation by means of news story recommendations tailored to the interests of individual users.

In this paper we describe mobile access to the Físchlár-News system and the soon to be deployed fully-automated version of the system. 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

MobileHCI '03, September 8, 2003, Udine, Italy.

[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.

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 a transitional system that we developed to kick-start the fully-automatic system and 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, 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-TREC2002 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-TREC2002 system was developed for our participation in the interactive search task in the annual activity at the TREC Video Track in 2002 [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. Figure 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.

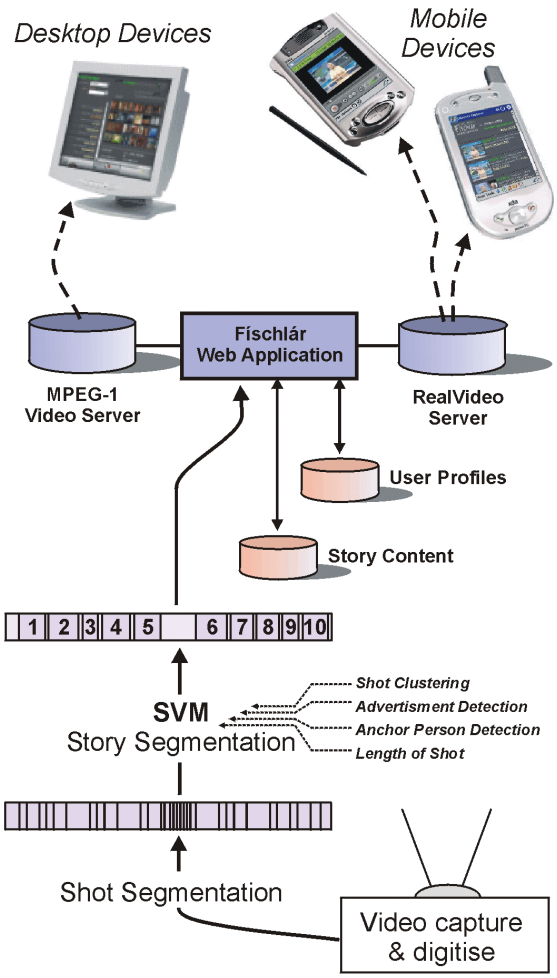


Figure 1. Architecture of Físchlár-News.

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 Figure 1. The inclusion 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 of an archive of news content 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 below.

2.1.1 Browsing News by News Program

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

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

When presented with a listing of news stories there are two options available to the user, 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 (Figure 3).

Figure 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 text[2] that is associated with that story, as shown in Figure 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.

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

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

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 allowing user queries against these textual descriptions of each story. This facilitates content-based retrieval of news stories based on textual queries. For example, in Figure 5, a query "SARS virus" has been presented to the Físchlár-News system.

[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.

Figure 5. Content searching the news archive.

The results of the search (23 news stories) are presented on the right side of the screen in decreasing estimated 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 is 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 Figure 6 which shows a listing of related stories to a SARS story).

Figure 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 Figure 7 below where a user has rated a news story as being one that the user likes (a small 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.

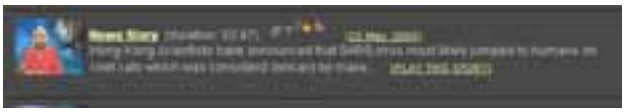


Figure 7. The five point story rating scale.

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, based on previous usage and provision of explicit story ratings by that user are used as one of the two primary access mechanisms for the mobile version of the Físchlár-News system, which we now describe.

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.

Figure 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.

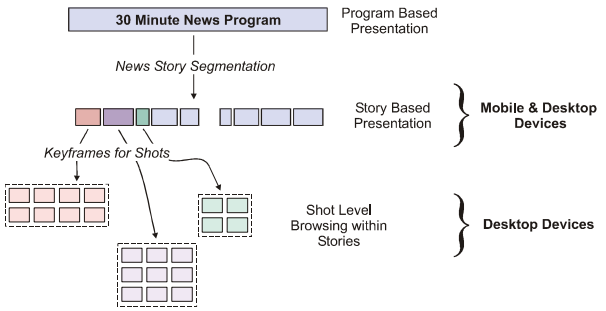


Figure 8. A logical breakdown of news programs.

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 Figure 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, 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 (Figure 10) which causes the story to be streamed in RealVideo format.

Figure 9. Personalised story recommendations.

Figure 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 thinks would be of interest to the user and 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 (Figure 11). Upon selecting a news programme, the user is presented with a listing of news stories from within that programme (Figure 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.

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

Figure 12. Story listing on a specific date.

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

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 are currently testing (and soon integrating) 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 some previous research in this area, upon which we now report.

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

[3] Unfortunately not all closed caption broadcasts contain such manual markings, RTE1 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.

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

Source of 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)
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 (person, location & organisation), 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

chaining analysis. Using only closed 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]. In the Físchlár-News system, we use an SVM (Support Vector Machine) to combine various evidences automatically extracted from the video content. Table 1 (above) shows a summary of analysis methods and combination methods used in six news video retrieval systems, including Físchlár-News.

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. We utilise 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 Figure 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[4] 2003, which will give us a better indication of SVM performance. Currently the automatic segmentation system is operational in a prototype system, which will be incorporated into Físchlár-News in September 2003.

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 story. 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.

[4] TREC Video Track, a special "track" in the annual TREC benchmarking exercise dedicated to video retrieval related tasks.

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 an XDA 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.

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 have built and are currently using a manually segmented version of Físchlár-News to kick-start the eventual, fully-automated mobile system. In order to support story-based access to news content, prior to having the automatic news story segmentation tool incorporated into the operational system, the stories had to be manually extracted from the news program.

Since April 2003, recorded daily news programmes have been 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 its 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:

- Start-time and end-time,
- A representative keyframe,
- Representative text to describe the story, usually extracted from the closed caption text.

While this initial manual segmentation just described serves to start collection of initial user ratings of news stories required for collaborative filtering, the automatic method of accomplishing story segmentation is operational on a prototype system and will be made fully operational in September.

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-July 2003) we have received over 13,000 individual story recommendations from these users for 958 news stories. The benefit of this is that in September 2003, when the fully automated Físchlár-News system goes live, it will be able to generate recommendations of stories based on collaborative filtering (for mobile devices) on an individual user basis and present these stories when a user accesses the Físchlár-News system on a mobile device.

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.

6. ACKNOWLEDGMENTS

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.

7. 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/t2002v/t2002v.html. Last visited July 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, Gordon Dodds and Chris Tweed
Virtual Engineering Centre, Queen's University Belfast, Cloreen Park, Malone Road, Belfast, BT9 5HN, Northern Ireland, UK
w.mai@ee.qub.ac.uk

ABSTRACT
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. 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.

Keywords
Building recognition, mobile devices, image processing

1 INTRODUCTION

With the increased availability and advanced features of 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]. In [3], project improves the GPS accuracy using the GPS data, orientation data, image, and also the Hough Transform method. The hardware system in [3] is quite similar with ours except that our system is mobile and portable. Moreover, 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 are several digital city projects [1, 6, 7, 8, 14] concerned with cityscape and city model 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 application [16], etc. However this project is trying to set up a system to help tourists identify buildings on the road and also provide immediate information in real-time. Image processing for object recognition and self-develop program for different parts of applications are two 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 in University Square, Belfast 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 stand, 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 on the PDA 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 about the object. 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 main building at Queen's University. 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, like the GPS data, orientation data and the image. The use of images removes some of the problems of low GPS 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 identify 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 good way to keep the model updated. A more detailed flow chart for the user operation sequence is shown in section 2.3.

The novel idea for this project is the integration of many different hardware devices. All these devices are portable and relatively low-cost. It is available for travellers carrying around, a PDA camera measuring 3.4x5.2x2.0 inches and weighing only 0.5kg. 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 sensor attached on the PDA) of the image as well. With these data, image processing for building recognition becomes possible.

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.

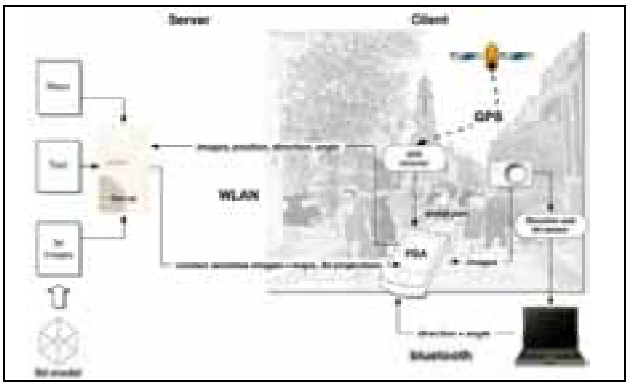


Figure 1: system components diagram.

2.1 Client Side

The client side is the portable PDA system. The system includes an *iPAQ 3870*, *NexiCam* PDA camera with resolution 600x800, orientation sensor and GPS receiver. Because of a current limitation of the PDA development, it 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. For this reason, the USB interface sensor is not able to connect to the Pocket PC. A laptop is used to receive the data from the sensor and Bluetooth supports 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.

After preliminary research, the iPAQ 3870 was found to be the only solution when this project started. GPS data can be improved to be DGPS data after software modification. In our application, DGPS data can be accurate to 2m.

2.2 Server Side

2.2.1 3D City Model

The selected city model will be built from site surveys conducted using Cyrax2500 laser scanner, CloudWorks and *3DMax* software. Digital images will be used to record details and texture of the buildings. Images from the PDA users will populate this database updating. 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 clients. This model image provides the reference for building recognition. 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. However the image texture mapping is not done on this model.

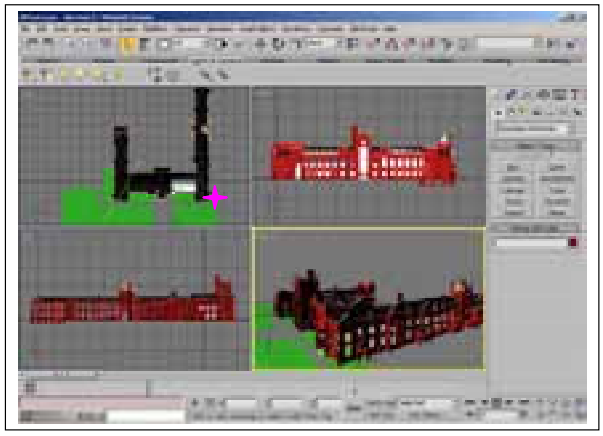


Figure 2: City model in *3DMax* environment.

2.2.2 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 from 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.

2.2.3 Applications

Some applications will be available on the server.

The first 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 second 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 processing and some methods, like line detection and segmentation are used for building recognition.
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 Network

2.3.1 Bluetooth

A Bluetooth SDK from WIDCOMM has been used to develop Bluetooth applications for communication between the PDA and laptop. The SDK was compiled with Microsoft Visual C++. 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 every second and the PDA transfers the data to the laptop through Bluetooth. This data will be synchronized with the orientation sensor data. Sensor data are also received every second. When the user takes a picture and sends it to the server for processing, Bluetooth will instruct the laptop to send a corresponding GPS and orientation sensor data.

2.3.2 WLAN

Several WLAN access points have been set up in 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. A WLAN card is used on the PDA to access the Internet

2.3.3 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

First of all, user must run the self-develop the program at the background before they start this project system. This program is mainly designed to read the serial port data from the GPS receiver and send it to the laptop through Bluetooth every 2 seconds, and the most recent (about 5 minutes) data will be saved in the laptop and will be sent to PDA whenever it is requested. And it also keeps polling the PDA if it takes any picture. Process starts as depicted in flowchart, shown in Figure 3.
User opens the camera application on PDA. When the user press the button to take a picture, background program record the time for this action After he is satisfied and tries to save the picture in PDA, background program will request the corresponding GPS and orientation data from the laptop to the PDA and send them to the server. When the data arrive, server first runs 3DMax to generate a picture from the same position in the 3d model (as showed in Figure 5). And then identify the income PDA picture with the reference image. If these two pictures are identified to be the same building, server will provide information for the building together with the user location. Otherwise, a notice "object unidentified" will be sent to PDA user. But user location in a 2D map is still available for user.

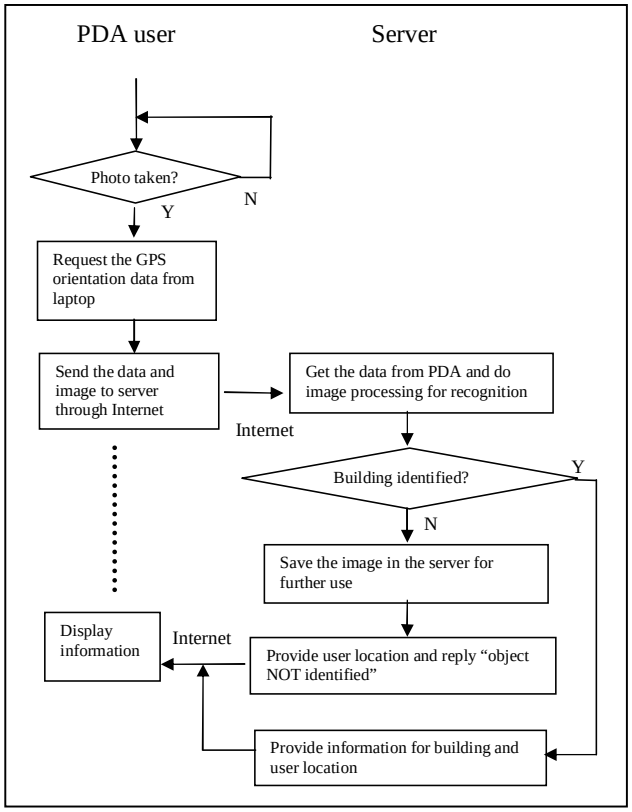


Figure 3 Flowchart for user

3 OBJECT RECOGNITION METHODS

There are two 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 of 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.

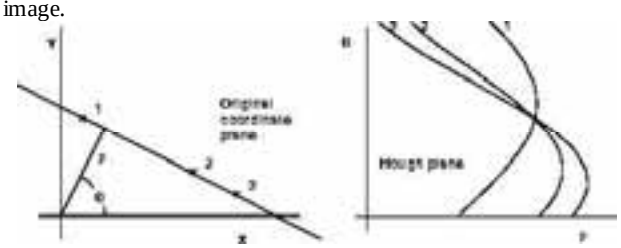


Figure 4 Hough/Radon Transform

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 included using different filtering and colour space conversions. The method for post-processing is that we set the difference between the parameters of the lines we found in the image must not be within the areas we defined, e.g. the theta must be more than 20º and *rho* to be more than 30 pixels. If parameters are within these areas, we consider them as the same lines (results are shown in Figure 7).

3.2 Segmentation

Clustering is one of the fundamental methods for image segmentation [5, 10]. It is designed as following steps [5]:

1. Data representation: In this project, data is represented as probability or distributional data. Then every block with size of 6x8 pixels is grouped and used to generate a Gaussian model based on its one feature value, e.g. Hue or Intensity. Each block is represented as the mean value and standard deviation as one input data.
   Image is represented in HSV space (Hue, Saturation and Intensity) provides a separate channel for hue, saturation and intensity information, from which texture features can be extracted.
2. Modelling: this step is to choose a method to formally characterize interesting and relevant cluster structures in data set. Here, K-Means clustering is applied. It is a least squares partitioning method that divide a collection of objects into K groups.
3. Optimization: apply EM (Estimation - Maximisation) algorithm.
   E step: K-Means is applied as in step 2. It is a least squares partitioning method that divide a collection of objects into K groups.
   M step: maximise its cost function. The likelihood maximization step estimates the mixture parameter, centres and standard deviation.
4. Validation: there is nothing done for this at this stage. It is necessary to find some ways to validate the result in the future.

4 RESULTS

4.1 City Model Image

The model image is treated as a reference. So, generating the correct image is very important to this project. Figure 5 shows the image (on the right bottom corner) rendered by a *3DMax* script. On the top right corner is the script window. This script sets the camera to the correct position and generates an image that is very similar to the one taken from the PDA camera (Figure 5).

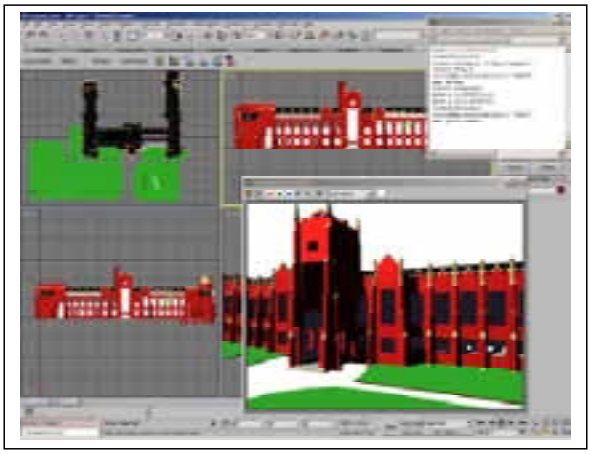


Figure 5: Image generation from city model.

Figure 6: User image from client PDA camera (600x800 resolution)

4.2 Radon Transform

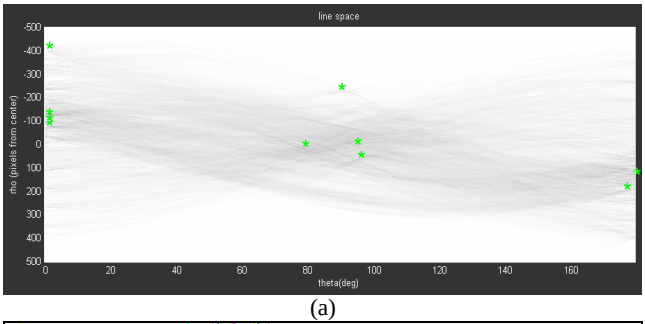
Figure 7 shows the lines found before and after the post-processing, which was mentioned in Section 3. Figure 7(b) contains more errors in vertical line detection (inside the oval in long dash). This is because around those edge areas, the dots are very dense and noisy (caused by the pattern in the real 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(c) are more reasonable. Figure 8 shows the lines in the model image after modification. You can see there are some difference between Figure 7(c) and Figure 8. This is caused by the pattern on the building and also the accuracy of the orientation and GPS data. However, as we will do the texture mapping on the city model and modify the GPS to DGPS (Differential GPS) data in the near future, this result will be improved afterwards.
Table 1 displays the parameters for the user image before and after post-processing and the city model image after modification. In this table, each couple of *rho* and *theta* defines a peak in Hough space (shown as the starts in Figure 7 (a)). In image space, this peak represents a line as showed in Figure 7 and Figure 8.

In the parameter data for Figure 7(b), Line 02 (67, 95) and Line 06 (66, 95) should be the same line. This error is caused by the dense and noisy dots in those areas. Also, line4 and line8, and line7 and line9 have the same problem. However, in the second column represented the line in Figure 7(c), Line 07, Line 08 and Line 09 are not there anymore.

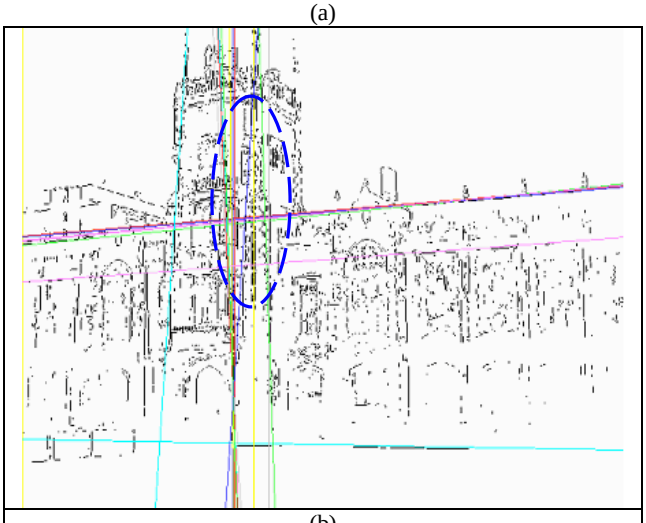
	Lines in Figure 7(b)		**Lines in Figure 7(c)**		**Lines in Figure 8**	
	rho	*theta*	*rho*	*theta*	*rho*	*theta*
Line 01	-398	0	-398	0	99	92
Line 02	67	95	67	95	-176	88
Line 03	-222	89	-222	89	44	179
Line 04	-71	0	-71	0	210	2
Line 05	-91	0	-91	0	-137	1
Line 06	66	95	-117	0	277	2
Line 07	-117	0	8	94	22	0
Line 08	-74	2	200	176	-158	0
Line 09	-119	0	19	78	287	176
Line 10	8	94	139	179	346	3

Table 1 Line parameters

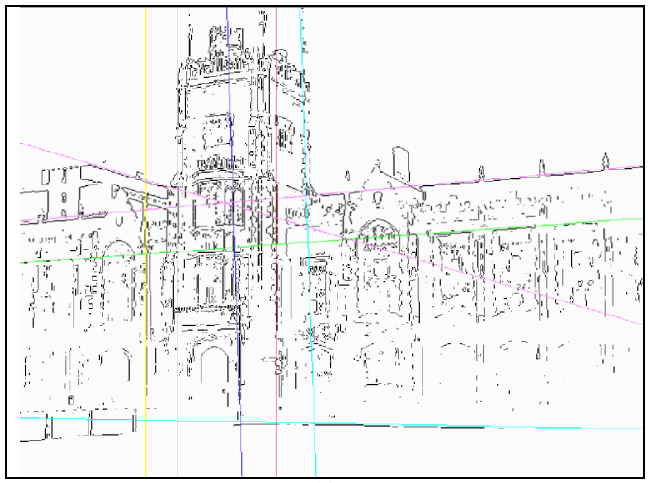
The line data for Figure 7(c) and 8 do not match very well. The most likely reason is that the image from the city model lacks texture. Note that the orientation data obtained are not accurate enough. This method will be used in conjunction with the segmentation approach and modelled data will be compared with sampled images in order to determine the usefulness of each method.



(a)



(b)



(c)

Figure 7: Lines detected in the user-supplied images using the Radon Transform (b) before post-processing and (c) after post-processing. Note pre-processing has been applied on both figures. (a) shows the peaks of (c) in Hough space

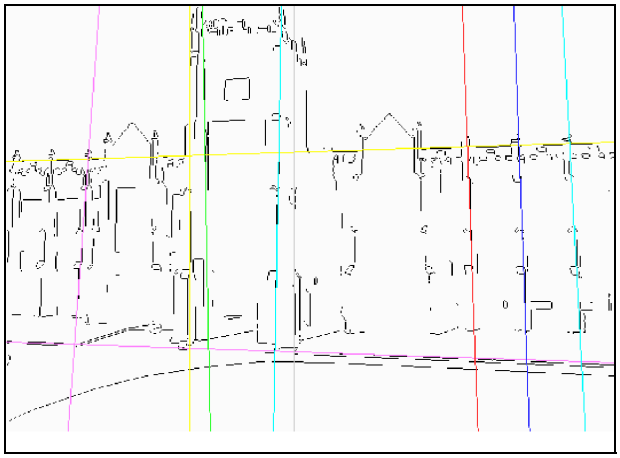


Figure 8: Lines in the city model image detected using the Radon Transform

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 9).

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, which means, this experiment is going to

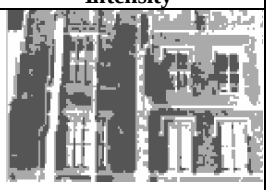
segment the picture into 3 groups by it Hue or Intensity values. In this case, K-Means is applied to do the estimation. After that, use M-step for optimization. Results of the segmentation are showed in Figure 9.

Figure 9 (a) shows the PDA camera picture (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. Figure 9 (c) shows the average and standard deviation of each of these groups. These results are displayed in Figure 10. The centres for Figure 10 are from two different red-brick buildings, but even here, it can be seen that the intensity values of the building show significant differences from the other. Work is presently developing appropriate limits for the segmentation sets.

In Figure 10, the cross, dot, and start respectively represent the centres 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. For another word, these centres are in some way representing the feature of the image.

Picture4	Picture5	Picture9

(a)

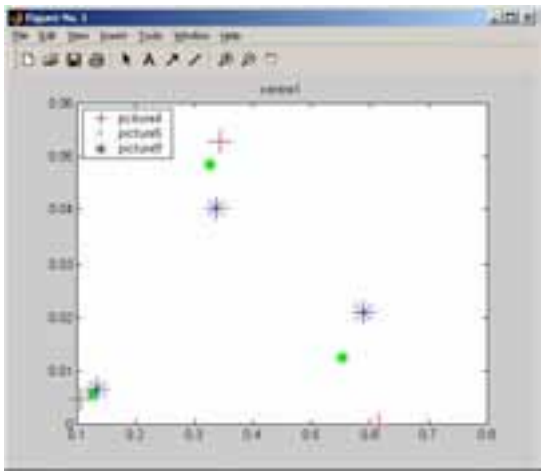
	Hue	Intensity
Picture 4		
Picture 5		
Picture 9		

(b)

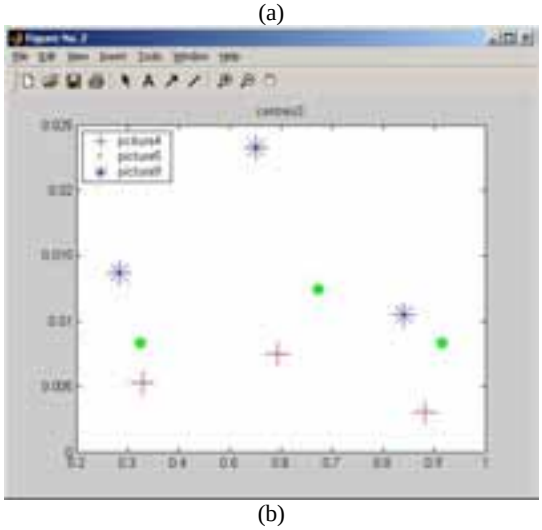
Picture 4	centres1 = 0.1077 0.0046 0.3444 0.0527 0.6160 0.0149	centres3 = 0.8831 0.0030 0.5934 0.0075 0.3293 0.0053
Picture 5	centres1 = 0.1286 0.0056 0.3282 0.0483 0.5537 0.0123	centres3 = 0.9162 0.0083 0.3261 0.0083 0.6748 0.0124
Picture 9	centres1 = 0.1353 0.0065 0.3386 0.0403 0.5901 0.0209	centres3 = 0.2836 0.0137 0.5516 0.0233 0.8409 0.0105

(c)

Figure 9 Segmentation for 3 different pictures



(a)



(b)

Figure 10: Centres for each group
(a) shows the centres from Hue segmentation and (b) shows the centres from Intensity segmentation.

4.4 GPS Data receive and Bluetooth transfer

Figure 11 shows the windows to display GPS data for PDA (shown as B in the figure) and laptop (showed as A), which enables the PDA to obtains the serial data from the GPS receiver (device D) every two second and passes it to the laptop. Laptop keeps the data in the memory and updates it every 5 minutes. Program is developed using eMbedded Visual C++ and WIDCOMM Bluetooth SDK. Device C in the figure is the Bluetooth adapter for laptop, which enables the laptop to communicate with the PDA in Bluetooth.

Figure 11: GPS data display on Laptop from PDA

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 travellers. 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 cards. Two methods for object recognition have been described and improvements have been discussed. A self-develop program has been shown in Section 4.4.

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. This causes two problems. First is that we need to use a laptop in this project. It makes the system heavier and not portable. Second is that it is not possible for us to debug the serial port as the serial port and the synchronization cradle are sharing the same port.
- The development tool doesn't fully support HOOK function. And it makes more difficult to catch the application event and combine the background program with active program, like the camera application.
- 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 self-develop program, e.g. the management between orientation data and GPS data and how to catch the "take picture" action
- Applying texture mapping to the city model
- Applying segmentation for building recognition
- Automating of the whole system

6 ACKNOWLEDGEMENTS

The authors wish to acknowledge the financial support of the Virtual Engineering Centre, Queen's University of Belfast, (www.vec.qub.ac.uk).

7 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, Incorporated, 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, Feb 2002.

[10] Puzicha J., Hogmannm T., Buhmann J. M., 1999. Histogram clustering for unsupervised segmentation and image retrieval, Pattern Recognition Letters, 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, 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

Spoken versus Written Queries for Mobile Information Access

Heather Du
Dept. of Computer and Information Sciences
University of Strathclyde
UK G1 1XH
heather@cis.strath.ac.uk

Fabio Crestani
Dept. of Computer and Information Sciences
University of Strathclyde
UK G1 1XH
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 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 Retrieve 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 procedural. 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 modality 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 ob-served 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. Seven of our participants members 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. There would be no language barriers for them to understand and formulate their information needs in English.

3.2 Text collection

Table 1: An example of 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.

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.

3.3 Experimental procedural

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 topics in text form, which were extracted from TREC topics. 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.

4. Experimental results & 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.

Table 2: Characteristics of WRITTEN queries

Data set	q1-q120
Number of queries	120
Unique terms in queries	328
Average query length (with stopwprds)	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

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 were very small. This was because that 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.

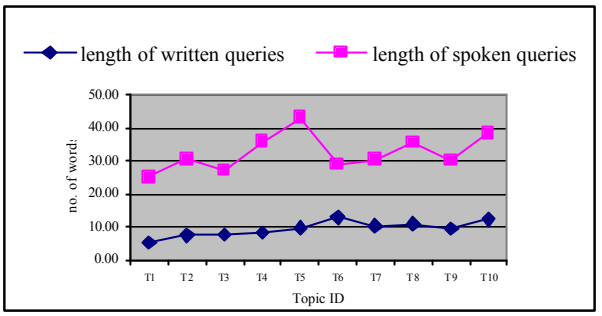


Fig. 1 Average length of spoken and written queries per topic

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 a case for every topic persistently. This was exactly what we would expect 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."

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.

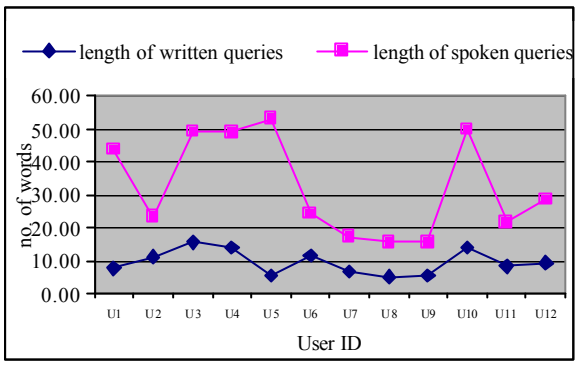


Fig. 2 Average length of queries per user

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 practise 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.

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-formulat e 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.

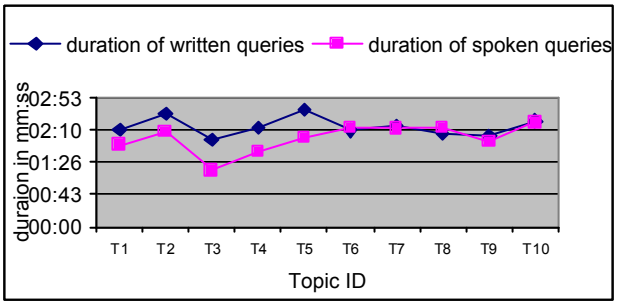


Fig. 3 Average duration of queries per topic

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 any 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.

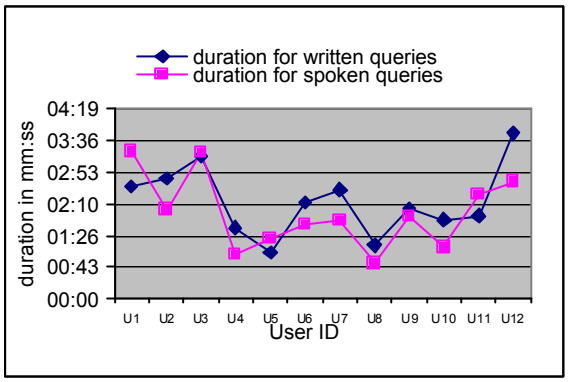


Fig. 4 Average duration of queries per user

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.

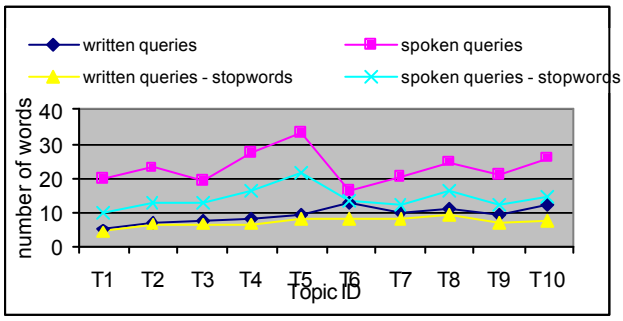


Fig. 5 Average length of queries across topics

As we can see from above figure, the line for spoken queries is consistently on top of the one for the written queries; after stopword removal, each of them are 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 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 this case in our experiment, 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.

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.

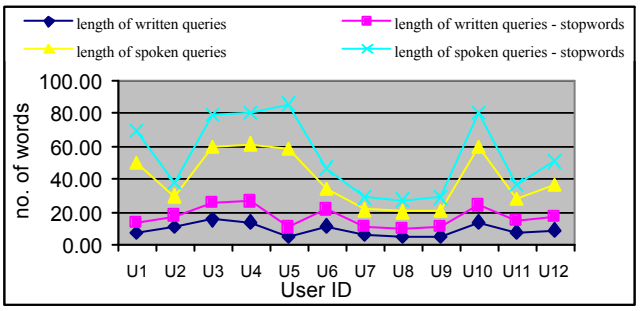


Fig. 6 Average length of queries per user

5. Conclusion & 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 come to the conclusion that spoken queries as a means of formulating and inputting information needs are utterly feasible.

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.

6. 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.

7. REFERENCES

[1] Cool, D., 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.

Aspect-Based Adaptation for Ubiquitous Software

Arturo Zambrano
LIFIA - UNLP
Argentina
arturo@
sol.info.unlp.edu.ar

Silvia Gordillo
LIFIA - UNLP
Argentina
gordillo@
sol.info.unlp.edu.ar

Ignacio Jaureguiberry
LIFIA - UNLP
Argentina
jauregui@
sol.info.unlp.edu.ar

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 system is a complex task, since a lot of concerns could get mixed with the application's core functionality. *Aspect-Oriented Programming* (AOP) [6] arises as a promising tool in order to design and develop ubiquitous systems, 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.

Keywords

Ubiquitous Software, Aspect-Oriented Programming, Adaptation, Context Awareness

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 modelled and the way 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 several 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 an example. The next section presents an analysis of advantages and disadvantages of this approach. Finally, we state our conclusions.

2. ADAPTATION IN UBIQUITOUS COMPUTING

An ubiquitous computing system consists of (a) a (possibly heterogeneous) set of computing devices; (b) a set of supported tasks; and (c) some optional infrastructure (e.g., network, GPS location service) the devices may rely on to carry out the supported tasks. [10]

Several approaches have been proposed to construct ubiquitous software artifacts. As expressed in [1] 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 are not aware of it.

To adapt system's behaviors 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]: *"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].

3. ASPECT-ORIENTED PROGRAMMING

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 [3]: *"...existing software formalisms support separation of concerns only along a* predominant dimension *neglecting other dimensions... with negative effects on reusability, locality of changes, understandability..."*. These secondary dimensions correspond to cross-cutting concerns. This is specially applicable to ubiquitous software, where a lot of dimensions are present.

Aspect-Oriented Programming (AOP for short) [8] is the technology 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 paramenter passed to the `printf` function).

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 such as those defined in [11]:

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 Advices define code that is executed when a point-cut is reached.

The program whose behavior is affected by aspects is usually called *base program*. To indicate where the code of the aspects has to be executed, it is necessary to define the join-points and point-cuts. The aspect's code is composed of advices and the point-cuts where those advices must be applied. Advices 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. These aspects contain the code to handle the concerns. Since the concern is a cross-cutting one, it is necessary to apply the behavior 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 advices to the proper point-cuts.

Asit will be shown in the next sections, we have used these AOP concepts to adapt the behavior of an ubiquitous system to runtime environment.

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 models 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 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 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 are a lot of servers which provide tourism information to the end user. 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 memory amount, 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.

4.1.1 Identifying System's Concerns

The system's functionality can be sumarize 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 modelled as aspects? Which of them as objects? How is their behavior 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 behavior 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.
- 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 [3][2] where we found the system's core, or the application itself. Other dimensions correspond to those concerns that have some effect on the predominant one. Since these concerns modify system's behavior 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 point + advices) are used as *glue* to attach the adaptive behavior in a seamless way.

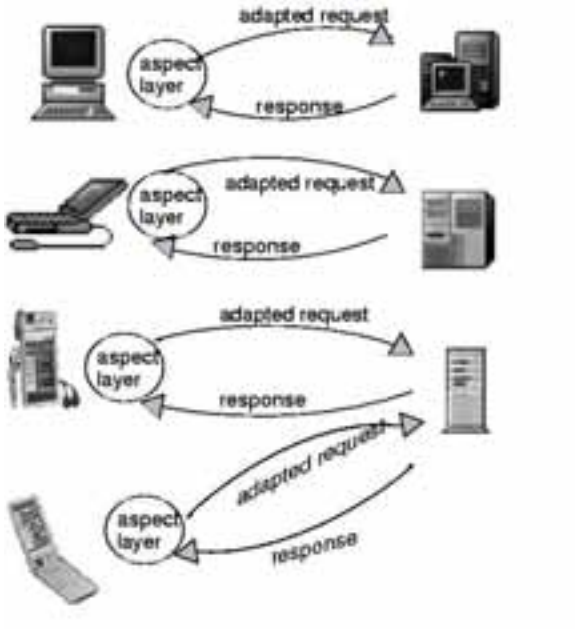


Figure 1: The Aspectual Layer adapts client's requests

4.1.2 Modular Division of System's Functionality

We will focus on the client-side which has to provide pervasive features. As far as this work 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 `Context` model. This is an standard object model which holds information about the current system's environment. 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 `request` word. Since *requests* are defined as *point-cuts*, each invocation to those messages is intercepted and automatically adapted by the aspectual layer. 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.

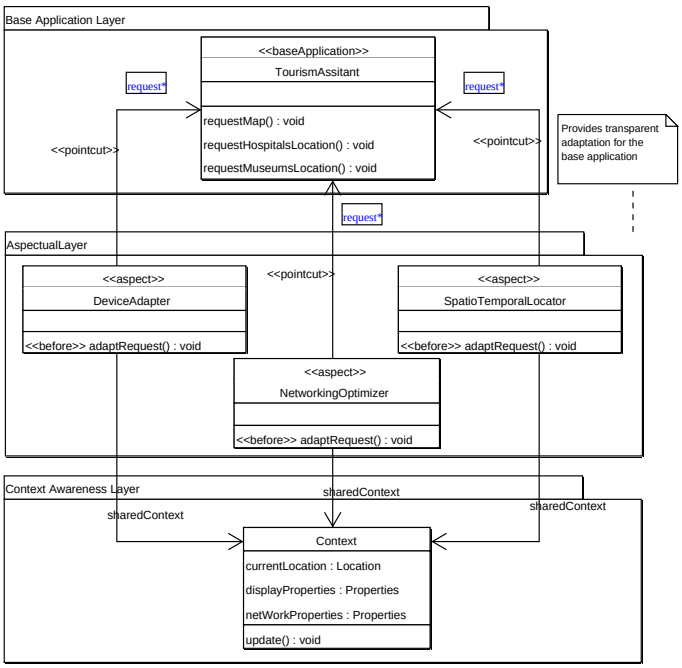


Figure 2: Simplified Client-Side Architecture

The second layer is the *Aspectual Layer*, which contains the adaptive behavior, i.e., base application's behavior 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 behavior to the current context in runtime. They constitute an adaptation layer that provide a completely transparent means to obtain this adaptive behavior. 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.

The 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 [7]. 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 behavior of base programs are included in the aspectual layer, which is invisible to them.
- Different concerns regarding ubiquity can evolve independently from one another.
- Since the context representation is stored at 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 *concern* modelled through aspects.

6. CONCLUSIONS

In this work we have analyzed how information flow can be affected and adapted by the runtime context in 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.

7. REFERENCES

[1] S.-W. Cheng, D. Garlan, B. Schmerl, J. Sousa, B. Spitznagel, P. Steenkiste, and N. Hu. Software architecture-based adaptation for pervasive systems.

In *Lecture Notes in Computer Science*. Carnegie Mellon University, 2002.

[2] S. Herrmann and M. Mezini. On the need for a unified mdsoc model: Experiences from constructing a modular software engineering environment. In *Proceedings OOPSLA 2000*. ACM-Press, 2000.

[3] S. Herrmann and M. Mezini. PIROL: A case study for multidimensional separation of concerns in software engineering environments. In *OOPSLA*, pages 188–207, 2000.

[4] G. Kappell, B. Prll, E. Kimmerstorfer, W. Schwinger, and T. Hofer. Towards a generic customisation model for ubiquitous web applications. In *2nd International Workshop on Web Oriented Software Technology. Proceedings*. Springer Verlag, 2002.

[5] G. Kappell, B. Prll, W. Retschitzegger, and W. Schwinger. Customisation for ubiquitous web applications. In *Int. Journal of Web Engineering and Technology (IJWET), Inaugural Volume, Inderscience*, volume 2299. Publishers 2003, 2002.

[6] G. Kickzales, E. Hilsdale, J. Hugunin, M. Kersten, J. Palm, and W. G. Griswold. Aspect oriented programming: Introduction. *Communications of the ACM*, 44(10):29–32, 2001.

[7] G. Kiczales, E. Hilsdale, J. Hugunin, M. Kersten, J. Palm, and W. G. Griswold. An overview of AspectJ. In J. L. Knudsen, editor, *Proc. ECOOP 2001, LNCS 2072*, pages 327–353, Berlin, June 2001. Springer-Verlag.

[8] G. Kiczales, J. Lamping, A. Mendhekar, C. Maeda, C. Lopes, J.-M. Loingtier, and J. Irwin. Aspect-oriented programming. In M. Akşit and S. Matsuoka, editors, *11th Europeen Conf. Object-Oriented Programming*, volume 1241 of *LNCS*, pages 220–242. Springer Verlag, 1997.

[9] R. Pawlak, L. Duchien, G. Florin, F. Legond-Aubry, L. Seinturier, and L. Martelli. A uml notation for aspect-oriented software design. In *AO modeling with UML workshop at the AOSD 2002 conference. Proceedings*, 2002.

[10] D. Salber, A. Dey, and G. Abowd. Ubiquitous computing: Defining an hci research agenda for an emerging interaction paradigm: Tech. Technical report, GVU, Georgia Tech, 1998.

[11] A. B. Tzilla Elrad, Robert E. Filman. Aspect oriented programming: Introduction. *Communications of the ACM*, 44(10):29–32, 2001.

PERSEND: Enabling Continuous Queries in Proximate Environments

David Touzet
IRISA/INRIA
Campus de Beaulieu
35042 Rennes
France
dtouzet@irisa.fr

Frédéric Weis
IRISA/Univ. Rennes 1
Campus de Beaulieu
35042 Rennes
France
fweis@irisa.fr

Michel Banâtre
IRISA/INRIA
Campus de Beaulieu
35042 Rennes
France
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.

Keywords

Mobile computing, Wireless communications, Databases, Continuous queries, Proximate interactions

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 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.

2.2.1 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.

nu_cd	cd_title	cd_price
1	War	8
2	Transformer	16
3	Animals	20
4	Kind Of Blue	32

Table 1: The cd_to_sell table

2.2.2 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).

2.2.3 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;
```

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.

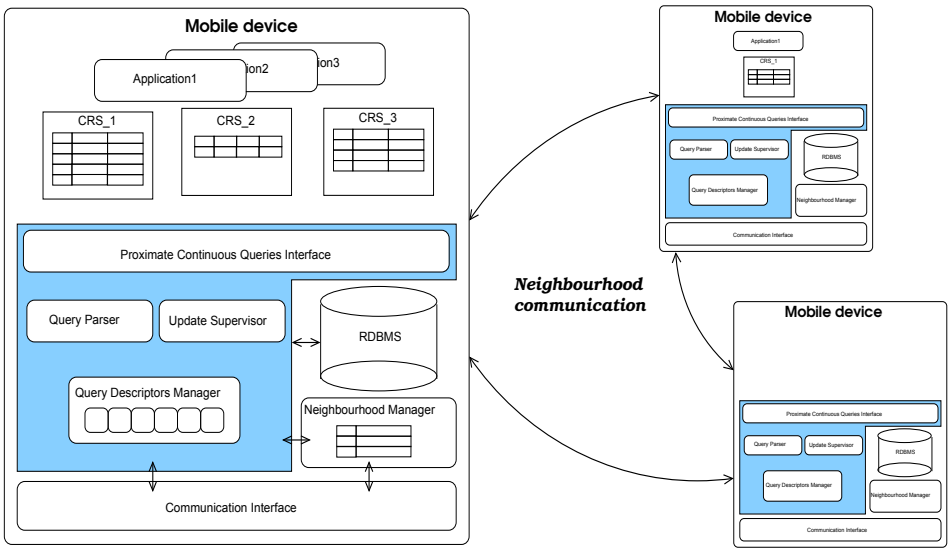


Figure 1: Architecture of the PERSEND querying system

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 neighbouring 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 Data-Base 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.

4.2.1 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(QueryText)`, 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.

4.2.2 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.

4.2.3 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.

4.2.4 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.

4.2.5 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.

4.3.1 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.

4.3.2 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 pre-defined 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.

8. 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.

A Localization Service for Proximity Applications

Marie THILLIEZ
Laboratory – CNRS UMR 8530
University of Valenciennes
Le Mont Houy
59313 Valenciennes Cedex 9 France
Marie.Thilliez@univ-valenciennes.fr

Thierry DELOT
Laboratory – CNRS UMR 8530
University of Valenciennes
Le Mont Houy
59313 Valenciennes Cedex 9 France
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. Proximity applications rely on the use of both different mobile devices and heterogeneous wireless networks. Thus, 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.

Keywords

M-Services, localization service, location queries

1. INTRODUCTION

The emergence of both handheld devices and wireless networks [11] 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 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 and finally, we conclude and present the perspectives of our works in section 5.

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 [14]. 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.

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.

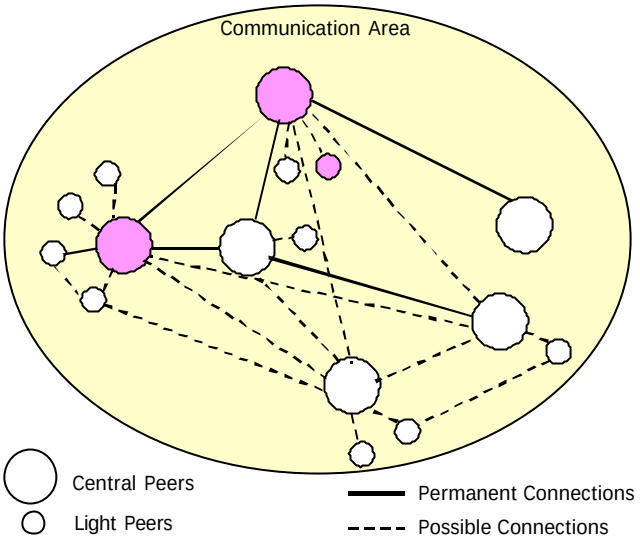


Fig. 1. Hybrid Peer-To-Peer Architecture

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 [12] or UDDI[1]). These solutions generally provide a central service, which registers the available information what 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

[1] Universal Description, Discovery and Integration : http://www.uddi.org

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 consume 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 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 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. 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.

[2] http://www.oasis-open.org/committees/dsml/

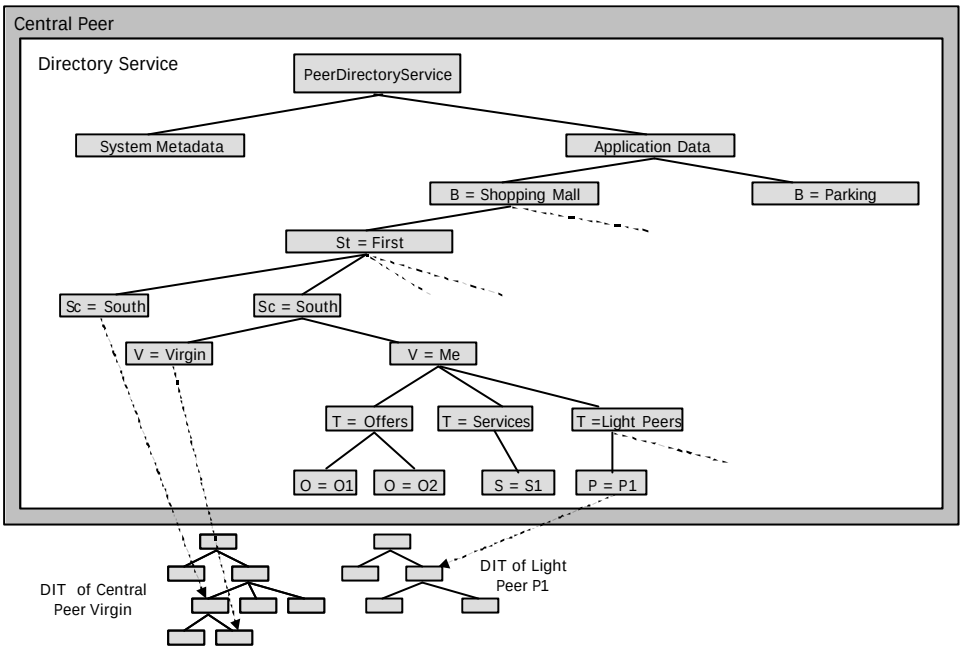


Fig. 3. Example of DIT deployed on a central peer

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.

- **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.

- **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, 13]. 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 :

1. *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.

[3] http://www.w3.org/TR/xquery/

```
<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

```
<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

2. 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.

3. 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.

```
<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

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

The localization service presented in this paper is under implementation. Data of central peers are stored in the OpenLDAP server. The query evaluator is implemented on the top of this server and distribution transparency is assured by the query evaluator thanks to the Java Naming and Directory Interface (JNDI) API. Pocket PCs Compaq Ipaq H3650 are used as light peers and their data are stored in XML files.

Our goal designing this first version of the prototype was to validate our approach. We are now considering performances issues since access times to distant objects are naturally much more important than access times to those stored locally what causes important problems. Moreover, the choices performed by the query evaluator in term of query forwarding are crucial when dealing with constrained evaluation.

6. CONCLUSION & 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 several 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.

7. ACKOWLEDGMENT

The authors wish to thank Sylvain Lecomte for his helpful comments on this paper.

8. 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] U. Varshney, and R. Vetter, Emerging Mobile and Wireless Networks, Communications of the ACM, Volume 43, Issue 6, 2000.

[12] M. Wahl, T. Howes, S. Kille, Lightweight Directory Access Protocol (v3), Internet RFC-2251, 1997.

[13] 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.

[14] B. Yang, H. Garcia-Molina, Comparing Hybrid Peer-To-Peer System, Int. Conf. On Very Large DataBases (VLDB) Conference, 2001.

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

Marco Cignini
Department of Mathematics and Computer Science
University of Udine
cignini@dimi.uniud.it

Stefano Mizzaro
Department of Mathematics and Computer Science
University of Udine
+39 0432 558456
mizzaro@dimi.uniud.it

Carlo Tasso
Department of Mathematics and Computer Science
University of Udine
+39 0432 558449
tasso@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.

Keywords
E-mail categorization, email filtering, email alerting, mobile devices, experimental prototype, evaluation.

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 [23]; 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. 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" [9]. Accordingly to Whittaker and Sidner [24], 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, 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 [23], i.e., the overloading of uses of this tool, and because of this phenomenon, email has been named a serial-killer application [8].

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

Mobile Information Access Workshop, Sep. 8, 2003, Udine, Italy.

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 [19]. 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 [7]: Cohen [6] uses the RIPPER algorithm; Payne and Edwards [16] compare CN2 (a rule induction algorithm) with IBPL1 (a modified version of K-nearest Neighbor algorithm using memory based reasoning); Rennie [17] exploits naïve Bayes classifiers; Segal and Kephart [20] 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. 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 [10].

Spam (or junk) email filtering has seen an increasing interest in last years, due to the increasing amount of unsolicited emails: Pantel and Lin [15] and Sahami et al. [18] 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 [12]. 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.

ifMail categorization and filtering are based on the IFT (Information Filtering Tool) system [11,12], capable of profile building, storing, and matching. IFT has been developed on the basis of the UMT (User Modeling Tool) shell [3] and has been applied to a variety of systems and domains, e.g., Web filtering [2], filtering of enterprise documents [21], and filtering of scholarly publications [13]. 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.

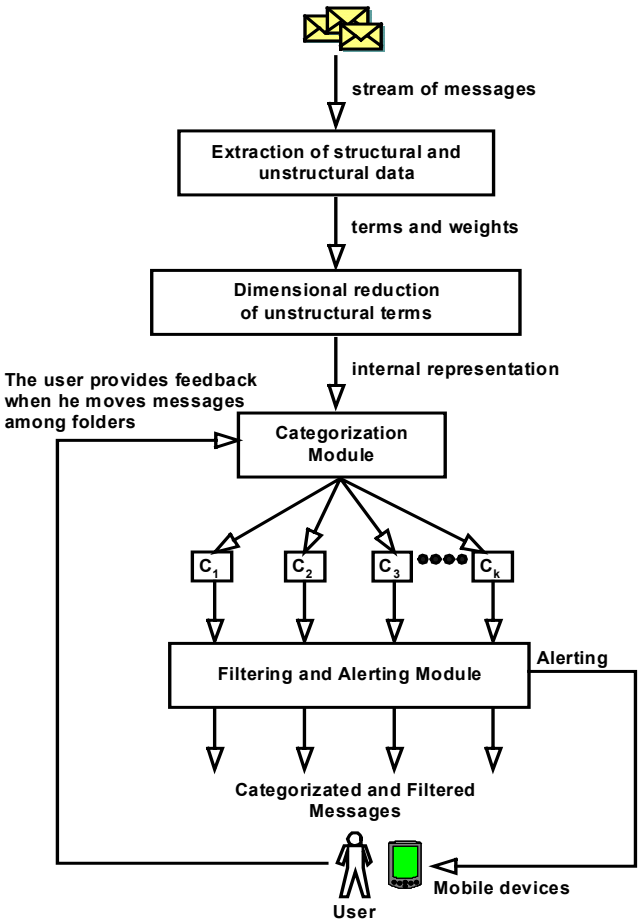


Figure 1. Conceptual model of ifMail operation.

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

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 sub-modules.
  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).

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.

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.

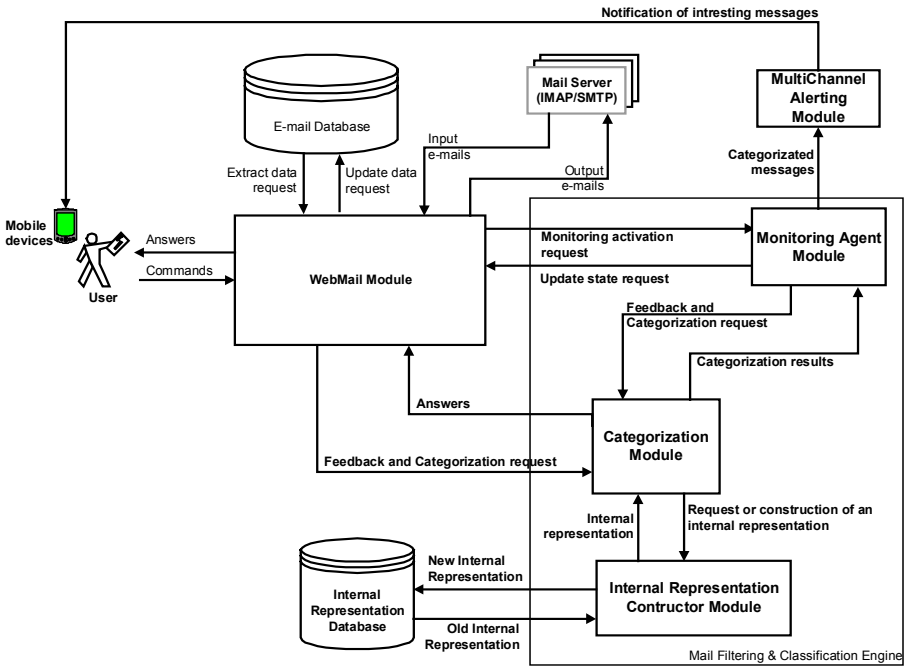


Figure 2. ifMail overall architecture.

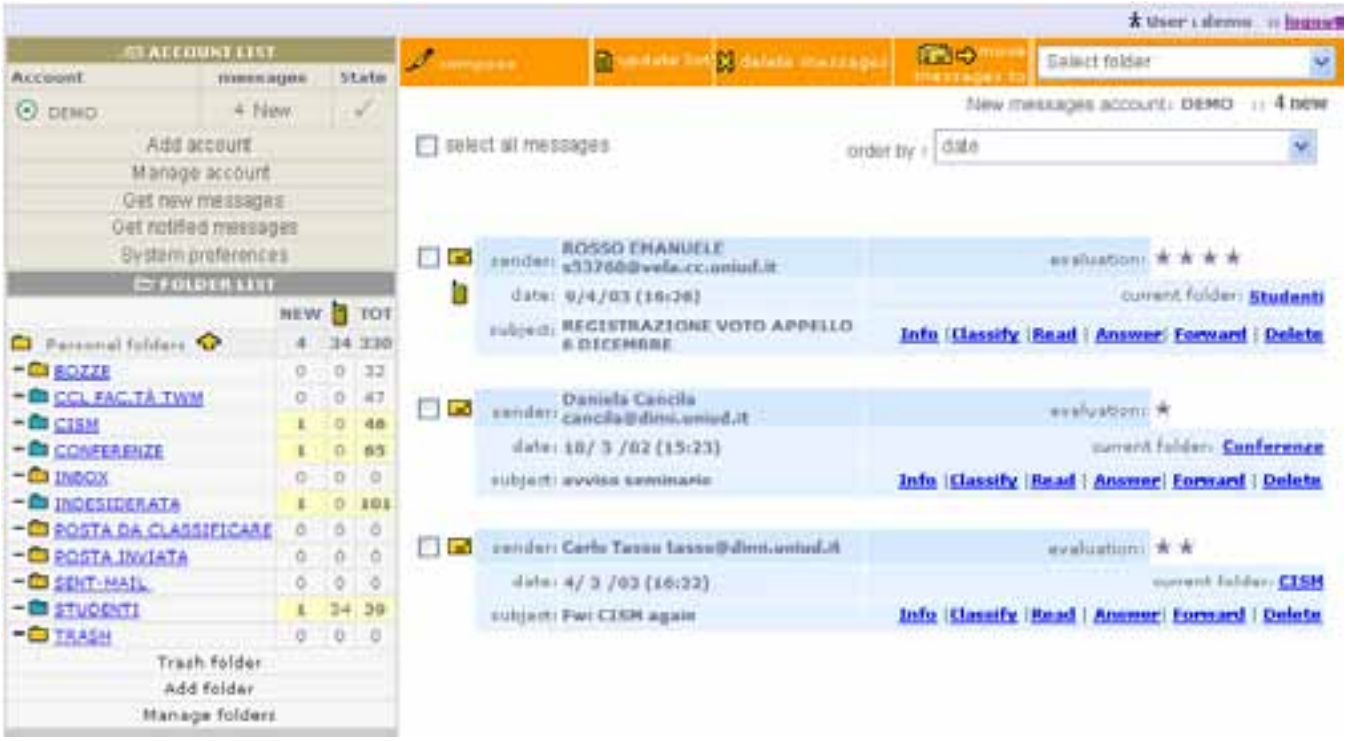


Figure 3. ifMail user interface for Web mail

Figure 4. ifMail user interface on a PDA: email notification (above) and reading (below).

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 [22, 25], where the results obtained for each category are combined using the micro-average indicator [19].

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 [14, 19] 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.

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).

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.

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

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 in 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).

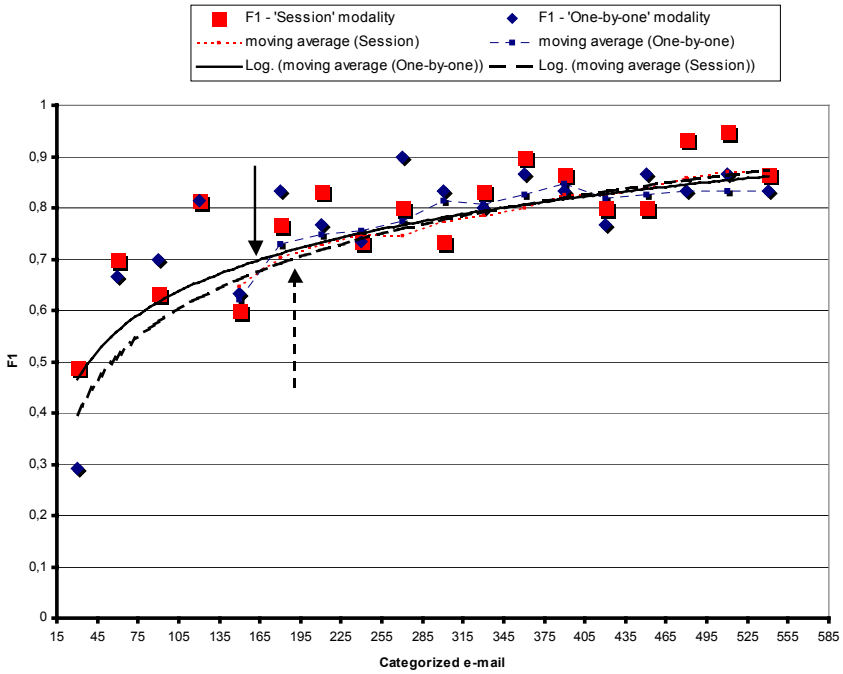


Figure 5. Microaverage F1 in both operation modes for collection E.

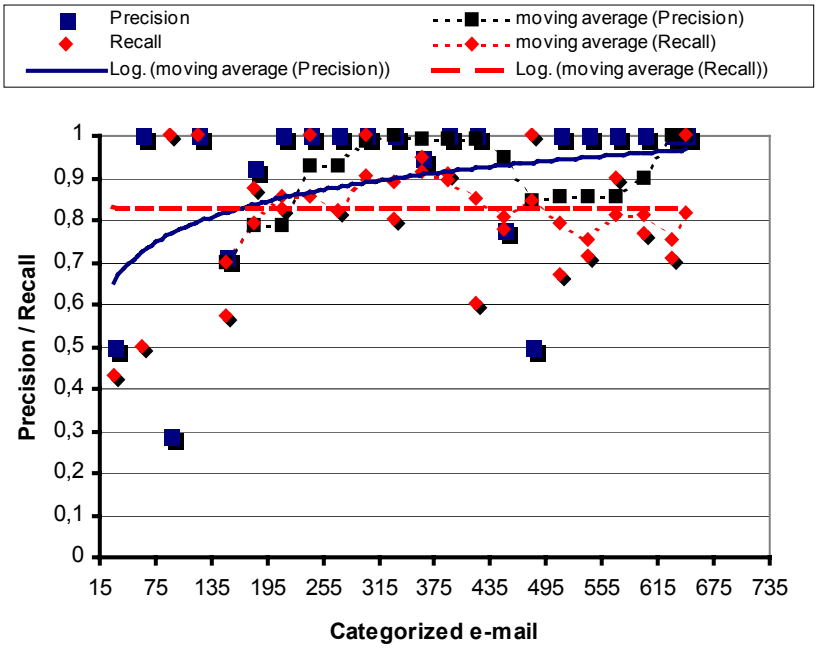


Figure 6. Precision and Recall for the Spam category of collection B.

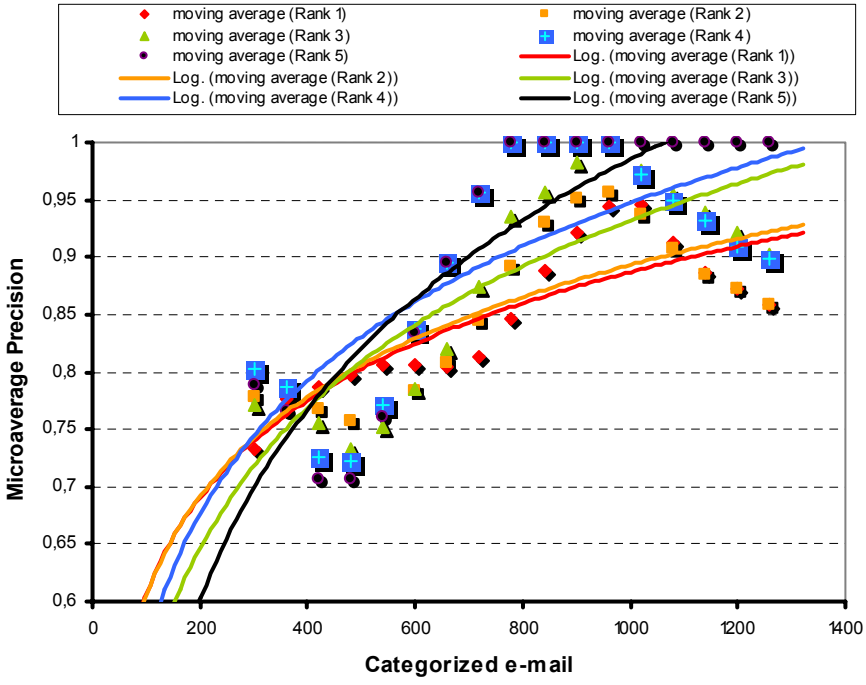


Figure 7. Precision with different values as alerting threshold for collection F.

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

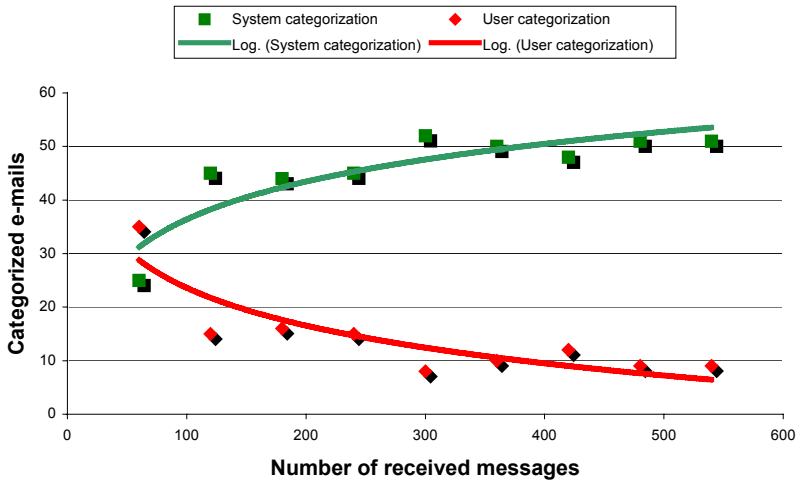


Figure 8. Comparison of the number of user and system categorization actions (session mode).

messages out of the incoming 60, with a missed-categorization rate of less than 16%.

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.

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.

REFERENCES

[1] I. Androutsopoulos, J. Koutsias, K.V. Chandrinos, G. Paliouras, and 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] 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.

[6] W. Cohen, Learning Rules that Classify E-Mail, Papers from the AAAI Spring Symposium on Machine Learning in Information Access, 1996, pp. 18-25.

[7] E. Crawford, J. Kay, and E. McCreath, Automatic Induction of Rules for e-mail classification, *Proceedings of the Sixth Australian Document Computing Symposium*, Coffs Harbour, Australia, Dec. 7, 2001

[8] N. Ducheneaut and V. Bellotti, Email as Habitat. *Interactions*, September/October 2001.

[9] W. Mackay, Diversity in the Use of Electronic Mail: A Preliminary Inquiry. *ACM Transactions on Office Information Systems*, 6(4), 380-397, 1988

[10] G. Manco, E. Masciari, M. Ruffolo, and A. Tagarelli, Towards An Adaptive Mail Classifier, *Atti dell'Ottavo Convegno AI*IA 2002*, Siena, Italy, 2002, pp. 63.

[11] 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.

[12] 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.
[13] 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, and 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
[14] 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.
[15] 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.
[16] T. Payne, P. Edwards, Interface agents that learn: An investigation of learning issues in a mail agent interface, *Applied Artificial Intelligence,* Volume 11, pp. 1-32, 1997.
[17] J. D. M. Rennie*,* ifile: An application of Machine Learning to E-Mail Filtering, *In Proceedings KDD00 Workshop on Text Mining,* Boston, 2000.
[18] 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.
[19] F. Sebastiani, Machine Learning in Automated Text Categorization, *ACM Computing Surveys*, 34(1), 1-47, 2002.
[20] 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.
[21] 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. .
[22] K. Van Rijsbergen, *Information Retrieval*, 2nd ed. Butterworths, London, UK, 1979. http://www.dcs.gla.ac.uk/Keith/pdf.
[23] G. Venolia, L. Dabbish, J.J. Cadiz, and A. Gupta, Supporting Email Workflow. Microsoft Research Tech Report MSR-TR-2001-88
[24] 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.
[25] Y. Yang, 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.

Towards The Wireless Ward: Evaluating A Trial Of Networked PDAs In The National Health Service

Phil Turner, Garry Milne, Susan Turner and Manfred Kubitscheck

School of Computing, Napier University, Edinburgh, UK.

+44 (0)131 662 2721

p.turner @napier.ac.uk

Ian Penman

Gastrointestinal Unit, Western General Hospital, Crewe Road, Edinburgh, UK.

+44 (0)131 537 1758

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.

General Terms

Security, Human Factors.

Keywords

Pilot study, medical informatics, intranet, PDA, wireless network

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. [6] 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 [7]. The hospital is researching the potential for closer integration between PDAs and Oracle databases. More generally, Shipman [8] 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 [9], while a feasibility study investigating the potential for PDAs in an Edinburgh intensive care unit [10] indicated benefits would be realised in both patient handover and the processing of 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

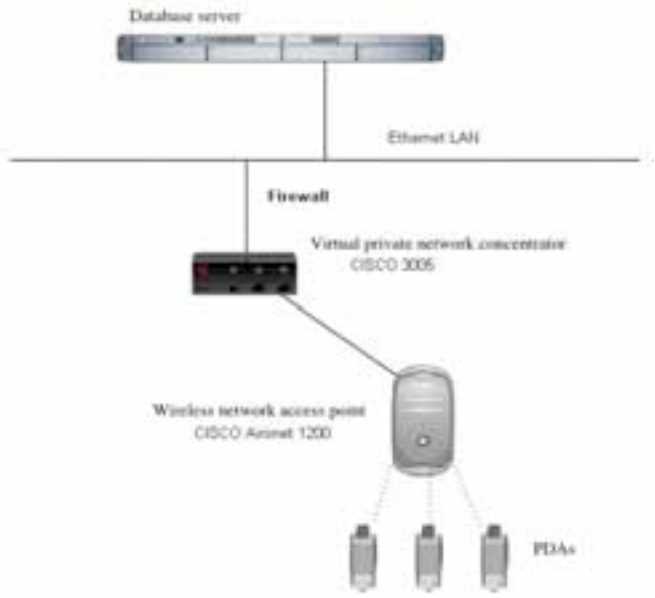
3.1 Security

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 will 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 will be achieved by the use of a VPN. A VPN router (the CISCO VPN concentrator 3005), will be the connection point to the hospital network, and will route all wireless communications through a secure tunnel connection to the PDA devices. The PDA devices will be equipped with the software client necessary to establish these end-to-end tunnel connections. Figure 1 is a diagram of the implementation.

Figure 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 currently 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 will be carried out to 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 will be 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. 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. 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.

Figure 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.

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 [1] 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.

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

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".

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 [2, 3, 4]. 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 [5]. 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.

Figure 3 – The 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, hand-writing 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.

Figure 4 – 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.

Table 2: Layers, indicative techniques and success criteria

Ergonomic / usability layer		
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.

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, e.g. consulting medical histories.	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 (single user and cooperative), for example, consulting test results.	Shadowing of staff, interviews, automatic usage logs, unintrusive user diaries	PDA is perceived by all staff to improve performance of relevant tasks.. Continued usage of PDA for relevant tasks by all staff.
Maintain security of patient data	Testing of data security	All patient data is secure. No access to any other data by unauthorised personnel

Activity-level layer		
Issues	**Techniques**	**Success factors**
Enhancement of patient care	Observation, interviews, collection of statistical data.	To be established with the clinicians. Also acceptance of publications in recognised academic forums, attraction of funding.
Enhancement of clinical governance		
Demonstration of effective innovation to wider NHS, clinical, technological and academic communities		

7. AS WE WRITE

As we write (July 2003), we have a working wireless network of PDAs which will be linked to the hospital's information system and interference testing is scheduled shortly. The clinicians have been introduced to both the technology (hardware and software) and have generated a long wish list of requirements. Familiarisiation with the device is in hand through the use of Outlook for arranging meetings, scheduling and dealing with email – the ability to do this away from the office and the desk already providing welcome benefits for some participants. Evaluation has started particularly at the ergonomic / usability layer and task level evaluation will begin in August.

8. REFERENCES

[1] Smithson, S. and Hirschheim. R. (1998). Analysing information systems evaluation: another look at an old problem. *European Journal of Information Systems*, **7**, 158-174.

[2] Turner, P. and Turner, S. (2002) Surfacing issues using activity theory, Journal of Applied Systems Science, 3(1), 51-60.

[3] 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.

[4] Turner, P. and McEwan, T. (to appear) Activity Theory: Another Perspective on Task Analysis. In D. Diaper and N. Stanton (Eds.) The Handbook of Task analysis. London: Kluwer.

[5] Leont'ev, A. N. (1978) *Activity, Consciousness and Personality*, (Eng. Tr. M.J. Hall) Prentice Hall Inc., Englewood Cliffs, NJ.

[6] 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.

[7] Corman, R. (2000) Cedars-Sinai uses Palm VIIs to Access Clinical Information. A news item reported on http://www.handheldmed.com/.

[8] Shipman, J.P. and Morton, AC (2001) The new Black Bag, PDAs, *Health Care and Library Services*, **29(3)**, 229-237.

[9] 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. Blanford, J. Vanderdonckt and P. Gray (eds.) *People and Computers XV – Joint Proceedings of HCI 2001 and IHM 2001*, London: Springer. 439-456.

[10] 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.

One-handed use as a design driver: enabling efficient multi-channel delivery of mobile applications

Mikko Nikkanen

Nokia Ventures Organization
P.O. Box 407
00045 Nokia Group, Finland
+358 50 487 6604
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.

Keywords

Multi-channel delivery, mobile services and applications, small interfaces, navigation, usability.

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.

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.

3.1.1 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	Under development	Under development
Voice	Yes	In pilot	-

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

3.1.1.1 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.

Figure 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.

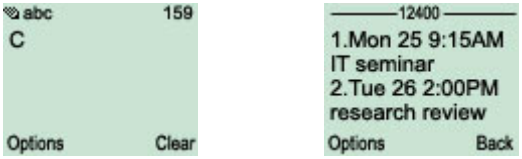


Figure 2. An example of calendar use through SMS. On the left, a request for calendar events in the near future, and on the right, a response displaying a list of two events.

Figure 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 figure 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.

3.1.1.2 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 Figure 4.

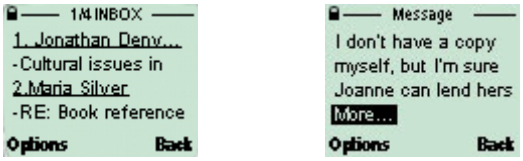


Figure 4. Examples of a WAP e-mail list and message screens. On the left, the list, and on the right, the message.

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 Figure 5.

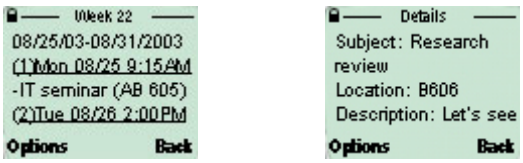


Figure 5. Examples of a WAP calendar event list and event detail screens. On the left, the list, and on the right, the event details.

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 Figure 6.

Figure 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.

3.1.1.3 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 4. Press 0 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 event 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…"*

3.1.1.4 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.

3.2.1 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.

3.2.2 Application specific remarks

The following remarks were made about specific applications.

3.2.2.1 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.

3.2.2.2 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.

3.2.2.3 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 were in the same meeting with them.

3.2.3 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.

3.2.4 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.

6. 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.

7. 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.

Ubiquitous Awareness in an Academic Environment

Miguel Nussbaum (mn@ing.puc.cl)
Roberto Aldunate
Farid Sfeid
Sergio Oyarce
Computer Science Department
School of Engineering

Roberto Gonzalez
School of Psychology

Universidad Católica de Chile
Vicuña Mackena 4860
SANTIAGO, Chile

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 with common needs. 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. Natural mobility of people is permitted under this model, by means of a distributed agent administration. These agents communicate among them to recognize common aims and obviousness of the different people in their vicinity.

We propose to enhance the idea of Socialware, to a face to face Socialware. Our aim 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 students profile and current needs of their owners. 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. The Ad Hoc network will be build with wireless interconnected Pocket PCs (IEEE 802.11b). In this way only people that are close to each other (within 50 meters) form the network, encouraging a face to face contact.

Keywords
Ad Hoc Networks, Ubiquitous Awareness, Distributed Agents, Meeting and Working with Strangers.

1. INTRODUCTION

Information exchange between two people occurs when they can 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 [4].

Therefore we do not talk with strangers in a bus since we have no queues about them. We do, however, give us to a stranger that is a Medical Doctor (MD) when we need his/her knowledge, due to our reliance to the MDs in general. Our trust is in this case with his/her membership and not with the specific person. Social actors need not to understand each other, they just relate. What is transcendent is the trust that the members of a system grant each other in given circumstances; rather than understanding their relationship what matters is their mutual compromise.

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 [7][10]. When one person sends a message to another, it is possible to understand it when a common language and a shared context exists [3][6][16]. 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 intra-group, the probability of establishing and maintaining an interaction is determined by their similitude and their differences [1][18]. 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. For this it is necessary to recognize and to communicate the common patterns between individual that are relevant in a given context.

Context refers mainly to our history and the way we have constructed our experience [11]. 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. Truth is therefore what the group think it is. In this way a person obviousness is constructed, what is implicit, what I known without saying it.

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. Communityware relates to amorph and diverse groups [14]; it is a dynamic community where there is not a fix organization and a clear aim [5]. How can people be organized in this dynamic milieu and what support is required to identify the relevant information for

This work was partially Funded by FONDECYT 1020734

each of them and make it available in the adequate context? [9].

We can find in literature model of agents that facilitate the encounter of people in the Internet [15][20]. In Socialware and Communityware usually there is a fix network, and when mobility is present, there is a server that store the user profiles and manages the users interactions [12][15][17][20]. This model has the problem of matching people that not necessarily are close and can only communicate through the network.

Our aim is to provide tools to ease the encounter of people that are physically close and are (usually) moving in a given setting with common needs. 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.

2. PROBLEM DEFINITION

In this work we propose an enhancement to the idea of Socialware, by the means of a face to face Socialware. Our aim 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. 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. The Ad Hoc network will be build with Pocket PCs using WI-FI (Wireless Fidelity; IEEE 802.11b). In this way only people that are close to each other (within 50 meters) form the network, encouraging a face to face contact.

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 our 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.
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.

Figure 1: Main Screen

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. GENERAL MODEL

Figure 2 illustrates the system components and their interactions:

1. Each of the users fills a questionnaire that defines his/her profile.
2. An agent supervises the user activities with the Pocket PC tools.

3. The agents search for other agents in the Ad Hoc network. Once another agent is found they compare their profiles and current needs.
4. Once two agents agree, they both communicate their findings to the user.
5. The agent finds out if the finding is transformed in an encounter. On the other hand, the agent tries to discover the reason, updating the corresponding profile and/or current list of needs.
6. Face to face contact is established when both users agree.

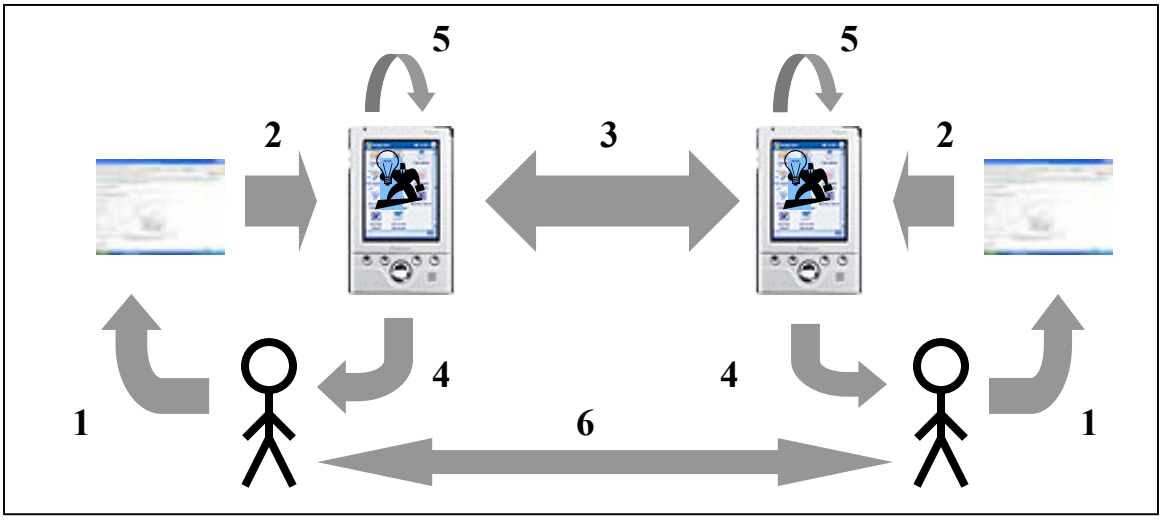


Figure 2. Processes among agents and users

4. MENTAL MODEL

An agent will essentially be the user "personification". Its profile will determine the user "Mental Model" [2]. One of the theories which has best contributed to understand interpersonal attraction is the similarity attraction hypothesis [13]. There are other factors besides similitude; attraction and attitude, as well as religious orientation [8] adhesion to traditional sexual roles [19] and preferred activities [13]. Factors that appear to be significant in the beginning of a relationship are affinity in basic values, interests and hobbies [13].

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 intercorrelations 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 3).

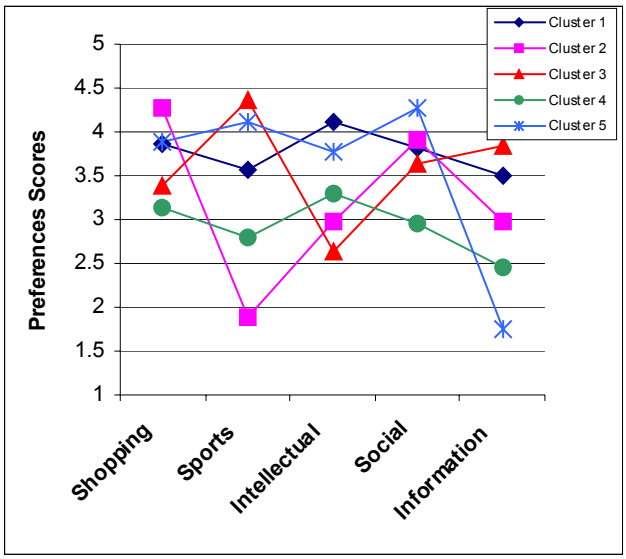


Figure 3: Cluster Analysis.

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.

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 3. 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.

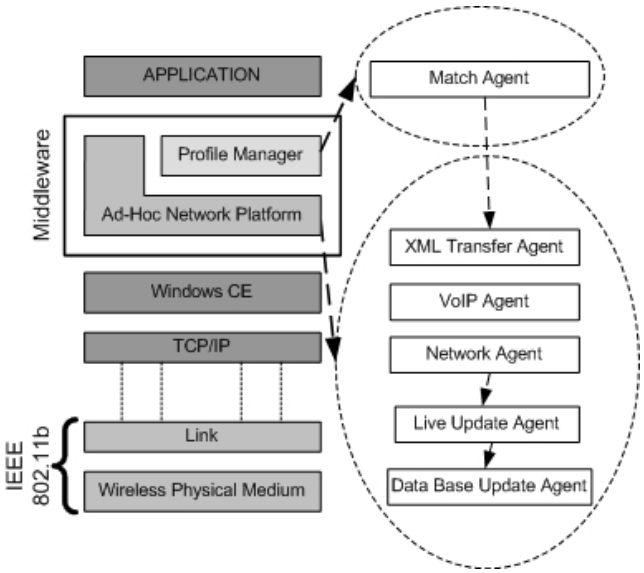


Figure 4. Ad hoc network architecture.

```
<!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)>
```

(c)

```
<?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>
```

(b)

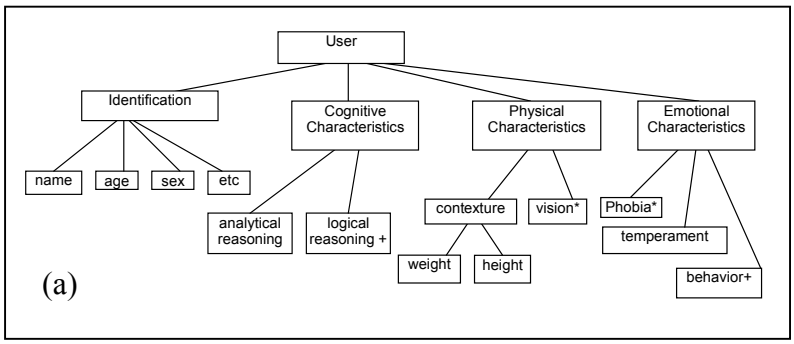


Figure 5. Data representation: (a) hierarchical data tree, (b) DTD associated to (a), (c) instance in XML that describes a user.

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.

6. CURRENT STATE

The academic year started in march 2003 and 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 is monitored by Wi-Fi enabled PCs, strategically located in campus. The PCs are part of the peer-to-peer network as an access point to the Internet to monitor the students machine usage. During the semester we want to find out:

1. A dimension of attraction and contact among participants. According to the similarity – attraction hypothesis, it is expected that those participants who share the same cluster membership will be more attracted to each other than members from different clusters. Thus, it is expected that at both the middle and end of the semester (second and third time of the measure) the experimental group will report more frequency and amount of contact with members of their own cluster than those of the control group. Second, participants of the experimental condition will contact students of their own cluster earlier than members of the control group. Based on the assumption of earlier contact and communication, it is expected that participants of the experimental condition, compared to members of the control one, at time three will develop more close and stable interpersonal relationships along the semester. Complementary, any potential impact of the academic achievement will be analyzed between the experimental and control group.
2. Record of Availability. How much will the students use the technology? Why and under which conditions?
3. Scalability. Will the experience work better when few students are present or when all of them are forming the Ad Hoc network?
4. Intrusion. It will measure the extent to which the use of this device affect the students academic lives in any negative and positive way, specifically if they feel that the agents invade their lives.

At publishing time of this paper, almost end of the semester, we have our first conclusions:

1. A big drawback in the experiment has been the batteries life. We are using Toshiba e740, that incorporates WiFi inside, which only last two hours on. Since it has no stand by mode where only the WiFi can be enabled 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.
2. We could experimentally find out the learning curve of new technologies (Figure 6) [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 getting 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. Those that stood with the tool, increasingly augmented their usage after the first month.

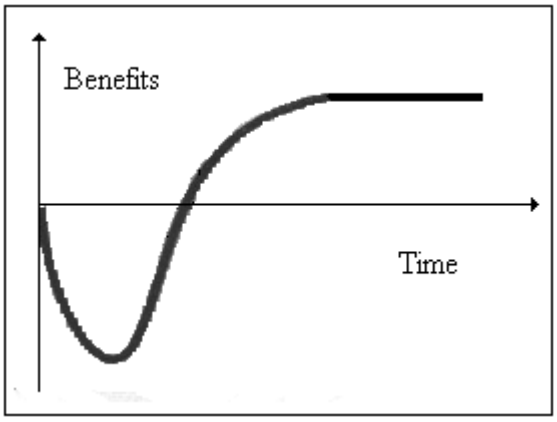


Figure 6: Learning Curve of New Technologies [21]

7. REFERENCES

1. Brown R.J. Group Processes. Second Edition. *Oxford: Blackwell*. 2000.
2. Byrne D. The Attraction Paradigm. *New York: Academic Press*. 1971.
3. Gardner H. Perspectives of Mind and Brain. *In the disciplined mind: What all students should understand*, N.Y. pp 60-85.
4. Giddens. A. The consequences of Modernity. *Stanford University Press*. 1990.
5. 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.

6. Hetland L. Understanding Goals: Teaching the Humanities for Undersatnding in Middle School, *AERA Annual Meeting*, April 1996, NY.
7. Ishida T. Towards Community Ware, *PAAM'97* invited talk, 1997.
8. Kandel, D.B. Similarity in real life adolescent friendship pairs. *Journal of Personality and Social Psychology*, 1978. Nro. 65, pp. 282-292.
9. La Liberte, D. & Wooley., D. Presentation Features of Text-Based Conferencing Systems on the WWW**.** *Computer-Mediated Communication Magazine***.** Vol 4, Nro. 5. May 1997.
10. MacIver R. Community, *Macmillan Co*. 1917.
11. Mansilla V.B. Historical Understanding: Beyond the past and into the present. *Conference on Knowing, Teaching and Learning History*. Pittsburgh, November 1998.
12. 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.
13. 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.
14. 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.
15. Okada, K., Maeda, F., Ichikawa, Y. & Matsushita, Y. Multiparty Videoconferencing at Virtual Social Distance: MAJIC Design. *Proceedings of CSCW' 94*. 1994, pp 385 - 393.
16. Perkins D. What is Understanding? In Teaching for Understanding *(D. M.S. Wiske)*, San Francisco.pp 39-57.
17. Root R.W. Design of a Multi-Media Vehicle for Social Browsing. *Proceedings of the CSCW'88*. 1988. pp 25-38.
18. Sherif, M. & Sherif, C.W. Social Psycology. New York: Harper & Row. 1969. chs 9-10.
19. Smith, E. R., Byrne, D., y Fielding, P. J. Interpersonal attraction as a function of extreme gender role adherence. *Personal Relationships*, 1995. Nro.2, pp.161-172.
20. Takemura, H. & Kishino, F. Cooperative Work Environment Using Virtual Workplace. *Proceedings of CSCW' 92*, 1992, pp 226 - 232.
21. Glass, Robert. The Reality of Software technologies Payoffs. Communications of the ACM February 1999, Vol. 42, No. 2.

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

Mario Pichler
Software Competence Center Hagenberg (SCCH)
Hauptstrasze 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 [23, 28, 29]. 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 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 [14].

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 ser-

[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.

vices 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 [18]

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* ([4], 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 [1], 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 [18], 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, rather than 'the ubiquitous system'" [18]. A possible set of context areas a human might meet during a workday is illustrated in Figure 1.

Figure 1: Relevant Context Areas During a Workday

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 [15] and José et al. [16]. 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. [30]. 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.

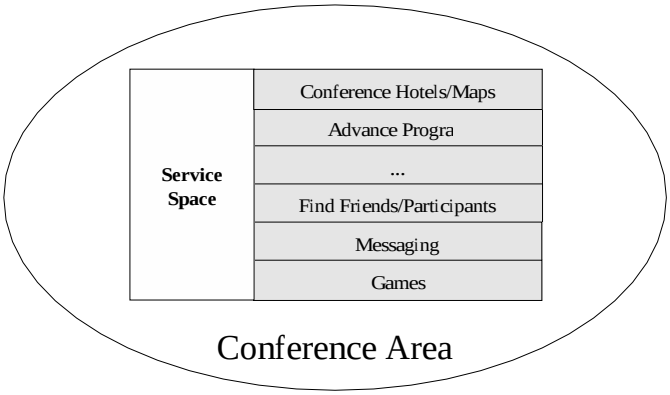


Figure 2: Service Space at Conference Site

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).

Figure 3: Interactive Conference: Available Services

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.

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.

Figure 4: Mobile Passenger Information and Ticketing: Available Services

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).

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

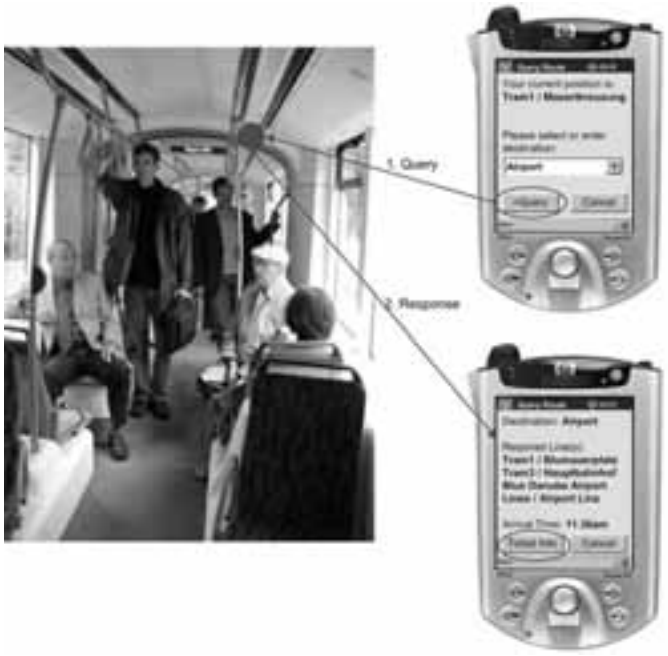


Figure 5: Query Route Service

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 [8, 9, 10] established the idea that there are distinct levels of proximity in interpersonal communication:

- *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 [19] 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 [6], 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 [15] and José et al. [16] is closely related in terms of providing services that are relevant within a specific area. The work of Kortuem [19] 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 [26].

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 [5] for instance focuses on the development of context aware applications. Proem by Kortuem et al. [20] and XMIDDLE by Mascolo et al. [22] 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. [24] 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 [2], JavaSpaces [13], TSpaces [27], Limbo [3], and GigaSpaces [7].

Ongoing work in the service discovery domain incorporating the characteristics of mobile environments is also of interest for the described work. Konark [12] 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 [11] are describing a service model built on top of LIME for service provision in ad hoc networks. JDSP (JESA Service Discovery Protocol) [25] also aims at efficient service discovery in ad hoc networks. And Lee and Helal [21] 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 [17] 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 common 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.

7. REFERENCES

[1] L. Capra, W. Emmerich, and C. Mascolo. Middleware for mobile computing. *UCL Research Note RN/30/01*, 2001.

[2] CORSO: Corso shared object space technology. http://www.tecco.at/en/eTechnology.html. Last visited: July 2003.

[3] N. Davies, S. Wade, A. Friday, and G. Blair. 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, May 1997.

[4] A. K. Dey. *Providing Architectural Support for Building Context-Aware Applications*. PhD thesis, College of Computing, Georgia Institute of Technology, December 2000.

[5] A. K. Dey and G. D. Abowd. The context toolkit: Aiding the development of context-aware applications. In *Workshop on Software Engineering for Wearable and Pervasive Computing*, Limerick, Ireland, June 2000.

[6] M. Esbjörnsson, O. Juhlin, and M. Östergren. The hocman prototype - fast motor bikers and ad hoc networking. In *Proceedings of MUM 2002*, Oulu, Finland, December 2002.

[7] GigaSpaces. http://www.j-spaces.com/. Last visited: July 2003.

[8] E. T. Hall. *The Silent Language*. Anchor Press/Doubleday, New York, 1959.

[9] E. T. Hall. *Proxemics: The Study of Man's Spatial Relations*. International University Press, Connecticut, 1962.

[10] E. T. Hall. *The Hidden Dimension*. Anchor Press/Doubleday, New York, 1966.

[11] R. Handorean and G.-C. Roman. Service provision in ad hoc networks. In *Proc. of the 5th International Conference COORDINATION 2002*, number 2315 in Lecture Notes in Computer Science, pages 207–219. Springer-Verlag, April 2002.

[12] A. Helal, N. Desai, and V. Verma. 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, March 2003.

[13] JavaSpaces technology. http://java.sun.com/products/javaspaces/. Last visited: July 2003.

[14] R. Johansen. *Groupware: Computer Support for Business Teams*. Free Press, New York, 1988.

[15] R. José. *An Open Architecture for Location Based Services in Heterogeneous Mobile Environments*. PhD thesis, Computing Department, Lancaster University, England, April 2001.

[16] R. José, A. Moreira, and F. Meneses. An open architecture for developing mobile location-based applications over the internet. In *6th IEEE Symposium on Computers and Communications*, Hammamet, Tunisia, July 2001.

[17] JXTA: Project JXTA. http://www.jxta.org. Last visited: July 2003.

[18] T. Kindberg and A. Fox. System software for ubiquitous computing. *IEEE Pervasive Computing*, 1(1):70–81, 2002.

[19] G. Kortuem. *A Methodology and Software Platform for Building Wearable Communities*. PhD thesis, Department of Computer and Information Science, University of Oregon, December 2002.

[20] G. Kortuem, J. Schneider, D. Preuitt, T. Thompson, S. Fickas, and Z. Segall. 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)*, pages 27–29, Linköpings, Sweden, August 2001.

[21] C. Lee and A. Helal. 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, January 2003.

[22] C. Mascolo, L. Capra, S. Zachariadis, and W. Emmerich. XMIDDLE: A data-sharing middleware for mobile computing. *Personal and Wireless Communications Journal*, 21(1), April 2002.

[23] MB-net: Network of excellence in mobile business applications and services. http://www.mbnet-forum.org/. Last visited: July 2003.

[24] A. Murphy, G. Picco, and G.-C. Roman. Lime: A middleware for physical and logical mobility. In *Proc. of the 21st International Conference on Distributed Computing Systems (ICDCS-21)*, pages 524–233, Phoenix, AZ, USA, April 2001.

[25] S. Preusz. 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.

[26] M. Román, C. Hess, R. Cerqueira, A. Ranganathan, R. Campbell, and K. Nahrstedt. A middleware infrastructure for active spaces. *IEEE Pervasive Computing*, 1(4):74–83, 2002.

[27] TSpaces: Intelligent connectionware. http://www.almaden.ibm.com/cs/tspaces/. Last visited: July 2003.

[28] Umts Forum. http://www.umts-forum.org/. Last visited: July 2003.

[29] WWRF. wireless world research forum. The book of visions 2001 - visions of the wireless world, December 2001.

[30] S. Yau, F. Karim, Y. Wang, B. Wang, and S. Gupta. Reconfigurable context-sensitive middleware for pervasive computing. *IEEE Pervasive Computing*, 1(3):33–40, 2002.

Models and Services for Mobile Learning Systems

Alfio Andronico
Dip. Ingegneria Dell'informazione
Università degli Studi di SIENA
Tel +39 0577 233613 – Fax 3602
andronico@unisi.it

Antonella Carbonaro
Department of Computer Science
University of Bologna,
I-40127 Bologna Italy,
tel +39 0547 642830, fax 610054
carbonar@csr.unibo.it

Luigi Colazzo
Department of Computer and
Management Sciences
University of Trento
Tel. +39 0461882144 Fax 2124
colazzo@cs.unitn.it

Andrea Molinari
Department of Computer and
Management Sciences
University of Trento
Tel. +39 0461882344 Fax 2124
amolinar@cs.unitn.it

Marco Ronchetti
Department of Information and
Communication Technology –
University of Trento
38050 Povo (Trento) – Italy
Tel. +39 0461882033
marco.ronchetti@dit.unitn.it

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. 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. 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.

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 evolution 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]

This contribution 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.

2. The three elements of building a mobile learning environment

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. 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 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:a key element for this is a constant monitoring, whose aim is to both evaluate users and

the whole process. 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 five points, concerning multiple choice test: 1) they are concerning the results of the learning process, not to the processes 2) they underline the knowledge level not the potential of learning 3) they are far from the working contexts 4) the memory can sometimes be more useful than the comprehension 5) the so-called tests-taking skills can affect the result. Possible answers to these problems are presented in [8]. In the context of the present project we would like to highlight two particulars. First, 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 [9], [10], [11], [12]. Several implementations are available on the market, like for instance LearningSpace™, WebCT™, Blackboard etc. [13]. 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 [14]. 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 teachers 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. 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.

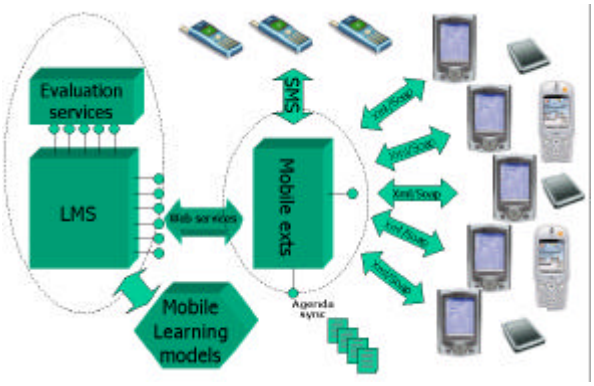


Fig. 1: A general schema of the prototype

3. 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. 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.

As just introduced in the introduction, 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 [15]. 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.

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 adaptively) 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: a) true/false; b) multiple-response question; c) extended matching item and drag and drop question types share the same process of selection. d) image hot spot; e) 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 these 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;

b) Creation of the test databases organized in atomic sets of different kind of requests (multiple choice, open, closed, fill in gap, building of sentences, problem-solver, …);

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;

e) system evaluation which allows to make experimentations on the principal platforms which at present the more diffuse PDA computers equip on the market.

4. Adapting a Learning management system to infomobility

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, user identification and authorization, gathering of data relative to the user-system interaction, provisioning of mobile services etc. 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: 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 several mobile devices (PDA, pocketPC, cell-phones, smart-phones, tablet-pc).

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. 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 infomobility will need different steps:

a) Extension of the traditional functions of a LMS to the mobile-computing needs required by the project.

b) Distribution of the educational material specifically created for the fruition on mobile equipment.

c) Integration of the self-evaluation system into the LIS.

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, 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. 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. As regarding the use of specific tools available with mobile technology, 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 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. 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;
- 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;
- Synchronizing the PDA agenda of the user with the central DB by using cradles and DB synchronization;
- 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.
- Building a web application that request a web service through the use of XML/SOAP messages. This is the best solution we found, as it provides the access in short time to the central DB through the use of open technologies 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.

5. 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, (p. 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, (p. 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, (p. 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, (p. 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] 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

[9] 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

[10] Hall B, Learning Management Systems. How to Chose the Right System for your Organisation, Brandon Hall, 2001

[11] McMahon M., Luca J, Courseware Management Tools and Customised Web Pages: Rationale, Comparisons and Evaluation, Proc. EDMedia 2001, Tampere, Finland, June 25-30 2001

[12] Hanna, D. E., Glowacki-Dudka, M. & Conceicao-Runlee, C. (2000). 147 Practical tips for teaching online groups: Essentials of Web-based education. Madison, WI: Atwood Publishing.

[13] Aggarwal, A. Web-based learning and teaching technologies: Opportunities and challenges.. Hershey, PA: Idea Group Publishing 2000.

[14] Colazzo L., Molinari A. (2002) 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

[15] Dalziel, J. R., & Gazzard, S. (1999b). 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, 93-96. Geelong: Deakin University.

Demonstrations
and
Short Talks

Taeneb:
Map centred tourist information access on palm-tops

Mark Dunlop
University of Strathclyde
Richmond Street, Glasgow, G1 1XH
Scotland
+44 141 548 3497
Mark.Dunlop@cis.strath.ac.uk

Categories and Subject Descriptors
H.1.2 [**User/Machine Systems**]: *Human factors.*

General Terms
Design, Human Factors.

Keywords
Palmtop mapping, starfield displays, tourism information.

1. DEMONSTRATION ABSTRACT
This demo will present the Taeneb CityGuide that provides tourist and venue information access on palmtop computers. It will concentrate on the two main technical developments in the project as they affect tourists: high quality, directly queryable maps on palmtops and community reviewing of venues.

Starfield display technology has been proved 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 domains such as house purchases and searching for movies. However, starfield technology has traditionally been used only for large colour screens. 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, and at the time monochrome, screens [2].

In this project we have developed high quality easy-to read maps (in colour and monochrome) for Palm OS palmtop computers based around starfield displays: in our prototype users are presented with a map of Glasgow City Centre with restaurants shown as small dots on the map. To complete the starfield design we have combined these maps with a set of filters so that users can chose to see only matching venues on the maps. In the RestaurantGuide prototype for example, users can filter on average price of a meal and type of food so could filter to show only, say, expensive Italian or French restaurants. The map reacts immediate to give users visual feedback on each selection. This style of map-centred querying supports both precise and imprecise searches; supports discovery of high level information, such as clusters of matching targets (e.g. areas of the city centre with many restaurants matching the filters such as the famous Brick Lane area of London for Indian restaurants); and overcomes many of the problems of both traditional querying and traditional hypertext style designs.

In addition to providing a queryable map as the core of the restaurant guide we have also developed a community review system. Here users can write reviews of restaurants and, subject to approval by an administrator, have them published so that all other restaurant guide users can read them. Not only does this allow more dynamic reviews than traditional book reviews (changes to restaurant ownership and quality of service, for example, can be reflected much quicker in this guide than traditional professionally reviewed guide books) but it allows filtering of messages to match particular users, in a similar fashion to Stick-E Notes use of filtering for ubiquitous message delivery [3]. For example, a young Californian visiting Scotland may only trust another Californian's or a Japanese person's judgement on whether Sushi in Scotland is typically any good. She is unlikely to trust the opinion of, say, a Scot who has never had Californian or true Japanese Sushi. By recording demographic information about users with their reviews this can be later be used to filter reviews so that users can chose to see a suitable subset of reviews given their own user profile. We believe this provides a powerful new review style to complement traditional professional reviews.

The demo will give a brief overview of the palmtop system for dynamic mapping community reviews of local restaurants.

2. ACKNOWLEDGMENTS
This project was funded by Scottish Enterprise through the Proof of Concept fund. The project was developed by the team comprising Alison Morrison and Chris Risbey from the Scottish Hotel School and Stephen McCallum, Piotr Ptasinski and Fraser Stuart from Computer and Information Sciences.

3. REFERENCES
[1] C. Ahlberg and B. Shneiderman, "Visual Information Seeking: Tight Coupling of dynamic query filters with Starfield displays", *Proceedings of CHI '94*, ACM Press, 313-317 & 479-480, 1994.

[2] M. D. Dunlop and N. Davidson, "Visual information seeking on palmtop devices", *Proceedings of HCI2000*, volume 2 pp 19-20, September 2000.

[3] D. Salber, A.K. Dey and G.D. Abowd, "The Context Toolkit: Aiding the development of context-enabled applications", in *Proceedings of CHI'99*, pp. 434-441, 1999

The Físchlár-News Archive

Cathal Gurrin
Centre for Digital Video Processing
Dublin City University
Ireland
353 1 700 5234
cgurrin@computing.dcu.ie

Alan F. Smeaton
Centre for Digital Video Processing
Dublin City University
Ireland
asmeaton@computing.dcu.ie

Hyowon Lee, Kieran McDonald, Noel Murphy, Noel O'Connor, Sean Marlow
Centre for Digital Video Processing
Dublin City University
Ireland

1. Introduction

The Físchlár-News Video archive is one of the results of research into analysis, browsing and searching of digital video content carried out at the Centre for Digital Video Processing, Dublin City University. Físchlár-News automatically records the 9pm, evening news from the Irish national broadcaster RTÉ1 each day and segments this programme into news stories. Currently there are several months of recorded daily news programmes in the archive (and another two year's news archived).

Físchlár-News supports browsing and retrieval of news stories on both desktop and mobile devices:

- Mobile access via a web browser for both PDAs (Compaq iPAQ on a wireless LAN) and XDAs (using a GPRS connection), each of which plays RealVideo content at 20Kbps (see Figure 3).
- In a desktop environment, a web browser is used with MPEG-1 video streaming at 1Mbps (see Figure 1).

Supporting multiple access devices is possible because Físchlár-News is based on XML technologies, which by incorporating XSL transformations for each new device required, can easily be extended to incorporate new access devices and standards.

Figure 1. The Desktop Interface to Físchlár-News

2. Físchlár-News on the Desktop

When using Físchlár-News on a desktop device, there are a number of ways of accessing news stories:

- **Browsing News by Programme**: A listing of news stories grouped by month is provided and selecting any news program displays a list of the news stories from that program (see Figure 1). More detailed keyframe browsing of news stories is also supported if the user requires.
- **Content Searching for 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 supporting user queries against these textual descriptions of each story.
- **Following Automatically Generated Links**: Using the closed caption transcripts for a given news story, we identify similar stories to any one given story, and thereby support content-based hyperlinking of news stories.

3. Físchlár-News on a Mobile Device

Small display size, awkward methods of data input and limited bandwidth are among the major constraints in designing systems for mobile platforms. In order to address these issues user interaction with a mobile device should be limited to a subset of the functionality of the desktop version. Consequently, the functionality of Físchlár-News on a mobile device is based around the following two core aspects:

- **Personalisation**: Providing personalised access to the news archive by presenting the user with a listing of news stories of interest to the user (see Figure 2).
- **Story Browsing**: Supporting user to access news stories in the archive by browsing a reverse chronologically ordered listing of news programmes.

Figure 2. Personalised story recommendations.

Figure 3. Playback on a mobile device.

See our full-length paper for a more detailed description of the Físchlár-News archive.

Human-system Interaction Container Paradigm

Célestin Sedogbo, Pascal Bisson, Olivier Grisvard, Thierry Poibeau, Jérôme Lard
Claire Laudy, Bénédicte Goujon, David Faure, Sébastien Praud

THALES Research & Technology France
Domaine de Corbeville – 91404 Orsay Cedex – France
[hit-trt]@thalesgroup.com

Abstract

As today interaction means become more and more various and sophisticated, interaction demands the integration of heterogeneous modalities, such as voice, gesture, graphics and animation, as well as appliances, such as classical laptop and desktop workstations, mobile phones, Personal Digital Assistants (PDA), PC tablets, etc.

A solution for providing users with wider access to existing systems and for enhancing user-friendliness of existing interaction means is to design intelligent interaction systems that dynamically adapt to the interaction environment and react appropriately in various contexts of use, without implying any modification to the core application.

We illustrate here the concept of Human Interaction Container (HIC) which introduces an important shift in the field of Human-Computer Interaction, moving from an application-centric to user-centric perspective, through the adoption of a service-oriented view of application and user interface capabilities.

The HIC aims at encapsulating all software components dedicated to user-system interaction management into a context-aware and context-sensitive container enacting as a mediator between the application services and the presentation services. As such, this container is designed so to ease the logical separation between application, interaction and presentation, and handle all the interaction processes enabling an application and its various user interfaces to communicate with each other. It offers application-independent and interface-independent interaction services which support intelligent adaptive interaction. These generic interaction services include dialogue processing, task and activity planning, user adaptation, multi-modality management and multimedia presentation generation.

Our approach aims thus at being able to dynamically generate or adjust a presentation which best fit the user's expectation according to its own context and its current task. Contextual information will be accessed to constrain the design or re-design of a presentation to take into account some user's data.

This research paves the way towards highly dynamic and mobile interfaces where the UI will be virtualized and will become real, only once an operator has been connected to it. With this approach UI will be no more static but highly dynamic, personalized and accessible through a broad range of various devices ranging from Desktop/Laptop to PDA/Mobile Phone.

Demo of the ifMail Prototype

Marco Cignini
Department of Mathematics and Computer Science
University of Udine
cignini@dimi.uniud.it

Stefano Mizzaro
Department of Mathematics and Computer Science
University of Udine
+39 0432 558456
mizzaro@dimi.uniud.it

Carlo Tasso
Department of Mathematics and Computer Science
University of Udine
+39 0432 558449
tasso@dimi.uniud.it

We demonstrate the ifMail prototype, described elsewhere in these proceedings. By ifMail we propose an integrated approach to email categorization, filtering, and alerting on mobile devices. ifMail is capable of:

- Categorize incoming email messages into pre-defined categories. Categorization is obtained on the basis of a profile attached to each user-defined folder and dynamically updated by means of user's feedback.
- Filter and rank the categorized messages according to their importance. 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.
- Alert the user on mobile devices when important messages are waiting to be read. We have been experimenting with PDA, where the notification of the arrival of a new relevant message for a specific category is shown together with a number of stars representing the message relevance computed by the system. The user can then archive the message, read message data like sender and subject, or read the whole message body. We are currently extending the system for WAP enabled phones.

These functionalities are performed mainly on the server-side, since they are, from the computational viewpoint, rather expensive.

In the demo we will present the typical usage of the system, both on various mobile devices and on a "normal" (i.e., desktop, or portable) PC. We also discuss how the choice of an integrated solution to email categorization, filtering, and alerting is a necessary choice if a high effectiveness is required.

Mobile Information Access Workshop, Sep. 8, 2003, Udine, Italy.