PROCEEDINGS OF THE THIRD WORKSHOP ON MOBILE AND UBIQUITOUS INFORMATION ACCESS

MUIA 06

Edited by

Fabio Crestani
Matt Jones
Stefano Mizzaro

12 September 2006
Mobile HCI 2006
Espoo, Finland

TABLE OF CONTENTS

MUIA 2006: Third International Workshop on Mobile and Ubiquitous Information Access

Fabio Crestani
Department of Computer and Information Sciences, University of Strathclyde, Glasgow, UK
f.crestani@cis.strath.ac.uk

Matt Jones
University of Wales Swansea, Singleton Park, Swansea, UK
Matt.Jones@swansea.ac.uk

Stefano Mizzaro
Department of Maths and Computer Science, University of Udine, Udine, Italy
mizzaro@dimi.uniud.it

ABSTRACT
The recent trend towards pervasive computing and information technology becoming omnipresent and entering all aspects of modern living, means that we are moving away from the traditional interaction paradigm between human and technology being that of the desktop computer. This shift towards ubiquitous computing is perhaps most evident in the increased sophistication and extended utility of mobile devices, such as mobile phones, PDAs, mobile communicators (telephone/PDA) and Tablet PCs. Advances in these mobile device technologies coupled with their much-improved functionality means that current mobile devices can be considered as multi-purpose information access tools capable of complex tasks. This Third Workshop on Mobile and Ubiquitous Information Access (MUIA 2006) aims to be a forum for the presentation of current research and exchange of experiences into technological and usability aspects of mobile information access.

Categories and Subject Descriptors
H.5.2 [**User Interfaces**]: User-centered design.

General Terms
Human Factors

Keywords
Mobile information access.

1. MOTIVATIONS
The ongoing migration of computing and information access from the desktop and telephone to mobile computing devices such as PDAs, tablet PCs, and next generation (G3) phones poses critical challenges for research in information access and, in particular, for Information Retrieval (IR). These devices offer limited screen size and no keyboard or mouse, making complex graphical interfaces cumbersome. This change in information access devices is also reflected by a radical change in user groups and tasks. Most future users will have low levels of IT literacy and will not be information access professionals, but causal users. Therefore, these mobile devices will be used in situations involving different physical and social environments and tasks, and they will need to allow users to interact wherever he/she is and using whichever mode or combination of modes are most appropriate given the situation, their preferences and the task at hand. Furthermore, unlike traditional library or office settings, users of mobile information access devices will, typically, be subject to much higher levels of interruption and task switching, thus needing very different interface designs.

This workshop aims to be a forum for the presentation and discussion of current research and development of technological and usability aspects of mobile information access. The workshop is interesting for both researchers and practitioners of several different communities, such as information retrieval, digital libraries, HCI, mobile devices, and so on. A wide range of topics are relevant to the objectives and aims of the workshop, in particular (but not limited to):

- information retrieval and filtering,
- user modelling and personalization,
- context awareness,
- new mobile devices,
- nomadic computing,
- ubiquitous computing,
- usability,
- ambient intelligence.

The workshop aims at addressing these (and related) topics both in terms of existing approaches and implementations, and in terms of theoretical foundations and emerging directions of research.

2. HISTORY
Two workshops with the same name and theme were held at Mobile HCI 2003 in Udine, Italy and at Mobile HCI 2004 in Glasgow, UK. Two of the current workshop organisers were in the organising committee of the first MUIA 2003 workshop; all three of them organized the second MUIA 2004 workshop. The workshops were very successful, both in terms of submissions (over 30 papers each) and participants (over 40 participants each). Selected papers of MUIA 2003 were invited to submit a revised and extended version for inclusion in a volume published after the

MobileHCI'06, September 12–15, Helsinki, Finland.
ACM 1-59593-390-5/06/0009.

event [1]. Selected papers of MUIA 2004 were invited to submit a revised and extended version for inclusion in a special issue published after the event [2].

3. PUBLICATION OF PROCEEDINGS

The MUIA 2006 workshop proceedings are going to be made available online. In addition, in a similar way to what was done for the two past MUIA editions, we envisage the preparation of either a special issue of an internationally recognized journal or an edited volume. Selected papers accepted for presentation at the workshop will be invited to submit a revised and extended version.

4. PROGRAM COMMITTEE

The following people, along with others, will serve on the Program Committee.

- Oscar de Bruijn, Imperial College, London, UK.
- George Buchanan, University of Wales, Swansea, UK.
- Fabio Crestani, University of Strathclyde, UK.
- Vincenzo Della Mea, University of Udine, Italy.
- Mark Dunlop, University of Strathclyde, UK
- Peter Froehlich, Telecommunications Research Center Vienna, Austria.
- Gareth Jones, Dublin City University, Ireland.
- Matt Jones, University of Wales, Swansea, UK.
- Steve Jones, University of Waikato, New Zealand.
- Mun-Kew Leong, Laboratories for Information Technology, Singapore.
- Gary Marsden, University of Cape Town, South Africa.
- Stefano Mizzaro, University of Udine, Italy.
- Joerg Roth, University of Applied Sciences, Nuremberg, Germany.

REFERENCES

1. F. Crestani, M. Dunlop, and S. Mizzaro, editors. *Mobile and Ubiquitous Information Access*, volume 2954 of Lecture Notes in Computer Science, Berlin, Germany, 2004. Springer-Verlag.

2. F. Crestani, M. Dunlop, M. Jones, S. Jones, S. Mizzaro, editors. Interactive Mobile Information Access, Special issue of *Personal and Ubiquitous Computing*, 10(4), 2006.

WS7: WORKSHOP ON MOBILE AND UBIQUITOUS INFORMATION ACCESS 2006
@
Mobile HCI 2006, Espoo, Finland

12 September 2006
Kivi, Innopoli 2, SoberIT premises, Tekniikantie 14, Espoo
(See http://www.tkk.fi/Yleista/Yhteystiedot/Ohjeet/kartta.html for a map)

Programme

9:00-9:15 Welcome

9:15-10:15 First session

- "Applying XML Retrieval Methods for Result Document Navigation in Small Screen Devices"
- "Summarisation and novelty in Mobile Information Access"

10.15-10.45 Coffee Break (diner on the ground floor)

10:45-12:15 Second session

- "Seamless Interface Transition Mechanism for Pedestrian Navigation Systems"
- "Designing game logics for dynamic Active Surfaces"
- "A middleware of services for spontaneous short-range ad-hoc network of mobile devices"

12:15-12:55 Open session: discussion on topics related to mobile and ubiquitous information access

12:55-13:00 Closing

Applying XML Retrieval Methods for Result Document Navigation in Small Screen Devices

Paavo Arvola
Department of Information Studies
University of Tampere
Finland
paavo.arvola@uta.fi

Marko Junkkari
Department of Computer Sciences
University of Tampere
Finland
junken@cs.uta.fi

Jaana Kekäläinen
Department of Information Studies
University of Tampere
Finland
jaana.kekalainen@uta.fi

ABSTRACT
Content-oriented XML retrieval is beneficial in reducing user's effort in finding the best parts of a document. This feature is stressed when using limited screen space. The present study combines XML retrieval with the problematic document presentation of small screen devices, and introduces an information retrieval (IR) system including a web user interface for mobile environment. In our approach the resulting documents have been modified by marking up the best matching parts of the document with hyperlinks and anchors. This mark up is based on the initial query expression given by the user. The hyperlink-anchor structure enables retrieval status value browsing within a document aside with conventional browsing. The core of the system is based on TRIX XML IR system.

Categories and Subject Descriptors
H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval – *search process.* H.5.4 [**Information Interfaces and Presentation**]: Hypertext/Hypermedia - *navigation.*

General Terms
Design.

Keywords
XML retrieval, Small Screen Devices.

1. INTRODUCTION
Small screen is one of the major constraints for a mobile device. For example, conventional browsing through a long text document with such a device requires a lot of vertical scrolling. However, in many cases the text document in whole is not needed to satisfy the user's information need. With the space limitation in mind we have developed methods for enhancing the access to the most relevant parts of a document. This is done without breaking up the continuity of the initial textual content presentation of the document. The conventional browsing methods within a document are also available.

MobileHCI'06, September 12, 2006, Espoo, Finland.

1.1 Text Retrieval with Small Screen Devices
Research and many applications within the scope of small screen devices are focused on optimizing the layout of documents for the device. Removing scrolling horizontal wise is a necessity, but reducing vertical scrolling is also crucial. As individual instances of such applications are PowerBrowser [3] and WebTwig [6]. However, in this study we are not interested in transformations of layouts for small screens. Also other constraints related to wireless mobile devices such as low bandwidth, battery capacity, cumbersome text input, and lack of processing power and memory are not of interest in this paper. Instead, this paper deals with how to retrieve relevant information effectively from long text documents especially with a limited screen size.

In addition to optimizing the layout for the device, it is also possible to take the user's personal behavior into account in the content presentation. Within the information retrieval (IR) process, an IR system may use the initial query not only for the selection of result documents but also for the presentation of the results. One possible approach is to use query biased summaries especially in a result list presentation. Research on summarization of resulting documents has also been done in the scope of small screen devices [10]. The summarization or a good presentation of result list is not always enough, though. Despite the summarization, the user has or is willing to access the document itself.

Hence, the initial query expression may have an influence also in the presentation of the result document. For instance SearchMobil [8] and Google Mobile [11] utilize the initial query in result document presentation. SearchMobil displays a thumbnail view of the document to the user. Further, it provides an indication of the most promising parts of the document by highlighting the query terms in the thumbnail view. From the view the user is able to select the best part of the document.

Likewise the SearchMobil, the Google Mobile also highlights (some of the) query keys in the resulting document. In addition, in order to reduce vertical scrolling, the Google Mobile may split the initial (textual) content of the document into several subdocuments returning only the most promising one. However, splitting documents into pieces restricts the possibilities to use conventional browsing within a document, because the initial logical structure of the document is lost.

1.2 XML Information Retrieval

The traditional IR considers a document to be an atomic retrievable unit. Hence, a traditional IR system presents results as whole documents. Instead of treating documents as a single text block, XML mark-up offers an opportunity to utilize the natural structure of documents, enabling a more focused retrieval strategy and allowing a system to return only parts of documents, i.e. XML elements. This approach, namely content-oriented XML retrieval has been widely studied within INEX (Initiative for the Evaluation of XML) since 2002 [5]. The initiative focuses on developing matching and evaluation methods for XML retrieval, and it also contains an interactive track for investigating interaction between users and XML retrieval systems. However, as far as we know, there has not been any element retrieval status value based XML browsing applications or research in context of small screen devices. Despite that, in our opinion it is evident that the paradigm of retrieving elements instead of the whole document is beneficial, when it comes to small screen devices, where the view to a document is very limited.

Likewise in the SearchMobil and Google Mobile, using the initial query keys in document presentation works also as a starting point in our effort to reduce vertical scrolling. But instead of highlighting the keywords only, in our approach different parts of a result document are weighted with a partial match IR system. The basis of weighting is XML retrieval, where each element is given a retrieval status value (i.e. weight) according to the query expression. The primary motivation for this is to reduce the effort of finding relevant content by directing the user to the best parts of the document. This is done by inserting hyperlinks to anchors which, in turn, are placed to the best locations of a document according to the initial query expression. This approach not only reduces user's effort in reduced vertical scrolling but also preserves the original document order and structure of the document. In addition, this allows the user to utilize conventional browsing methods within the document if needed. The anchoring of elements is done simultaneously with rendering XML to standard XHTML format for viewing and browsing in a device independent fashion. When using XML retrieval, our method is optimal especially for content-oriented online XML collections, such as the Wikipedia XML collection used in INEX 2006 [5]. The following chapter gives a detailed view of the system.

2. SYSTEM OVERVIEW

2.1 Two Phase Matching Technique

In order to fetch the best matching elements from the best matching documents, we use a two phase matching technique in the retrieval. In a nutshell, the documents are first sorted according to their retrieval status value, and after that the elements are clustered by the documents. The weighting function is discussed in more detail in Section 2.3.2. First in the search process the user inserts the keywords with the available text input methods in order to perform the search. Phase 1 is plain full document retrieval where, according to the query expression, the system presents a result list having links to the documents in a matching order. Figure 1 illustrates this phase. Phase 2 is element retrieval and it is done for a single document selected by the user.

Now, let us assume the present article (document) to be marked-up with XML. Let us also assume this mark-up to follow a typical representation schema for structured documents with an article-section-subsection-paragraph hierarchy. Then, the root element of this document consists of (i.e. has as child elements) sections 1, 2 and 3 (in addition to abstract, references etc). The sections consist of subsections, which may consist of subsubsections. We could also consider that the descendant elements of the (sub) sections are paragraphs, headings, figure captions or equivalents as siblings. This level is the lowest level to index in our approach, even though there might be subelements below this level, such as style formatting, references and so on.

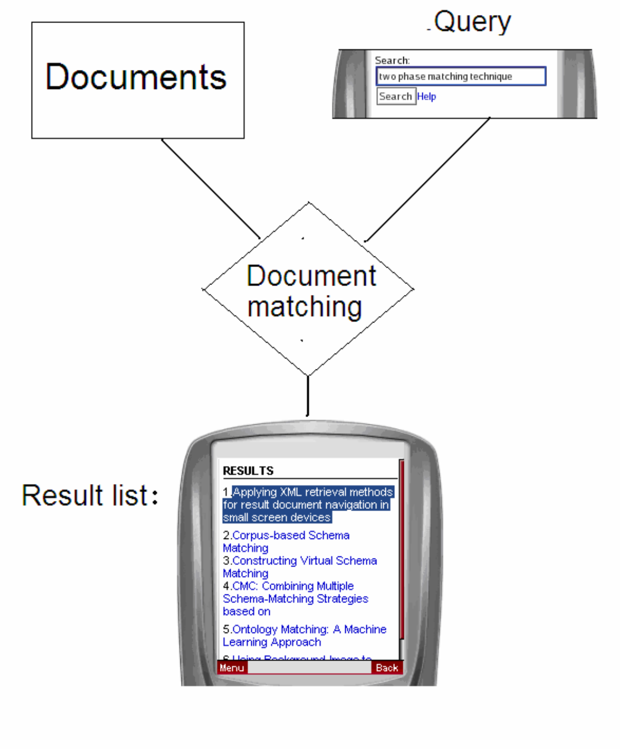

Figure 1: Phase 1 – document ranking

According to the Figure 1 the query: 'two phase matching technique' is forwarded into the IR system. This launches the full document retrieval phase and the system presents a result list, with the current document on top of it. Preferably, the user selects this document from the result list by clicking a pointing link. Clicking the link triggers the Phase 2, which is called *relevant in document retrieval* (Figure 2). In this phase the system marks up the best matching parts of the XML document. Thereafter the system renders the resulting XML document into XHTML document for viewing and browsing. This includes also inserting anchors into the beginning of the best matching parts. Finally, the browser shows the user the beginning of the result document.

In the Phase 2 all elements of the selected document are considered. From the elements the function in Appendix 1 selects a set of elements which fulfils the conditions for relevance mark-up. The conditions:

1. the weights of the elements are bigger than a given threshold value
2. the elements are not structurally overlapping with each other

The threshold value is dependent on the weight of the whole document (i.e. root element). Thus, only elements with a greater weight than the document weight are considered. The second condition ensures excluding an element, which has a smaller weight than its ancestor or descendant. This holds even if the element has a weight greater than the threshold value. For example, let us say that the Sections 2.1 and 2 exceed the threshold value in this document, with the Section 2.1 having a bigger weight. Thus, the Section 2 as a whole is excluded from the results, because it is an ancestor, i.e. overlaps with an element having a greater weight.

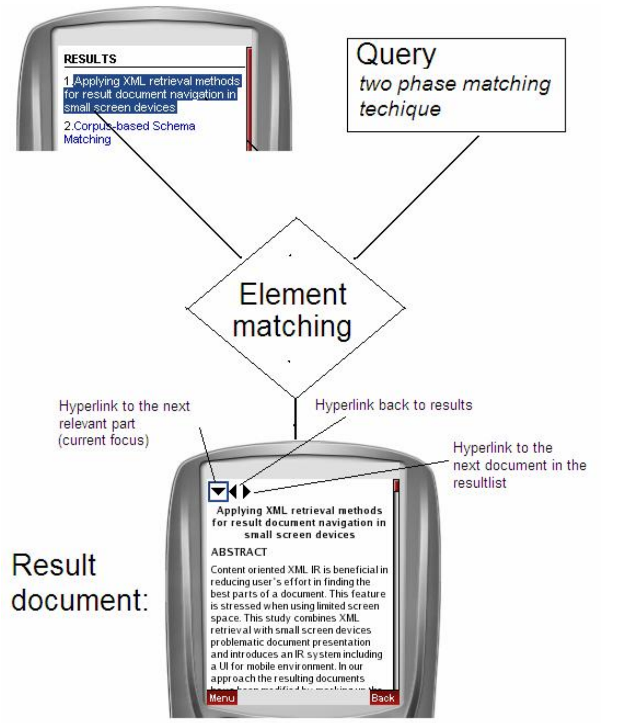

Figure 2: Phase 2 – relevant in document retrieval

2.2 Creating a Matching Chain by Linking the Best Parts with Anchors

After the selection of the best parts of the selected document, we set hyperlinks and anchors into these parts. In order to insert the anchors meaningfully, we have to ensure that:

1. The user is guided to the very best parts of a document
2. The anchor-link structures are not too complex to use

The former one is handled in the function in Appendix 1. Removing the structural overlap in the function is a necessity in response to the latter case as well. However, despite of the removal of the structural overlap, elements fulfilling the threshold value condition may lay very close to each other also in the document order. Inserting links between such elements is definitely disturbing for the user. We say that elements laying too close to each other for relevance anchoring are *window overlapping*. This is measured by counting displaying characters (or rows) from the end of the preceding element down to the start of the following one.

In our approach, in the beginning of a resulting document there are three arrow icons. The first one, an arrow down, is a link to the first anchor in the beginning of the first matching element, which is not window overlapping. The arrow left is a link for getting back to the result list, and the third one directs to the next document. This removes the step of going back to the result list. However, the arrow down is for relevance browsing within a document. By clicking it, the viewer ends up to the point the anchor is at. In the end of the matching element, the arrows are presented again. Now the arrow down is a link to the next matching, not overlapping part of the document.

For instance, the user selects the current document from the result list in Figure 2. The system places two arrow icons as hyperlinks to the top left corner of the result document. In Figure 3 the current user interface focus is on the first hyperlink, which is presented as an arrow down. By clicking the link the viewer ends down to the place the anchor is at. Preferably the anchor is just before a relevant part of the result document. Now the user scrolls and reads the whole section, which is estimated to be relevant by the system. Because there are no further relevant parts in the document at hand, in the end of the section there are only two hyperlinks: back to results and a direct link to the next document in the result list. Otherwise there would also be a link to the next relevant part. Nevertheless, clicking the arrow right icon triggers Phase 2 for the next document in the result list.

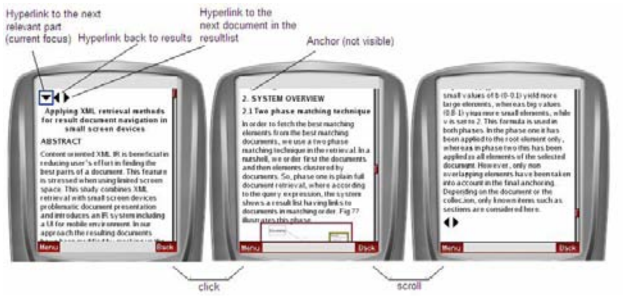

Figure 3: Matching Chain

The function in Appendix 2 is used for creating a matching chain for a document (XMLDoc). Creating a matching chain requires element processing in the document order. Hence, first the set of elements fetched with the function in Appendix 1 is sorted according to the elements' temporal reading positions in the document. By default a browser shows first the top of the document. Thus, we use a cursor variable pointing to the beginning of the document, in other words at a position before the first element containing printable characters. The cursor is utilized for checking whether elements are window overlapping or not. Hence, the cursor is always updated to have the current value showing the number of characters between the document start and the current element. Apropos of that, the functions *start_loc* and *end_loc*, return the start and end cursor positions of given element. The functions *place_link_at* and *place_anchor_at* are used to place links and anchors to the required places in the document, respectively.

2.3 TRIX Indexing and Matching

2.3.1 Structural Indices

The methods described above involving the manipulation of complex XML structures require the application logic layer to function smoothly both vertically and horizontally. The vertical dimension stands for handling ancestor-descendant relationships in element weighting and overlap removal. The horizontal dimension, in turn, means the temporal document order of the elements. To meet these challenges we introduce the TRIX retrieval system [1, 2, 7].

In TRIX the management of structural aspects is based on the structural indices. In Figure 4 there is a tree presentation of an XML document with indices and element names for each node. The idea of structural indices in the context of XML is that the topmost (root) element is indexed by ⟨1⟩ and its children by ⟨1,1⟩, ⟨1,2⟩, ⟨1,3⟩. Further, the children of the element with the index ⟨1,2⟩ are labeled by ⟨1,2,1⟩, ⟨1,2,2⟩ and so on. This kind of indexing enables analyzing of the relationships among elements in a straightforward way. For example, the ancestors of the element labeled by ⟨1,2,2,1⟩ are associated with the indices ⟨1,2,2⟩, ⟨1,2⟩ and ⟨1⟩. In turn, any descendant related to the index ⟨1,2⟩ is labeled by ⟨1,2, ξ ⟩ where ξ is a non-empty part of the index.

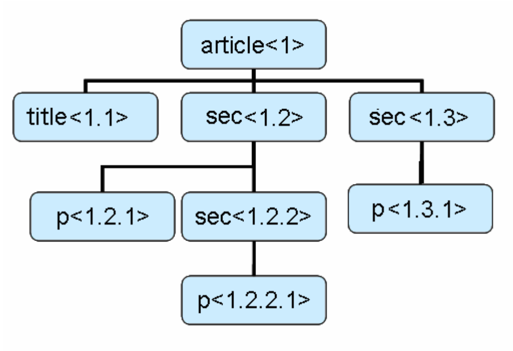

Figure 4: Tree presentation of an XML document with element names and indices

Moreover, because the labeling is executed in the document order for the siblings the indexing works well in figuring out the preceding-following relationship between known indices as well. This feature is needed for sorting the elements by their document order in the function for creating matching chains (Appendix 2). A comparison between indices is straightforward by reading the comparable indices from left to right. For example, it is trivial to say that an element labeled by ⟨1,2,1⟩ is preceding the element ⟨1,3,1⟩. In contrast, with e.g. the path expressions it is not possible to figure out the document order.

2.3.2 The Matching Formula

The matching method for calculating element weights in our approach is based on the TRIX retrieval system. The basic matching formula of TRIX is derived from Okapi BM25 [4, 9] and in our system the formula is used in both phases. In Phase 1 it has been applied to calculate the weights of the root elements only. In other words it has been applied only for weight calculation for indices with length 1. The TRIX matching methods have been proven fairly effective with INEX metrics [5, 1, 2]. In TRIX, the weight of an element with a unique index ξ is an average of query key weights. In the system weight for one key in the element is calculated with a tf*idf formula as follows:

$$w(k,\xi) = \frac{kf_\xi}{kf_\xi + v \cdot \left((1\text{-}b) + b \cdot \frac{\xi f_c}{\xi f_k} \right)} \cdot \frac{log\left(\frac{N}{m}\right)}{log(N)}$$

in which:

- kf_ξ is the number of times k occurs in the ξ element
- N is the total number of content elements [2] (leafs) in the collection
- m is the number of content elements containing k in the collection.
- $\frac{\xi f_c}{\xi f_k}$ is a length normalization component (LNC), in which ξf_c is the number of all descendant content elements of the ξ element and ξf_k is the number of descendant content elements of the ξ element containing k.

The other components of the formula are the constants v and b, which allow us to affect the LNC and tune the typical element size in the result set. In many experiments [7] we have found that small values of b (0-0.1) yield more large elements, whereas big values (0.8-1) yield more small elements, while v is set to 2. Hence, with big values the amount of elements exceeding the threshold value for mark-up is higher than with small values.

3. DISCUSSION AND FUTURE WORK

A method for effective, relevance based browsing with a mobile device has been presented. Our hypothesis is that this method is beneficial for the user

- in finding relevant parts of the document effectively
- in assessing documents relevance effectively

In addition this has been done so that the initial atomic document structure remains as untouchable as possible. In this study, we have discussed XML documents. However, document formats with weaker structure and semantics, like HTML, can also be handled in the manner this study presents. For instance conversions of HTML to XHTML are viable.

Tuning the system in the optimal way is also a challenge and contains numerous attributes to set: Setting properly the window overlapping value, for one. This can be seen as a slightly device dependent matter, for the screen size may vary a lot from a small screen device to another. However, in general, many challenges are related to the classical problems of XML retrieval. For instance: What is the right granularity of returned elements in content-oriented retrieval? Should we search for known items, such as sections, or heterogeneous elements? Should we prefer more specific than exhaustive elements? For now this study lacks user evaluations to enlighten many issues mentioned here and to confirm our hypothesizes. A fruitful test setting would be a comparison between a browsing model presented here and conventional browsing or another method. Nevertheless, in our opinion, the framework presented in this paper gives tools to study the above mentioned issues.

REFERENCES

[1] Arvola, P., Junkkari, M., and Kekäläinen, J. Query evaluation with structural indices. In *Advances in XML Information Retrieval: 4th International Workshop of the Initiative for the Evaluation of XML Retrieval, INEX 2006*, LNCS 3977. 134-145, 2006.

[2] Arvola, P., Junkkari, M., and Kekäläinen, J. Generalized contextualization method for XML information retrieval. In *Proceedings of CIKM '2005*. 20-27, 2005.

[3] Buyukkokten, O., Garcia-Molina, H., Paepcke, A., and Winograd, T. Power Browser: Efficient Web browsing for PDAs, In *Proceedings of CHI '2000*. 430–437, 2000.

[4] Hawking, D., Thistlewaite, P., and Craswell, P. ANU/ACSys TREC-6 experiments. In *Proceedings of TREC-6*, 1998, Retrieved 10.3.2004 from http://trec.nist.gov/ pubs/trec6/papers/anu.ps

[5] INEX 2006: Initiative for the Evaluation of XML, 2006, Available at URL: http://inex.is.informatik.uni-duisburg.de/2006/

[6] Jones, M., Buchanan, G., and Mohd-Nasir, N. Evaluation of WebTwig—a site outliner for handheld Web access. In *Proceedings of International Symposium on Handheld and Ubiquitous Computing*, LNCS 1707. 343–345, 1999.

[7] Kekäläinen, J., Junkkari, M., Arvola, P., and Aalto, T. TRIX 2004: Struggling with the overlap, In *Advances in XML Information Retrieval: Third International Workshop of the Initiative for the Evaluation of XML Retrieval, INEX 2004*, LNCS 3493. 127-139, 2005.

[8] Milic-Frayling, N., Sommerer, R., Rodden K., and Blackwell, A. F. SearchMobil: Web viewing and search for mobile device. In *WWW '2003(Posters)*. 2003.

[9] Robertson S.E., Walker, S., Jones, S., Hancock-Beaulieu, M.M., and Gatford, M. Okapi at TREC-3. In *NIST Special Publication 500-226: Overview of the Third Text REtrieval Conference (TREC-3)*. 1994, Retrieved 21.11.2004 from http://trec.nist.gov/pubs/trec3/papers/city.ps.gz

[10] Sweeney S, and Crestani, F. Supporting searching on small screen devices using summarization. In *Proceedings of Mobile HCI 2003 International Workshop*, LNCS 2954. 187-201, 2003.

[11] http://www.google.com/xhtml, referred 21.6.2006

Appendix 1: A function for best matching elements extraction

(The code blocks have been separated using the tabular expression.)

```
Function:
EXTRACT_BEST_MATCHING_ELEMENTS

Input:
S: Set of ordered pair of values <element, weight>
within a document.
TVM: Threshold value for mark-up

Output:
CM: Candidate elements for mark-up.

CM := empty set;
sort_by_weight(S);
for each <ind,w> in S
  if(w > TVM)
    if(!contains_overlapping(ind, CM))
      add ind to CM;
  else
    break;
return CM;
```

Appendix 2: A function for creating matching chains

(The code blocks have been separated using the tabular expression.)

```
Function:
CREATE_MATCHING_CHAIN

Input:
CM: Candidates for mark-up
(from EXTRACT_BEST_MATCHING_ELEMENTS)
NextDoc: Address of the next document in the
result list
XMLDoc: the XML document the chain should be
created for

Output:
XMLDoc: updated XMLDoc with hyperlinks and anchors

if(CM is not empty)
  sort_by_document_order(CM);
  cursor := start(XMLDoc);
  anchor_id := 0;

  for each ind in CM
    if(!window_overlapping(cursor, start_loc(ind,
    XMLDoc))
      place_link_at(cursor, anchor_id, XMLDoc);
      place_anchor_at(start_loc(ind,     XMLDoc),
      anchor_id, XMLDoc);
      cursor := end_loc(ind, XMLDoc);
      anchor_id := anchor_id + 1;
    else
      cursor := end_loc(ind, XMLDoc);
  place_link_at(cursor, NextDoc, XMLDoc);
return XMLDoc;
```

Summarisation and novelty in Mobile Information Access

Simon Sweeney
Dept. Computer and Information Sciences
University of Strathclyde
Glasgow, Scotland, UK
simon@cis.strath.ac.uk

Fabio Crestani
Dept. Computer and Information Sciences
University of Strathclyde
Glasgow, Scotland, UK
fabioc@cis.strath.ac.uk

David E. Losada
Depto. de Electrónica y Computacion
Universidad de Santiago de Compostela, Spain
dlosada@usc.es

ABSTRACT

The paper presents a user study which investigates the effects of incorporating novelty detection in automatic text summarisation. The motivation being the need to provide access to information that is tailored to small screen displays. Automatic text summarisation offers a means to deliver device-friendly content. An effective summary could be one that includes only new information. However, a consequence of focusing exclusively on novel parts may result in a loss of context, which may have an impact on the ability to correctly interpret the meaning of a summary given the source document. In the user study we compares two strategies to produce summaries that incorporate novelty in different ways; an incremental summary and a constant length summary. The aim is to establish whether a summary that contains only novel sentences provides sufficient basis to determine relevance of a document, or do we need to include additional sentences to provide context. Findings from the study seem to suggest that there is minimal difference in performance for the tasks we set our users. Therefore, for the case of mobile information access a summary that contains only novel information would be more appropriate.

1. INTRODUCTION

The continued development of mobile device technologies, their supporting infrastructures and associated services is important to meet the anytime, anywhere information access demands of today's users. The growing need to deliver information on request, in a form that can be readily and easily digested on the move, continues to be a challenge. This is despite improvements in device handsets with greater battery life, support for a greater range of applications with Java compatibility (J2ME[1]) and high speed 3G[2] network access. A key feature in providing access to information in a mobile context are the limitations on interaction, in particular the size of screen displays. While font rendering and screen resolutions can, and do improve, there remains an upper threshold dictated by what is legible to the human eye, given the inherent requirements on device sizes to be portable. Therefore, the design of content for mobile delivery remains an important factor.

Automatic summarisation can be employed to condense a document, presenting only the important parts of a full text thereby reducing the need to refer to the source document. Adopting such an approach removes the need to manually construct device friendly content, while offering means to tailor summaries to be informative or indicative [4]. The intended use, and therefore type of summary employed is an important characteristic, another is the length of the summary in relation to the display screen size. In terms of an optimal summary size, according to findings of previous work [11] it would appear that short summaries (7% of the document length) perform well for a range of display screen sizes.

Assuming then short summaries as the basis for length, are there other factors that could improve the effectiveness of summaries, particularly in light of the task of identifying items of interest, or relevant content? An effective way to produce a short summary could be to include only those parts that contain novel information. However, producing a summary that contains only novel sentences (assuming we employ an approach that uses sentence extraction) may result in a loss of context. Therefore, will the novel sentences alone provide sufficient basis to determine relevance, or do we need to include additional sentences to provide context? In this sense, we refer to context as the background, or more specifically to information previously digested from a source text. If we consider that the full text of a document consists of 3 types of sentences: (i) relevant sentences, (ii) novel sentences and the (iii) remaining sentences. A summary based on relevance will have sentences that contain content relevant to an information need. In contrast a summary based on novelty will contain only sentences that are both relevant and novel. Within a summary based on relevance there may be redundant information since sentences appearing later in the summary may repeat earlier concepts.

[1]The Java^{TM}2 Platform, Micro Edition (Java ME) provides a robust, flexible environment for applications running on consumer devices, such as mobile phones.

[2]3G refers to the third generation of developments in wireless technology, especially mobile communications, which support much higher data rates intended for applications other than voice.

In this paper we consider summarisation with novelty detection, where information is not only condensed but also an attempt is made to remove redundancy. We adopt the same strategy as we employed previously to produce query-biased summaries [10, 11, 12], with the difference that given an initial summary, subsequent summaries will not only be query-biased (presenting those sentences that are relevant to the query) but also take account of novelty by reflecting the history of previously seen summaries. The scenario that describes our experimental approach is as follows: given an interest in a topic describing a point of view, a user wishes to satisfy further interest by exploring a number of document summaries to identify relevant documents. We adopt two strategies to produce summaries that incorporate novelty in different ways; an incremental summary ($SumN_i$) and a constant length summary ($SumN_c$). We compare the performance of groups of users with each of the test systems to gain insight into the following research question. Will a summary that contains only novel sentences provide sufficient basis to determine relevance of a document, or do we need to include additional sentences in the summary to provide context?

The remainder of the paper is as follows. First, we outline briefly work related to novelty detection and how it can be combined with summarisation (section 2). We then describe the process of how we generated the novel summaries for our experiment (section 3). Next, we present details of the experiment we carried out (section 4), and some of the results we collected (section 5). Finally, we conclude the paper with a short discussion of the implications of our findings in combining summarisation with novelty and indicate directions for future work (section 6).

2. RELATED WORK

A large proportion of work in novelty detection has been carried out in Topic Detection and Tracking (TDT) for the purposes of new event, or first story detection [2, 1, 5, 7, 9, 15, 16]. Typically, work in this area applies TDT to news stories where the concern is event-based novelty detection. The emphasis then is on detecting overlaps in event coverage and to identify whether two news stories cover the same event. It is often the case that many of the techniques applied in TDT, to detect events, make use of temporal clues and other features that are particular to the structure of stories in news reporting.

Another area where novelty detection research has been actively pursued is at the Novelty tracks of the Text REtrieval Conferences '02-04 (TREC)[3]. In contrast to TDT, the Novelty track is concerned with topic-based novelty detection. Here, the focus is novelty detection at the sentence level where the importance is not only on finding whether two sentences discuss the same topic, but also identifying where there is new information on the topic. For the track participants are required to build a novel ranked list of relevant sentences, which consists of a two part process: (i) identify relevant sentences from a set of retrieved documents for a topic; and (ii) using the list of relevant sentences, identify those that contain new information. It is implicitly assumed that the process of topic learning happens within the task, and effects of prior knowledge are ignored. Techniques that have been demonstrated at the Novelty track include those that are word-based and those that make use of other textual features. Using TREC'02 data, UMass experimented with a range of techniques from a simple count of new words (we adopt a similar approach in this paper) to more complex techniques that use language models and Kullback-Leibler (KL) divergence with different smoothing strategies [3]. More recent approaches have investigated features in sentences, such as, various types of word combination patterns ranging from named entities and phrases, to other natural language structures.

The combination of summarisation paired with novelty detection is not a new concept. Early work combining query-relevance and information-novelty was in [6], where Maximal Marginal Relevance (MMR) was used to reduce redundancy while maintaining query relevance in re-ranking retrieved documents and in selecting appropriate passages for text summarisation. For the purposes of this paper we approach novelty detection in a slightly different way. Rather than treat each sentence independently and assess novelty at a sentence level, we instead apply novelty detection at a summary level, on previously seen summaries. In this way we provide the most relevant important parts of the document in response to the query first, any subsequent requests for more content, present only new information with respect to what has been already seen.

3. GENERATING THE NOVEL SUMMARIES

3.1 Query-biased summaries

A detailed description of the methods used to build the query-biased summaries can be found in [10, 11, 12, 13, 14]. The summarisation system employed in the experiment we report is similar to one described in [11]. The system uses a number of sentence extraction methods [8] that utilise information both from the documents of the collection and from the queries used.

The underlying process relies on scoring sentences in a document to reflect their importance for inclusion in the document's summary. Scores are assigned based on evidence from the structural organisation of the document (title, leading text and heading scores), within document term-frequency information (significant term score) and the presence of query terms (query score). The final score for a sentence is computed as the sum of the partial scores.

The summary for a document is generated by selecting the desired number of top-scoring sentences, and outputting them in the order in which they appear in the original document. Summary length, the number of sentences picked, can be controlled to restrict the level of information a user would be presented with in relation to the original document.

3.2 Summaries used in the experiment

We shall now focus on those parts concerning the integration of novelty in the summary generation process. To provide a point of reference for the rest of this section we first illustrate the complete range of summaries built for the user study. Figure 1 shows both the levels and types of summaries prepared. Reading in a vertical perspective the diagram can

[3]For a more details of the TREC Novelty track, and listing of other techniques submitted to the (more recent) novelty tracks refer to http://trec.nist.gov/pubs.html.

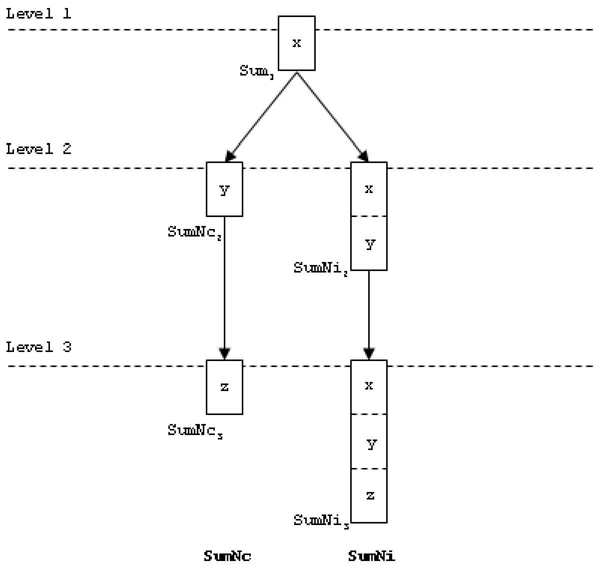

Figure 1: Illustrating the summary types built for the user study.

be divided along an imaginary central axis (beneath Sum_1) to show two approaches: one that combines novelty with constant length (left of centre), $SumN_c$; and the other that incorporates novelty with increasing length (right of centre), $SumN_i$. The horizontally dotted lines indicates increments in summary, which depending on their type may increase in length ($SumN_{i_2}$, $SumN_{i_3}$) or maintain a constant length ($SumN_{c_2}$, $SumN_{c_3}$). Example summaries for a sample document are given in Figure 2.

Key decisions made at the outset, which influence the production of summaries, relate to the number of summary levels and the length of summaries. We restrict the number of summary levels to 3, primarily to avoid overburdening users' in the experimental tasks. Also, including the document title with summaries we aim to assist users in associating summary levels with the source text. In terms of summary length, for each document a number of sentences equal to 7% of its length (with a minimum of 2 sentences and maximum of 6 sentences) were used. This is supported by our previous experiments with summary length, where we found short summaries performed well in similar tasks [11].

A further feature shown in the diagram (referring to Figure 1) is an indication of differences in how information content is presented. In the figure, x represents information gained from the summary at level 1. The contrasting methods of delivery are apparent then at levels 2 and 3. For $SumN_i$, levels 2 and 3 consists of the union of what was seen previously and the additional new information, whereas for $SumN_c$ only the new information is shown. The overall pattern then is that the same information is conveyed in both cases and only the method of delivery is varied.

3.3 Novel summaries

In a comparison of techniques to detect novelty at a sentence level, Alan et al. [3] found that simple word counting methods (e.g. *NewWords*) performed no worse than other methods tested; indeed performing best in the case where non-relevant sentences were present. This is the most realistic case when considering use in a real environment. We therefore make use of a similar approach to *NewWords* as our first attempt to take account of novelty when building summaries.

We shall now outline the process of building the summaries. We start from a ranked set of sentences, $s_{r_1}, s_{r_2}, ..., s_{r_n}$, obtained by the methods explained earlier (refer to section 3.1). This rank is used to produce an initial summary, Sum_1, (relevance-based only) whose length is l_1, determined from the original document length. The idea is that Sum_1 is the first summary presented to the user and, then, she/he can ask to see more information. The first method increases length (N_i) and increments the size of the next summary to be l_2, $= 2 * l_1$ producing a new summary where some the material which appeared in Sum_1 is also present in $SumN_{i_2}$. The second method maintains a constant length (N_c) and takes a very different approach producing a new summary, $SumN_{c_2}$, whose size l_2 is equal to l_1. The idea here is to avoid the presentation of material that the user has already seen and instead focus on the sentences which, in the original (relevance-based) rank, were ranked right after the ones selected for Sum_1. That is, $SumN_{c_2}$ will be composed of sentences selected from $s_{r_{l_1+1}}, s_{r_{l_1+2}}, ..., s_{r_n}$.

The generation process for both $SumN_i$ and $SumN_c$ is for the most part the same with the key difference at the final stage. Taking the original relevance-based rank we first establish a list of sentences to form the history log of previously seen summary text and a set of candidate sentences, whose relevance score is greater than zero.

To compute the novelty scores for candidate sentences we generate a WordsSeen list from the history log. The WordsSeen list remains static and is not updated with new words identified as candidate sentences are evaluated. The score is based on the proportion of new words with respect to the WordsSeen and compared to all words in the sentence. We compute this as the count of the number of new words, divided by the sentence size, including only those words in the sentence that have been stopped and stemmed. Weighting is applied to the novelty score to emphasize novelty scoring over the previous scoring matrix for a sentence. The final score for a candidate sentence is then the sum of the novelty score combined with the existing relevance score. Candidate sentences are then ranked according to the combined score.

On the basis of the score ranking and on the required size, a summary is produced. It is at this stage that the generation process differs depending on the summary type. The difference in strategy is as below:

- **Increasing length summaries** A combination of the sentences taken from the history log, and the top N scoring candidate sentences form the final summary. Therefore, given $SumN_{i_1} = x$, then $SumN_{i_2} = x + y$ and $SumN_{i_3} = x + y + z$, where x, y and z represent the information content of summaries;
- **Constant length summaries** The top N scoring candidate sentences form the final summary. Given

$SumN_{i_1} = x$, then $SumN_{i_2} = y$ and $SumN_{i_3} = z$.

The final stage of the process involves summary sentences being reordered as they occurred in the original document.

3.4 Sample summaries for a typical document

To illustrate the described process for building novel summaries we now provide an example for a typical document, e.g. APW19981020.1368. Table 1 shows the output of the summarisation processes; highlighting the difference between the summaries generated using the different settings for a sample document. For each distinct level there are the associated sentence identifiers, which are assigned during an initial phase of summarisation process. The differences between Sum_c and Sum_i are clearly shown, with the increasing length summary containing previously seen summary sentences. Also evident is the shared seed summary at level 1 which is generic (Sum_1, shown previously in Figure 1).

Figure 2 contains the summaries generated for the sample document. Annotations marking the type of summary, have been added for the purposes of reporting here. Also, for easy cross-referencing with Table 1, sentence identifiers have been included in the summary text.

Level	$SumN_i$	$SumN_c$
1	0,1,5	0,1,5
2	0,1,5,15,16,19	15,16,19
3	0,1,5,7,8,15,16,19,20	7,8,20

Table 1: Listings of summary sentence IDs for summaries of a typical document, e.g. APW19981020.1368.

Level 1: Sum_1(Generic)

UN warns that arrears issue will again threaten US vote, future

- UNITED NATIONS (AP) _ Washington's continued failure to pay its bills will again threaten its vote in the General Assembly next year and will lead to a backlash against enacting U.S.-demanded reforms, the United Nations warned. (0)
- Secretary-General Kofi Annan, who has been outspoken in the past week in criticizing the United States, said in a statement that the U.S. Congress and administration had reneged on personal promises to pay its bills this budget season. (1)
- But Congress failed to act on a separate spending bill concerning the dlrs 1.3 million the United Nations says the United States owes in back payments. (5)

Level 2: $SumNc_2$

UN warns that arrears issue will again threaten US vote, future

- The United States now accounts for two-thirds of the outstanding U.N. arrears. (15)
- The United Nations has managed to keep its operations going by borrowing money from a separate peacekeeping fund once the regular budget runs out, usually in September. (16)
- ``Where we stand today is that a large number of other member states are underwriting the United States' dues in the United Nations by agreeing to permit us to borrow from peacekeeping funds that are really owed to them,'' the official said. (19)

Level 3: $SumNc_3$

UN warns that arrears issue will again threaten US vote, future

- President Bill Clinton has threatened to veto the arrears bill because it contains a provision denying U.S. contributions to international family-planning organizations that advocate abortion rights. (7)
- In a related issue, Congress failed to allot any funding for the U.N. Population Fund _ a decision that will mean ``the unnecessary death and suffering of women who are deprived of the information and means to plan their families,'' the agency's executive director, Nafis Sadik said in a statement. (8)
- Annan has suggested asking the General Assembly to decide whether it wants to continue the practice, but the issue hasn't been placed on the assembly's agenda yet. (20)

Figure 2: Summary text for a typical document, e.g. APW19981020.1368 ($SumN_c$ only).

4. EXPERIMENTAL SETTINGS

4.1 The Test Collection

The documents used were taken from the AQUAINT collection used at the Novelty track, consisting of newswire stories from the New York Times (NYT), Associated Press Wire (APW) and Xinhua News Agency (XIE). Topics selected were used both as a data source and as a standard against which the users' relevance assessments were compared, enabling precision and recall figures to be calculated. For this last purpose the relevance assessments that are part of the collection and that were made by TREC assessors are used (refer to the discussion at the end of section 4.4).

A total of 5 randomly selected TREC queries and for each query, the 10 top-ranking documents were used as an input to the summarisation system. To ensure suitability of the documents for the experiment, a minimum of 4-5 relevant documents were present in each test set. The test collection then consisted of a total of 50 news articles.

4.2 Experimental Measures

The experimental measures used to assess the effectiveness of user relevance judgements were the *time to complete the task* and *accuracy*. We quantify accuracy as precision, recall and decision-correctness. In the experiment we focused on the variation of these measures in relation to the different experimental conditions ($SumN_i$ and $SumN_c$). This is in contrast to the absolute values normally used in information retrieval (IR) research.

We define *precision* (P) as the number of documents marked correctly as relevant (in other words, found to be relevant in agreement with the TREC judges' assessments) out of the total number of documents marked. This definition corresponds to the standard definition of precision. *Recall* (R) is defined as the number of documents marked correctly as relevant out of the total number of relevant documents seen. A further measure we used to quantify the accuracy of a user's judgment was *decision-correctness* (DC), that is the user ability to identify correctly both the relevant document and the non-relevant (irrelevant) documents. We define decision-correctness as the sum of the number of documents marked correctly as relevant, plus the number of documents correctly marked as non-relevant out of the total number of documents marked for that query.

4.3 Experimental Design

For the experiment we recruited 20 users to form four experimental groups ($Group_1$ to $Group_4$). Participants were recruited from members of staff and postgraduate students of the Department of Computer and Information Sciences at the University of Strathclyde.

Order	Group			
	1	2	3	4
1	$SumB_i$	$SumN_i$	$SumB_c$	$SumN_c$
2	$SumN_c$	$SumB_c$	$SumN_i$	$SumB_i$
3	$SumB_i$	$SumN_i$	$SumB_c$	$SumN_c$
...	...	...	...	...

Table 2: Assignment of summaries to the experimental user groups ($Group_1$: users 1-5; $Group_2$: users 6-10, $Group_3$: users 11-15; and $Group_4$: users 16-20).

The experiment was divided into two sessions with two of the user groups completing the experimental tasks in each of the sessions. Care was taken to ensure consistency in the conditions experienced by all groups.

For the experiment, each user was given 5 queries, and for each query, the top 10 retrieved documents. These 10 documents were represented as 5 documents summarised using technique which included novelty, $SumN$, and 5 summarised using a baseline technique that did not use novelty detection, $SumB$. The baseline summaries were in fact query-biased summaries. For each document there are three levels summary as, Sum_1, Sum_2, and Sum_3 (Figure 1).

The experiment was conducted in such a way that each user group experienced using the different system settings. The system configurations that were shown to users alternated so as to mix the different summary types. For example, the first document might be $SumB_i$, then the next document $SumN_c$, and then $SumB_i$ and so on. Table 2 depicts the experimental conditions used. The allocation of summary types were assigned in such a way as to avoid users' gaining preference for a type of summary over another. Both the user group and session assignments were selected randomly.

To summarise, each user was given a total of 50 documents to work through, each represented by 3 summaries. At the end of the experiment a user had visited a total of 150 document summaries (75 with novelty $SumN$ and 75 without novelty $SumB$).

4.4 Experimental Procedure

Each user was presented with a retrieved document list in response to a simulated query (TREC topic) and tasked with identifying correctly relevant and non-relevant documents for that particular query. Further, so as not to biased quick decisions, users were informed that their performance scores would be penalised if they made mistakes. The information presented for each document was the automatically generated summaries.

Following an initial briefing about the experimental process and instructions by the experimenter, users were presented with a list of 5 queries. To start the experiment users were asked to select the first query from the list. The title and the description of each query (i.e., the 'title' and 'description' fields of the respective TREC topic[4]) provided the necessary background to their 'information need' to allow users to make relevance judgements. For each query, an initial period was allowed to read and digest the query details. Following this, the first of the 10 documents were presented to users and timing for that specific document started.

Users were shown documents from the list where the content for a document consisted of the level 1, 2 and 3 summaries (e.g. $SumN_{c1}$, $SumN_{c2}$, and $SumN_{c3}$). This order, based on level, was the sequence in which summaries were presented. Having seen summary $SumN_{c3}$ users' were required to make a decision as to whether to mark the document as relevant, or non-relevant. After indicating their decision users were presented with the first summary of the next document. On completing the final document for a query users were returned to the list of queries. The process was repeated until all queries have been evaluated.

Once all query tasks were complete, a questionnaire was given to the users. The key quantative data of interest: user decisions, and the individual summary timing data, were recorded in logs file.

Some shortcomings to the methodology used in our experiment relate to the use of TREC topics to simulate information needs imposes an unnatural overhead on users to carry out relevance assessments. Added to this is the use of TREC relevance assessments as the basis for comparing user decisions in order to obtain precision and recall values. However, despite this limitation the same experimental conditions applied to all of the test systems. A further factor imposed as part of the experimental design corresponds to permitting users to make relevance decisions only after viewing all of the summaries, and not at individual summary levels. In removing the ability to make an early decision it could be argued that we are not giving users a true representation of the case for 'show me more'. The motivation for the restriction was to ensure a consistent basis for comparing all systems. It was an assumption of the study that users would make better decisions if shown more of the original document contents. With this in mind we are therefore evaluating the best strategy for showing the user more.

5. RESULTS

We now report the results of the experiment described in the previous section. Table 3 provides a view of results in the context of the experimental methodology, depicting the allocation of users to groups and associated summary types. Focusing on the different summary settings the relative performance across the experimental queries in terms of DC, P, R and average time spent is shown.

The results show a slight increase in DC and R performance with summaries that provide novelty with additional context, $SumN_i$. For P, the baseline summary with a constant length, $SumB_c$, performs best. However, the margins of improvement are somewhat minimal. Carrying out appropriate statistical tests (Chi-Squared test) we found no significance difference in the overall results for the different approaches.

Interestingly, the margin of difference in the time spent on $SumN_i$ compared to $SumN_c$ does not agree with what we might normally expect. The additional effort to digest a

[4]Examples of TREC topics are available at http://trec.nist.gov/data/testq_eng.html

Group	Type	DC	P	R	Time (secs)
1 & 4	$SumB_i$	0.764	0.822	0.845	66
2 & 3	$SumB_c$	0.768	0.850	0.798	53
2 & 3	$SumN_i$	0.776	0.809	0.852	64
1 & 4	$SumN_c$	0.760	0.803	0.752	63

Table 3: Average performance across all queries for the different summary types based on techniques assigned to users.

longer summary (e.g. $SumN_i$) we would expect to translate into more time spent compared to shorter summaries (e.g. $SumN_c$). However, the results show that is not necessarily the case and the times are instead very similar. A possible reason to explain the similarity could be that users may skim the longer summaries, glancing over content already seen, and instead focusing on the new parts. The baseline summaries follow a more expected pattern, though again the margin of difference is small.

If we consider results at a query level, then Table 4 shows performance for each query separately. In terms of DC and P then performance levels show a degree of alignment according to whether they contain novelty, or are from the baseline. On the whole there is a pattern of improvement over the first query, with performance levelling out for intermediate queries and a drop in performance for the final query. However, an exception to this pattern is DC for the baseline approaches in the second query seen by users, query 58 (Q58), where there is a drop in performance. For R, the different summary types share a similar performance profile, with a greater spread in the range of performance levels. However, $SumN_c$, performs noticeably worse in R compared to all other approaches, particularly in query 78 (Q78). Comparing queries in terms of the average time spent, the first query takes the greatest amount of time with a decrease in time spent on all other queries. Interestingly, despite spending less time, users perform no worse in making relevance decisions for the later queries. This may be attributed to learning effects as users become more efficient in completing experimental tasks. Beyond the second query there is little variation in the times for the remaining queries, which may suggest a threshold in task efficiency. An explanation for the fluctuation in observed query level performance could be a period of learning as users become familiar with the experimental task. This pattern may also be observed at a document level for queries, as users' refine their interpretations of relevance. The performance drop for the final query may be explained by an element of user fatigue. Other factors that may help explain variations in performance for queries being the degree of query topic difficulty, and the language and writing style of documents.

The average length of documents for the queries is shown in Table 5. The average length of a summary (for the generic Sum_1 at level 1) being 6 sentences. Table 6 provides some indication of the range in the lengths of summary used in the experiment.

In summary, the results from the user study suggest that there is little difference in performance (DC, P, R and in time spent viewing content) between novel summaries that include context ($SumN_i$) and those that contain only novel information ($SumN_c$). Since the same level of performance is achieved using both strategies then for the case of mobile information access, a novel constant length summary ($SumN_c$) is best. Therefore, for the point of view of mobile information access, given issues of bandwidth, we can concluded that an effective way to produce a short summary is to build one that includes only novel information. Other factors that support a short summary include: reduced transmissions costs, both financially for pay-per-view content and in bandwidth usage; less navigation requirements in terms of scrolling and paging; finally, less cognitive effort to assimilate the information contained in a summary due to a smaller amount of text to digest. However, the lack of improvement over the baseline does place doubt over the merit of building novel summaries and will require more investigation.

Query	Type	DC	P	R	Time (secs)
54	$SumB_i$	0.700	0.758	0.850	103
	$SumB_c$	0.760	0.868	0.783	89
	$SumN_i$	0.640	0.760	0.675	104
	$SumN_c$	0.660	0.800	0.700	106
58	$SumB_i$	0.600	0.750	0.750	72
	$SumB_c$	0.600	0.733	0.725	55
	$SumN_i$	0.720	0.733	0.875	65
	$SumN_c$	0.740	0.733	0.800	61
76	$SumB_i$	0.920	0.900	0.967	43
	$SumB_c$	0.940	0.942	0.950	38
	$SumN_i$	0.920	0.900	0.950	46
	$SumN_c$	0.900	0.917	0.850	50
78	$SumB_i$	0.900	1.000	0.875	55
	$SumB_c$	0.820	0.950	0.800	41
	$SumN_i$	0.920	0.950	0.925	49
	$SumN_c$	0.820	0.833	0.675	45
84	$SumB_i$	0.700	0.702	0.783	57
	$SumB_c$	0.720	0.758	0.733	44
	$SumN_i$	0.680	0.700	0.833	54
	$SumN_c$	0.680	0.733	0.733	51

Table 4: Average performance for individual queries for summary types based on techniques commonly seen by users.

	Q54	Q58	Q76	Q78	Q84
Avg. document length	55	47	54	42	49

Table 5: Average document length (in sentences) for queries in the experiment.

6. CONCLUSIONS AND FUTURE WORK

Automatic text summarisation is a potential solution to achieving device-friendly content for devices that have limited display screens. An effective way to produce a short summary maybe to include only novel information. However, producing a summary that only contains novel sentences (assuming we employ sentence extraction to build summaries) might imply a loss of context.

In this paper we considered summarisation with novelty de-

Summary Size	Count	% of total
Long (5 and 6 sentences)	254	63.5
Medium (3 and 4 sentences)	138	34.5
Short (2 sentences)	8	2
Total	400	-

Table 6: Range of summary lengths (in sentences) generated for the experiment.

tection, where information is not only condensed but also attempt is made to remove redundancy. We adopted two strategies to produce summaries that incorporate novelty in different ways; an incremental summary ($SumN_i$) and a constant length summary ($SumN_c$). We compared the performance of groups of users with each of the test systems. The aim was to establish whether a summary that contains only novel sentences provides sufficient basis to determine relevance of a document, or do we need to include additional sentences in the summary to provide context?

Findings from the user study suggest that there is little difference in performance (DC, P and R) between novel summaries that include context ($SumN_i$) and those that contain only novel information ($SumN_c$). Therefore, for mobile information access where issues of bandwidth and screen size are paramount then we can conclude that an effective way to produce a short summary is to build one that includes only novel information. However, the performance of the baseline summaries, for the task we set our users, questions the benefits of using novel summaries.

Extensions to the work we have presented include investigating the performance of a more refined approach to novelty detection beyond a simple count of new words. In addition, a further point of interest being to study the effects of permitting users to make decisions at any levels; to investigate summary level preference and if there is a corresponding impact on accuracy.

Acknowledgements

We would like to thank all the participants who took part in the user study. This work is supported by the EU Commission under the IST Project PErsonalised News content programminG (PENG) (IST-004597). More information about PENG can be found at http://www.peng-project.org/.

David E. Losada thanks the support from the “Ramón y Cajal” program, whose funds come from “Ministerio de Educación y Ciencia” and the FEDER program. His research is also partially supported by projects TIN2005-08521-C02-01 (from “Ministerio de Educación y Ciencia”) and PGIDT03PXIC10501PN (from “Xunta de Galicia”).

7. REFERENCES

[1] J. Allan, V. Lavrenko, and H. Jin. First story detection in TDT is hard. In *Proceedings of ACM CIKM'00*, pages 374–381, McLean, Virginia, USA, November 2000. ACM Press.

[2] J. Allan, R. Papka, and V. Lavrenko. On-line new event detection and tracking. In *Proceedings of ACM SIGIR'98*, pages 37–45, Melbourne, Australia, August 1998. ACM Press.

[3] J. Allan, C. Wade, and A. Bolivar. Retrieval and novelty detection at the sentence level. In *Proceedings of ACM SIGIR'03*, pages 314–321, Toronto, Canada, July 2003. ACM Press.

[4] R. Brandow, K. Mitze, and L. Rau. Automatic condensation of electronic publications by sentence selection. *Information Processing and Management*, 31(5):675Ű–685, 1995.

[5] T. Brants and F. Chen. A system for new event detection. In *Proceedings of ACM SIGIR'03*, pages 330–337, Toronto, Canada, July 2003. ACM Press.

[6] J. Carbonell and J. Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of ACM SIGIR'98*, pages 335–336, Melbourne, Australia, August 1998. ACM Press.

[7] M. Franz, A. Ittycheriah, J. S. McCarley, and T. Ward. First Story Detection: Combining Similarity and Novelty Based Approaches. Topic Detection and Tracking Workshop'01, 2001.

[8] C. Paice. Constructing literature abstracts by computer: techniques and prospects. *Information Processing and Management*, 26(1):171–Ű186, 1990.

[9] N. Stokes and J. Carthy. First Story Detection using a Composite Document Representation. In *Proceedings of ACL HLT'01*, pages 1–8, Morristown, NJ, USA, 2001. Association for Computational Linguistics.

[10] S. Sweeney and F. Crestani. Supporting Searching on Small Screen Devices using Summarisation. In *Proceedings of Mobile HCI'03 International Workshop*, pages 187–201, Udine, Italy, September 2003. Lecture Notes in Computer Science, volume 2954, Springer, Berlin.

[11] S. Sweeney and F. Crestani. Effective search results summary size and device screen size: Is there a relationship? *Information Processing and Management*, 42(4):1056–1074, 2006.

[12] S. Sweeney, F. Crestani, and A. Tombros. Mobile Delivery of News using Hierarchically Query-Biased Summaries. In *Proceedings of ACM SAC'02*, pages 634–639, Madrid, Spain, March 2002.

[13] A. Tombros and F. Crestani. Users's perception of relevance of spoken documents. *Journal of the American Society for Information Science*, 51(9):929–939, 2000.

[14] A. Tombros and M. Sanderson. Advantages of query biased summaries in information retrieval. In *Proceedings of ACM SIGIR'98*, pages 2Ű–10, Melbourne, Australia, August 1998.

[15] Y. Yang, T. Pierce, and J. Carbonell. A study of retrospective and on-line event detection. In *Proceedings of ACM SIGIR'98*, pages 28–36, Melbourne, Australia, August 1998. ACM Press.

[16] Y. Yang, J. Zhang, J. Carbonell, and C. Jin. Topic-conditioned novelty detection. In *Proceedings of ACM SIGKDD'02*, pages 688–693, Edmonton, Alberta, Canada, July 2002. ACM Press.

Seamless Interface Transition Mechanism for Pedestrian Navigation Systems

ABSTRACT
This paper presents a posture recognition system for mobile computing devices to switch between different visualization modes seamlessly. It introduces a method to estimate orientation of a mobile user's hand. The recognition algorithm is based on state transitions triggered by time-line analysis of pitch angle and angular velocity of an orientation sensor. Currently the system can differentiate between three postures successfully. We associated each posture with different contexts which are of interest for pedestrian navigation systems: investigation, navigation and idle. We implemented a prototype system with an orientation tracker and GPS receiver connected to a PDA running an OpenGL|ES application. The system allows users to navigate and investigate in a campus environment.

Categories and Subject Descriptors
H.5.2 **[Information Interfaces and Presentation]:** User Interfaces—Interaction Styles; I.3.6 **[Computer Graphics]:** Methodology and Techniques—Interaction Techniques

General Terms
Algorithms, Measurement, Design.

Keywords
Posture recognition, interaction techniques, inertia sensors, context-aware retrieval, mobile computing.

1. INTRODUCTION
Location based services are becoming more available for different use cases, most notably navigation and information services. Vehicle based navigation systems became commercially successful systems as they can provide accurate data consistently. Most car navigation systems combine GPS and car telemetry data with Geographical Information Systems (GIS) data. One important advantage of vehicle systems is the available infrastructure to access the telemetry and other electronic systems (radio, lights, phone) and estimate the user context and adjust the system accordingly by adjusting the visual and audio characteristics, such as loudness of the radio, of the system if necessary.

Pedestrian navigation systems also require precise estimation of a user's context, which consists of environmental entities affecting the interaction between a user and an application. Context-aware systems use context to provide relevant services to the user, where relevancy depends on the user's task such as walking around, gazing in a particular direction [17].

With the advent of more capable mobile terminals such as mobile phones and PDAs it has become possible to perform visually appealing 2D and 3D renderings in real-time [2]. These can be maps, task specific applications (rescue, cultural heritage), and interactive guides; and have variable visual complexity. The size constraints of mobile devices make them cumbersome with minimal space for user interfaces. Moreover users expect to retrieve information using mobile computing devices with less effort. Therefore creating natural interaction techniques for mobile devices is an important challenge. One such interaction technique is to utilize users hand posture or gestures to trigger tasks, switch between applications or play games. With the advent of camera phones, first examples of such interfaces begin to be developed [16]. Even the computer vision algorithms are adequate for simple gesture recognition tasks on limited performance terminals; those techniques have not become mainstream, because of their high power consumption.

In this paper we are investigating context estimation methods for PDAs. In case of tour-guide applications we can simplify the context of a mobile user into three: investigation, navigation and idle. In the investigation state user is analyzing the façade of a building or an artifact in detail, where the application should present detailed visual information, if possible, using an Augmented Reality framework [11, 12, 18]. However at the navigation state the application should present 2D or 3D maps of the area with task specific data: street layout, names etc. The application is expected to detect also the idle state conditions when the user is not looking to the screen and trigger power save actions such as CPU clock adjustment. We propose to use a single orientation sensor to estimate the context of the user. This estimation is based on a novel posture recognition technique. The location will be computed using data from GPS sensor.

We first give a review on related work. In section three we describe our recognition method in detail. Afterwards we present a prototype system and case study to illustrate our ideas. The final chapter concludes the paper and presents our ideas on future research.

2. RELATED WORK
Location based services require estimation of mobile users location with adequate accuracy. Depending on application's needs this accuracy can be between kilometers or centimeters. Moreover most of the services require orientation data, at least the heading. GPS is the most prominent technology around for delivering up to centimeter accuracy (Differential-GPS). However, the fact that it only works outdoors is a major drawback. An example GPS based guide application is developed by Burgit [5]. Several alternative positioning technologies such as GSM-Triangulation, Wireless LAN Triangulation can estimate one's position indoors, using access points with well known positions [6, 7]. A recent work of Peternier et al [4] presents an

integrated mobile mixed reality system using Wireless LAN technologies. Another possibility is to track walking data for positioning purposes. There are several research groups working on walking behavior and pedestrian tracking [1, 3].

Until now all location based services we have investigated rely heavily on precise position and orientation of the user in global coordinate frame. However to create natural interfaces for handheld devices we need to recognize upper body postures, most importantly the position and orientation of the terminal relative to user's head.

Recognition of upper body postures and gestures is studied by several research groups: Amft et al reporting a Hidden Markov Models based gesture recognition algorithm up to 94% success rate for eating and drinking arm gestures [15].

Bao and Intille developed a system to detect everyday physical activities using five body-worn wire-free biaxial accelerometers with high accuracy (84%) [9]. They report that even with two sensors – thigh and wrist their recognition performance does not deteriorate. Similarly Wee and Mase report up to 91% success rate with their wearable sensors for activity recognition.

Widgor and Balakrishnan implemented a novel text input technique for mobile phones using one 2 DOF tilt sensor (acceleration sensor) to recognize user hand actions [14]. They report that 20 to 50 Hz sampling rates should be required for robust tilt implementation.

3. POSTURE RECOGNITION

As presented in the previous section, all of the orientation sensor work is based on either to assist precise tracking and positioning of the user in space or gesture recognition using several sensors.

Our goal is to create a stable differentiation mechanism between several hand postures and map them to several application dependent contexts.

The developed recognition algorithm is based on state transitions triggered by time-line analysis of orientation and angular velocity of the sensor. The orientation and angular velocity are the natural arm posture data of a pedestrian, which can be used as context widgets for a context-aware application.

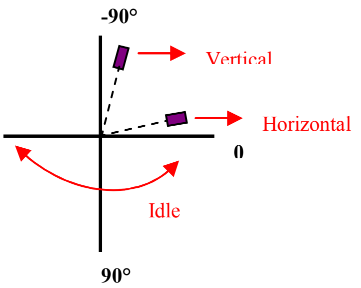

Figure 1. Drawing of target three hand postures from side view

The angle between user's forearm and upper arm is obtained from the orientation sensor as pitch angle, α, and analyzed to recognize different postures. We have gathered sample data from mobile users with various walking speeds, while moving their hands between three postures:

- *vertical*, where pitch angle is around 0°,
- *horizontal*, where pitch angle is around 90°,
- *idle*, where the hand may move freely (Figure 1).

Figure 2 shows pitch angle measurements of the user's arm movement in three different conditions: standing, walking, and running. Transitions between diverse arm postures can be inferred from the top left plot of Figure 2: For $0s \leq t < 5s$, the posture is on idle state. After this interval the user moves her hand up and stabilizes on horizontal posture until $t \approx 10s$. For $10s < t < 20s$, the user moves her hand down, stabilizes on idle state and moves her hand up. For $20s \leq t < 27s$, vertical posture is observed, and so on.

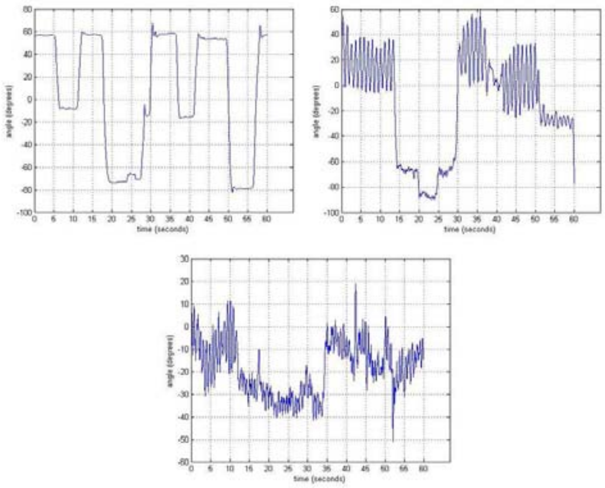

Figure 2. Pitch Angle Measurements while user is standing (top left), walking (top, right) and running (bottom).

The measurements indicate that with the increase of velocity the noise on the measured signal increases significantly. The noise can be observed on the top right plot of Figure 2, where the transition from idle posture to horizontal posture is not clearly recognizable at $t \approx 40$. Our current algorithm performs acceptably with users walking with low speed but the accuracy decreases significantly with increased speed due to the high frequency noise introduced into data by walking and running motion.

Table 1. Description of state transitions

1	2	3	4	5	6	7	8	9
$\alpha > 0°$ $\omega < -0.75$ $\Delta\alpha < 0$	$\alpha > 0°$ $\omega > 0.75$ $\Delta\alpha > 0$	$\alpha \approx 0°$ $\omega \approx 0$ $\Delta\alpha \approx 0$	$\alpha < 0°$ $\omega < -0.75$ $\Delta\alpha < 0$	$\alpha < 0°$ $\omega > 0.75$ $\Delta\alpha > 0$	$\alpha \approx 90°$ $\omega \approx 0$ $\Delta\alpha \approx 0$	$\alpha > 0°$ $\omega > 0.75$ $\Delta\alpha > 0$	$\alpha < 0°$ $\omega < -0.75$ $\Delta\alpha < 0$	$\alpha > 0°$ $\lvert\omega\rvert > 0.75$ $\Delta\alpha \neq 0$

$$\omega = angular\ velocity\ (rad/s)$$
$$\Delta\alpha = \alpha_{true} - \alpha_{estimated}$$

We implemented a sliding window to detect changes of the hand on pitch angle, α. A window, which contains five angle values obtained in time interval [t-1, t-5], is created at each time step and upcoming angle is estimated by multiplying them with increasing weights.

$$0.1 * \alpha_i + 0.1 * \alpha_{i+1} + 0.1 * \alpha_{i+2} + 0.2 * \alpha_{i+3} + 0.5 * \alpha_{i+4} = \alpha_{estimated}$$

The $\alpha_{estimated}$ angle is compared with the measured angle α_{i+5} to identify if the hand is moving up or down.

$$\alpha_{i+5} > \alpha_{estimated} \Rightarrow downside\ change$$

$$\alpha_{i+5} < \alpha_{estimated} \Rightarrow upside\ change$$

$$\alpha_{i+5} \cong \alpha_{estimated} \Rightarrow no\ change \begin{cases} \alpha_{i+5} \to -90^\circ, vertical \\ \alpha_{i+5} \to 0^\circ, horizontal \end{cases}$$

However using the pitch angle in one single direction is not sufficient enough to have robust posture recognition. We have also evaluated the case, where the user performs short tilts (rotations around the longitudinal axis) causing an inference on the state transition. For such cases, a filter is implemented on the system which increased the state transition accuracy.

Figure 3 shows plots of sample pitch and tilt angle measurements of the same motion and corresponding state estimations. For 7s< t <9s, tilt angle is increasing and decreasing instantly (top right plot, Figure 3) which affects the pitch angle. In spite of the fact that the user holds her hand stable around -20° during angular data measurement, the top left plot of Figure 3 shows that the pitch angle is changing up to 20°. Same erroneous measurement can be observed for 11s < t < 15s. These unexpected changes cause inaccurate state estimation (bottom left plot, Figure 3). Therefore, estimation accuracy is increased (bottom right plot, Figure 3) by introducing the system with a tilt angle filter, which locks the state to the previous one if major changes occur on tilt angles.

The system becomes unstable and produces erroneous results when users perform other occasional movement patterns. Therefore we have introduced an additional data, angular velocity, to the recognition system. The change of angular velocity together with the angle allows us more stable recognition results.

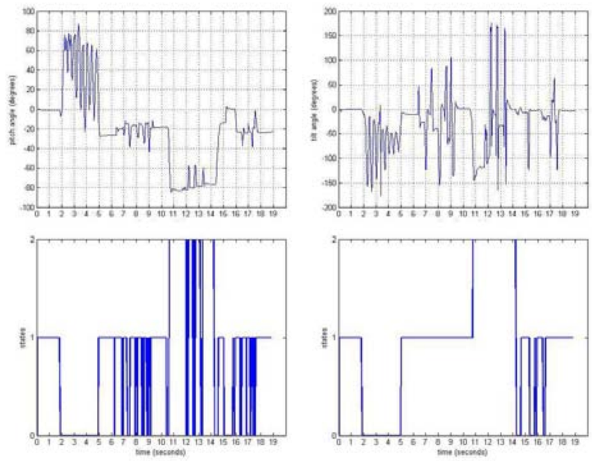

Figure 3. Sample pitch angle measurement (top left), Tilt measurement of the same posture (top right) causing erroneous estimation – 0: idle state, 1: navigation state, 2: investigation state (bottom left) and increased estimation accuracy with tilt filter (bottom, right).

Finally we developed a finite state machine to map all possible postures into one of the three states: investigation, navigation and idle (Figure 4). The investigation state is when a user holds a mobile terminal in vertical position to use it in an Augmented Reality context. In this condition the user needs to investigate point of interest buildings and receives environmental information according to her gaze direction in the local coordinate frame. The navigation state is when a user holds a mobile terminal in horizontal position to use it to render maps or GIS information. Thus, the user receives environmental information in the global coordinate frame. There is a third idle state, where the user is not in either posture and moves her hand freely. In this state, rendering is minimized to allow power save property.

The conditions satisfying the state transitions in Figure 4 are defined in Table 1. In this algorithm, firstly, the estimated pitch angle value is compared with the angular value perceived from the orientation sensor at that time step. If they are approximately equal, user's arm posture is estimated to be stable and either in investigation or navigation state (3rd and 6th columns of Table

1). Other enumerated transitions include conditions which define possible changes between states, i.e. while arm posture is on idle state and the user moves her hand upwards, then it is possible to switch state to navigation or upside change on arm posture continues and state is switched to investigation. The state estimation algorithm is empowered by introducing angular velocity and tilt angle filter to the system.

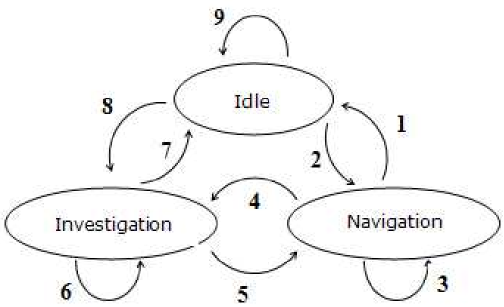

Figure 4. State transitions between three contexts.

Unexpected arm movements of the user can affect the accuracy of the system. While the user holds the PDA in horizontal or vertical position (navigation or investigation state) and suddenly performs fast upward or downward movements with her hand, i.e. waving to somebody, the system is stabilized in the former state with a tolerably accuracy rate.

4. SYSTEM SETUP & CASE STUDY

We have developed an integrated hardware and software system to demonstrate our ideas. We have connected two inertia sensors over RS232 port to a PocketPC PDA separately. This enabled us to verify our algorithm with more than one sensor technologies. For outdoor case study we also connected a GPS sensor over Bluetooth wireless link (Figure 5).

Figure 5. Prototype recognition platform. From left to right: Bluetooth enabled GPS, PDA, RS232 connection with battery, Orientation Sensor

The recognition and estimation software is developed in C++. For the case study we have developed an OpenGL|ES based application, also reading the GPS data and orientation data from communication ports. Our OpenGL|ES implementation is based on Vincent Platform [8] and GLUT|ES Windowing libraries [13] (Figure 6).

We have implemented a prototype navigation system, which guides visitors in downtown Istanbul between Istanbul Biennale venues. However we have experienced serious problems with using both of the orientation sensors we experimented (MTx orientation tracker from Xsens and InterSense's InertiaCube^2) as mobile devices. Both require either large battery packs to operate without a power supply over a long time or a 6V DC power adapter connection. Therefore we have limited the posture recognition experiments in indoors use cases in form of a campus guide (Figure 7). Note that the investigation mode is not a live video image annotated with data, but a snapshot displayed when the user directs her PDA to the computer science building. For MTX the pitch angle and velocity is acquired from the sensor, for InterSense's InertiaCube^2 we have to compute the angular velocity. For both sensors we experienced stable posture recognition and state switch.

We performed a user study to examine the accuracy of our system. In this test, all possible state transitions emphasized in Figure 4 are performed. The overall accuracy rate is calculated as approximately %87. Performing sudden up-down movements in navigation state and investigation state produced some erroneous results.

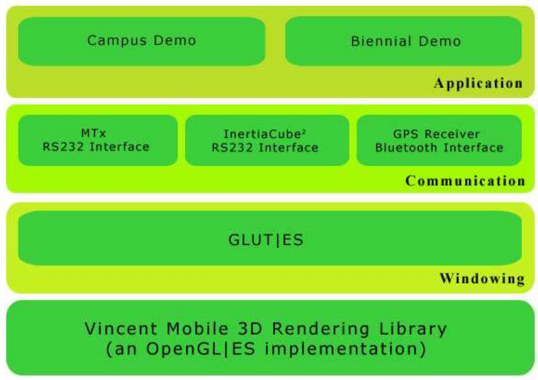

Figure 6. Software Architecture

Figure 7. Campus Demo Concept: Two navigation views (left and middle) and one investigation view (right)

5. CONCLUSION AND FUTURE WORK

We have implemented a recognition algorithm with one orientation sensor attached to a PDA to distinguish between two different postures of the hand and an idle state. This data can be used to differentiate between three states to switch between different applications seamlessly. We tested our approach on two different orientation sensors.

In the global coordinate frame, we used GPS sensor data to locate the user, acquire her gaze direction, embed GIS data and provide information about point of interest buildings. In the local coordinate frame, we used orientation sensor data to allow the user interact with the mobile device while performing natural arm postures and perceive information on different user interfaces. By combining these interaction techniques of global and local coordinate frame, we provide a context-aware interaction framework for pedestrian navigation systems on mobile devices by seamlessly changing graphical user interfaces.

Our work shows that once orientation sensors became part of mobile terminals, they can be used to create natural interaction techniques with mobile terminals.

In future we plan to improve our technique to recognize postures while the user is walking and running. Therefore, a robust battery pack connection must be supplied to the system to test the prototype outdoors completely. Also, we want to improve Augmented Reality context of our system by implementing an investigation state with real-time camera input of the environment, where the buildings are annotated with virtual text labels as shown in Figure 7.

6. REFERENCES

[1] Lee, S., and Mase, K. “Recognition of Walking Behaviors for Pedestrian Navigation”, Proc. of IEEE International Conference on Control Applications, September 2001, Mexico City, Mexico

[2] Shimada, S., Tanizaki, M., and Maruyama, K. “Ubiquitous Spatial-Information Services Using Cell Phones”, IEEE Micro, 2002

[3] Foxlin, E. “Pedestrian Tracking with Shoe-Mounted Intertial Sensors”, IEEE Computer Graphics and Applications, Nov/Dec 2005.

[4] Peternier, A., Vexo, F. and Thalmann, D. “Wearable Mixed Reality System In Less Than 1 Pound”, Eurographics Symposium on Virtual Environments, 2006

[5] Burigat S., and Chittaro L.” Location-aware visualization of VRML models in GPS-based mobile guides. Web3D ’05: Proceedings of the tenth international conference on 3D Web technology (New York, NY, USA, 2005), ACM Press, pp. 57–64.

[6] Cheng Y., Chawathe Y., Lamarca A., and Krumm J. ”Accuracy characterization for metropolitanscale wi-fi localization”, Third International Conference on Mobile Systems, Applications and Services (2005).

[7] Howard A., Siddiqi S., and Sukhatme G. S., “An experimental study of localization using wireless ethernet. Proceedings of the 4th International Conference on Field and Service Robotics, Lake Yamanaka, Japan (2003).

[8] Vincent Mobile 3D Rendering Library, http://sourceforge.net/projects/ogl-es (as of April 2006)

[9] Bao, L., and Intille, S.S. “Activity Recognition from User-Annotated Acceleration Data”, Pervasive 2004, LNCS 3001, pp1-17

[10] Lee, S., and Mase, K. “Activity and Location Recognition Using Wearable Sensors”, IEEE Pervasive Computing, 1(3):24–32, 2002.

[11] Schmalstieg, D., and Reitmayr, G. “Augmented Reality as a Medium for Cartography” Multimedia Cartography, 2nd ed. (ed. W. Cartwright, M. Peterson, G. Gartner), Springer-Verlag, 2006

[12] Feiner, S., MacIntyre, B., Höllerer, T., and Webster, A. ”A touring machine: Prototyping 3D mobile augmented reality systems for exploring the urban environment”. In: Proc. ISWC’97. Cambridge, MA, USA, pp 74–81

[13] GLUT|ES - The OpenGL|ES Utility Toolkit, http://glutes.sourceforge.net/, (as of April 2006)

[14] Widgor, D., and Balakrishnan, R. “TiltText: Using Tilt for Text Input to Mobile Phones”, UIST 2003, CHI Letters, Volume 5, Issue 2

[15] Amft, O., Junker, H., and Troster, G. “Detection of eating and drinking arm gestures using intertial body-worn sensors”, IEEE Symposium on Wearable Computers 2005.

[16] Rohs, M. “Real-world interaction with camera-phones.” In International Symposium on Ubiquitous Computing Systems.2004

[17] Abowd, G. D., Dey, A. K., Brown, P.J., Davies, N., Smith, M., and Steggles, P. “Towards a Better Understanding of Context and Context-Awareness”. In Proc. Handheld and Ubiquitous Computing: First International Symposium, HUC'99, Karlsruhe, Germany, September 1999.

[18] Wagner, D. and Schmalstieg, D. “First Steps Towards Handheld Augmented Reality“. In Proc. Seventh IEEE International Symposium on Wearable Computers, 2003.

Designing game logics for dynamic Active Surfaces

Gronvall Erik
Communication Science Dpt.
University of Siena
Via dei Termini 6,
53100 Siena IT
+39 0577 270565
gronvall@media.unisi.it

Pollini Alessandro
Communication Science Dpt.
University of Siena
Via dei Termini 6,
53100 Siena IT
+39 0577 270565
pollini@media.unisi.it

Rullo Alessia
Communication Science Dpt.
University of Siena
Via dei Termini 6,
53100 Siena IT
+39 0577 270565
rullo@media.unisi.it

Svensson David
Computer Science Dpt.
Lund University
Ole Römers väg 3
SE-223 63 Lund
+46 46 2224249
david@cs.lth.se

ABSTRACT

This paper describes the development of a modular system of interactive tiles to support therapists' in performing therapeutic activities together with impaired children in a swimming pool.

Active Surfaces support mobile interactions and dynamic configuration of assemblies of tiles. Each tile represents an interactive unit, able to communicate with other tiles and to exchange data. The tiles can be assembled on a physical and logical service level to support activities of different complexity. This creates technical challenges where assemblies are re-created over time between devices with limited input and output capabilities. Furthermore, Active Surfaces challenge the concept of understandability and how users can make sense of assembled systems with no or limited output capabilities.

Categories and Subject Descriptors

C.3 **[Special-purpose and application-based systems]** *Microprocessor/microcomputer applications, Process control systems, Real-time and embedded systems, Signal processing systems.*

General Terms

Performance, Design, Reliability, Experimentation, Human Factors.

Keywords

Ubiquitous computing, Palpable computing, end-user composition, distributed game logic, dynamic assembly.

1. INTRODUCTION

Mobile devices and small computer based systems can have a strong need for interaction, communication and information retrieval. These needs and how they are handled are not always clear in a standard ubiquitous setting. Traditionally, the notion of ambient computing has been consolidated focusing on the design of distributed, pervasive and reactive systems able to communicate with us and to continuously adapt to our current needs and expectations [1]. The exploration of smart objects and distributed systems highlight, from an interaction design perspective, subjects like devices, services, assemblies and distributed logic.

Ubiquitous and portable devices will be used in a wide range of situations, involving different physical and social environments and tasks. They will need to allow usage in many different ways and combination of modes. They should be configurable in the most appropriate way given the objectives, the users' competencies and the task at hand. Furthermore, unlike traditional ICT, users of mobile and pervasive devices will, typically, experience dynamic properties in use, for example higher levels of degradation and task switching.

Issues like inspection, of both breakdown situations and system configuration becomes important key-points in modular, mobile devices with limited interaction possibilities. These themes become crucial if applied to critical domains such as health care and rehabilitation. Active Surfaces tries to investigate these properties using an evolutionary approach on ambient computing with the notion of Palpability. Palpable computing aims at supporting user control by composing and de-composing assemblies of devices and services, while supporting meaningful interaction with users. These assemblies are configurable by the user depending on the context, the user's needs and expectations. Consequently, these systems should support continuous attribution and negotiation of meaning through interaction.

These challenges and others are addressed in the EU-funded Integrated Project PalCom: Palpable Computing – A new perspective on Ambient Computing (http://www.ist-palcom.org). This project uses the term 'palpable computing' to denote a new kind of ambient computing which is concerned with the above mentioned challenges in complex and dynamic ambient computing environments.

In this paper we will present a design concept supporting rehabilitation practice performed in a swimming pool embodying some of the qualities that creates the foundation of palpable computing. The practice built around rehabilitation activities performed in a swimming pool for disabled people serves as the main inspiration for this application.

At the public swimming pool in Siena (Italy), a volunteer association provides training and group activities for disabled people. The swimming pool by itself represents a powerful and privileged setting, because water supports the body and takes the weight off the joints making for example movements less painful. Moreover, water is a great 'equalizer' for disabled people: they find their movements easier and less different from those of non-disabled people while inside the water.

Today, different aspects of the rehabilitation practice are not well integrated with each other. Specific tasks and tools are designed for the motor physiotherapy; whereas other tasks, aids and tools are defined to support the acquisition of cognitive skills. The two activities are usually never integrated. You train one thing at the time, either A or B, rarely the two of them together.

Our aim is to join cognitive and physical rehabilitation objectives by providing the children with meaningful activity games supported by a distributed system of smart surfaces. The application system relies on the specification and reference implementation of a virtual machine for palpable devices, the PalVm, designed to match the evolving needs within the PalCom project [2]. The PalVm is a language-neutral virtual machine designed to support object oriented languages and small footprints. PalVm programs are deployed in binary components which are instantiated as run-time components, objects that encapsulate a set of classes and their required and provided interfaces. Applications running on top of the PalVm can take advantage of the palpable qualities offered by this platform. Active Surfaces is one example of such an application.

2. ACTIVE SURFACES

Active surfaces is thought of as mobile smart devices that serve for mobile gaming and rehabilitation activities. In this section the concept of Active surfaces will be explained.

2.1 Initial suggestions from fieldwork

An intense period of observations in the swimming pool permitted us to capture some peculiar features of the rehabilitation practice. We consider these suggestions as relevant for triggering our design process.

On-the-fly configuration. Dealing with continuously changing conditions and demands, the therapists need to find out creative solutions. They ask for adaptable tools to fit the needs of their patients even along the single treatment session. A core characteristic is that the tools have to be easily reconfigurable on the fly so they can adapt to these emerging situations.

Re-using games and keeping track of best practices. The therapists can rely on their experience and creativity in designing solutions. They would like to have the possibility to collect best practices and to re-use successful configurations of objects (devices) and games. Such a resource would contain also the know-how concerning alternative usage of the existing devices and services configurations.

Dealing with failures and degradation. The therapists usually deal with dynamic settings and changing conditions. This implies the ability to rearrange the available resources even in case of degraded performance of system. The tools they use have to guarantee robustness to the water setting (e.g. waterproof material and tight cases) and resilience in working functionalities. They would need also to recover from components failures by using substitutes devices to not interrupt the session.

Combining cognitive and physical rehabilitation. The lack of integration of physical and cognitive rehabilitation represents a constraint for current rehabilitation practice. The cognitive tasks are usually too static and children may lose attention. On the other hand, motor rehabilitation is very demanding at a physical level and is based on repetitive sequences of actions: patients often perceive them as tiring and not engaging.

2.2 Concept and components

The tiles act as building blocks that can be combined with a library of content (e.g. images, sounds and pictures). Furthermore they have reactive behaviours in relation to different input actions and orientation. Each tile provides outputs as visual or tactile feedback to support the accomplishment of the tasks given and to guide the patients in the interaction as described in section 4 below.

Today three kinds of Active Surfaces components exist: there is an Assembler Tile and a number of 'normal' tiles, and there is also a user interface, the Migrating User Interface (MUI)1 [3]. The PalVM is the platform on top of which these components run."

The ‘normal’ tiles are the ones used in the different activities and games together with the different users. Then there exists a privileged tile, the Assembler Tile. It is used by the therapists to program the other tiles. The Assembler Tile is equipped with a touch-sensor, realized by an infrared transmitter and receiver. This touch-sensor allows two interaction modes, a ‘one-touch’ or a ‘double-touch’ that evoke different functionalities in the Assembler Tile. These functions and their use will be described further down in this document. The MUI is used in a pre-activity phase to create general game logics and rules that later can be used in the pool. A therapist downloads the general rules created in the MUI into the Assembler Tile and brings it to the pool. Now the therapists can assemble different ‘normal’ tiles using the Assembler tile to support a wide range of activities. The tiles themselves once assembled constitute a network of physical (and software) objects that communicate and exchange data and are able to recognize their relative positions.

These features allow constructing meaningful configurations of different tiles. Each configuration is intended as an assembly of components. The therapists can configure these assemblies of components to define rehabilitation tasks. They can save successful configurations, keep memories of previous configurations and generate new assemblies to support patients’ specific needs. The rehabilitation activities enabled by the active surfaces allow a smooth integration of cognitive and physical tasks.

The Active Surfaces concept accounts for the need of configurability, constructability, modularity, physicality and creativity in rehabilitation practice. ‘One’ Active Surface consists of a tile, measuring 30*30 cm. Each Active Surface is thought of as a modular unit that can communicate with the others by its six sides. The tiles are able to recognize their relative positions in respect to other tiles. A number of tile

1 The MUI End-User Composition Tool is used for programming the behaviour of the tiles. MUI, developed within the PalCom project (www.palcom.dk) at the University of Lund, enables a therapist to browse existing tiles and their configurations. She can discover the tiles currently in use, and design a new exercise in the MUI browser (e.g. running on a PDA or a laptop).

components can be assembled to constitute a network of physical (and software) objects that communicate and exchange data. Many qualities of palpable devices are embodied in the Active Surfaces concept. Today a prototype is being used for evaluation purposes based upon a Basic Stamp 2 micro controller and IR communication. These tiles (as seen in figure 1) offer limited functionality, but sufficient for initial trials and proof-of-concept. The next generation of tiles embedding the full vision and the palpable framework is currently under development together with the University of Aarhus (Denmark) and Lund University (Sweden).

Figure 1. Initial working prototype under construction and final result

The main idea is to rethink the environment of the pool, making it a place for rehabilitation and play activities. Today the swimming pool is designed for swimming: the water serves as the mean of interaction. People usually don't have any (strong) relation with the pool by itself: the edges and the bottom are not conceived for any purpose of interaction. The design process aims at re-considering the surfaces of the pool and to change the activities that usually take place there [4],[5].

As interaction designers our focus is on enabling environments and tools supporting engaging rehabilitation activities. The Active Surfaces is the concept that embodies these issues. The surface of the pool becomes active re-designing the bottom, the edges and even the water surface. In this vision the floating tiles constitute one of the main supports for the interaction and the therapeutic activity.

3. PALPABLE QUALITIES

Active surfaces represents an exemplar application of the palpable computing framework being developed in the PalCom project. Limited resources and basic input- output profiles characterize these devices.

Palpability emerges as a property-in-use of the tiles' assembly. Dynamics of physical- logical construction/deconstruction, mobile interaction and services communication sum up the palpable qualities of the application.

Being conceived as an assembly, active surfaces could provide a valuable example of physical construction/deconstruction of components. Thus the physical construction of assemblies [2] provides end-users with control of the system behaviour and adaptation to the context. As Active Surfaces don't rely on data and information visualization; the tiles assume physical construction and physical interface as major strengths for the users.

The Active Surfaces constitute assemblies on different levels: on the logical level the therapist can define what the rules are and what the purpose is. On the functional level the user can mark out the relations and the sequences and on the physical levels the user can define which patterns and connections can take place, to reach the final solution.

By focusing on the virtues of therapist (end-user) composition, the Active Surfaces is also complemented by the MUI browser mechanism for programming the rules and the behaviours to be instantiated in the tiles. The therapists then create patterns by physically building tiles' sequences [6]. The tiles address also scalability, offering the opportunity to produce scalable solutions still relying on low level resources management. Palpable systems can be described, as many ambient computing system, as a heterogeneous mix of distributed, embedded devices with different capabilities. The Active Surfaces must provide scalability and persistence across different devices so that errors in one part the system do not spread to other parts of the system. These features are supported by an appropriate degree of decoupling between different parts of the implementation.

By adopting labels and grouping of the devices, Active Surfaces have chance to survive through system degradation: when interruptions occur the users can substitute components while continuing the activities.

The system can still guarantee understandability also concerning the balance between system automation and therapist' (i.e. user) control. The tiles have to preserve the understandability and support the users to maintain control over the technology.

Regarding usability of resources and resource access, the Active Surfaces create flexible ad-hoc networks connecting the single devices. Networked tiles preserve their own identity and dynamically seek for available tiles in the vicinity. The tiles continuously inspect what communication processes are taking place at the moment looking for specific connection on all its sides.

The proof of the Active surfaces concept may also enable the designers to explore the relation between change/stability in use. In fact the assembly' behaviours are instantiated in physical configurations that can be saved, reused (also in part) and instantiated in different physical patterns. The therapist can show the right pattern (sequence) to the system and record (save) the configuration by using an assembler tile. The dynamics between configurations' change and stability may address the future practice of rehabilitation and the way in which the Active Surfaces could support it.

4. SCENARIO

Along the different phases of the work analysis we used scenarios to evaluate, together with our stakeholders, how the defined concepts could suit their needs and to envision possible usage of the final tools. Scenarios themselves were used as design objects and they evolved along the design process being created, refined and also sometimes dismissed [7].

As far as we have developed the early prototype, we also designed games of different kinds and of different degrees of complexity, which can easily show the potentials of the application. The basic activity Active Surfaces can be used in is a position-based game. The position games regard all the sequence-composition activities where users have to place letters, pictures, textures or colours in the right order following the activity task. For instance the position game we give as an example in the scenario below is based on the composition of the word "C I A O". This is a linear, sequential word construction. In this kind of games the tiles have a unique ID and they know exactly what is their identity and position in the sequence.

The more complex kind of games can be described assuming the Scrabble task as main example. Providing multiple simultaneous combinations, the scrabble is such a meaningful setting in which the same letters are present in a number of occurrences. For instance, considering many words, it is possible that the letter E occurs several times and each E should know what is its correct position and orientation in the crossword assembly. Thus all the letters E need a label that group them, through which they can better identify their role in the game and have to receive the right behaviour for the accomplishment of the task. The Scrabble game logic will be described in par. 4.2.

The games we have been designing with Active Surfaces follow a logic based on condition satisfaction rules. Tiles' states are described through the use of a "happiness" state. These terms are used with specific meanings in the scenario and in the code development. We consider different states of happiness (conditions' satisfaction) for the position and orientation of the tiles in the assembly.

- *SideHappiness* means that a tile realizes that it is correctly connected on a particular side. On the side(s) that are Happy the tile provide the users with HappySide feedback. If all its sides are correctly aligned, *LocalHappiness* is instead achieved.
- *LocalHappiness* means that the tile is properly connected to the others and it has on each side the tiles it was looking for. It is in the right position and it is correctly orientated in the assembly.
- *UnHappy* – A tile is unhappy when it lacks *LocalHappiness* (i.e. while no happiness state or only *SideHappy* is reached, a tile is still *UnHappy*). While a tile is *UnHappy* it broadcasts its *UnHappy* state to the other tiles.
- *AllHappiness* means that all the tiles satisfy the *LocalHappiness* state. A tile that is not *UnHappy*, does not communicate anything to the other tiles. This gives, the lack of *UnHappy* messages (given a certain timeout) within the system together with *LocalHappiness* realizes the *AllHappiness* state. The game is solved.

4.1 Active Surfaces in use

The therapist or trainer configures the activity outside the pool; at her home or in a remote office. She can also configure the activity in the vicinity of the pool, but there is no specific need for that from a system perspective.

The following PalVm services enable this early phase of configuration:

- *PalCom device discovery* The device discovery enables a device running the PreWM to autonomously discover new (palpable ready) devices in the vicinity.
- *Service discovery* As the Device discovery was used to automate the discovery of devices, the Service discovery is used to query and identify running services on remote devices.

Furthermore is each tile aware of its own unique id, a dynamic label and can propagate this data through its 6 sides. The tiles can also listen on incoming traffic on their 6 sides once the PalVm has communicated via the device discovery and service discovery that there is another tile present on a particular side.

4.1.1 Configuring the activity

The therapist attaches the assembler tile (AT) and the MUI Browser with each other. The PalCom device discovery now allows the MUI and the assembler tile to dynamically connect and exchange data. The PalVm then uses the service discovery to locate the correct or required services. Once these services have been identified, behaviours and game rules can pass from the MUI to the AT. The therapist configures the activity by setting those parameters in the tiles that constitute a game or therapeutic activity, such as the kind of activity (e.g. Position game or Scrabbles game) and the output mode (e.g. Blinking light).

In a first version of Active Surfaces the AT will have a cable connection (i.e. RJ-45 Ethernet) between the system hosting the MUI and the AT. This connection will later on be wireless.

4.1.2 Configuring the game tiles

The therapist can now bring the AT to the pool and align the tiles she would like to include in the game, showing physically the tiles the 'winning' position and pattern. This activity will initiate device and service discovery activities inside the tiles. A tile now knows which tiles are connected to its sides, and their orientation as illustrated in Figure 2.

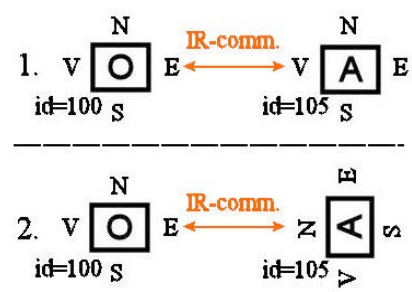

Figure 2. Row 1. Tile id=100 knows that on side 'E' it has 'V' side of the tile id=105 and vice versa.

Row 2. Tile id=100 knows that on side 'E' it has 'N' side of the tile id=105 and vice versa.

The AT is now being connected to the sequence of tiles. The therapist can rely on a simple physical interface on the AT: by using 'one touch' input she settles the tiles in the sequence. This command sends a broadcast message to the connected tiles to set their neighbour's ID and positions. This means that each tile looks which other tiles are currently connected to its sides and saves this information together with the orientation of these neighbour tiles.

The broadcast message can appear as follows:

AT to all Tiles: Set Neighbours(){ Set Identity & Orientation }

Now all the tiles have memorized their own and their neighbour's ID, position and orientation as shown in figure 2.

The feedback the tiles have to produce, when reaching the different Happy-states are now being downloaded from the AT to all the tiles.

The tiles now notify the therapist when the configuration is successfully achieved by providing the settled output. Since they are now in their winning-position (i.e. they are all in a *LocalHappiness* which gives an overall *AllHappiness* state, this since no *UnHappy* messages are being broadcasted).

At this point the AT asks for the current assembly of tiles, to store the current assembly for later use as a trace of best practice to serve as an inspiration for other therapists or patient activities.

The therapist is now ready to start the activity with the patient and throws the tiles into the pool. The child now starts to play with the Active Surfaces. The initial positions can be viewed in figure 3.

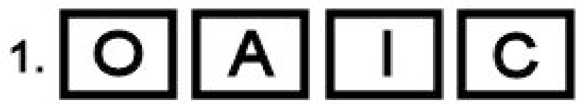

Figure 3. Initial configuration, the tiles are thrown into the pool. There is NoHappiness.

A *Discovery report (on a PalVm level)* takes place as soon as the game starts. Each tile starts to report which tile is close to its sides constantly matching these data with the stored *Neighbours* data.

In the early tentative trails the child tries different wrong alternatives by moving around the tiles. This provokes new *Discovery* actions (on a PalVm level). When a tile is recognized on a side, a getNeighbour action is performed (on a service level). This gives the tile information of the Identity and Orientation of the tiles surrounding it. The message the different tiles broadcast is: *AllUnHappy*. This since they are all in a non *LocallyHappy* state.

Step by step the child finds out local solutions for the tile game but still not the global solution that solves the complete sequence.

The child puts two tiles aligned following the right configuration, but still not with the complete solution presented. This gives a local feedback that the two tiles are correctly placed while the final feedback is still not given as shown in figure 4.

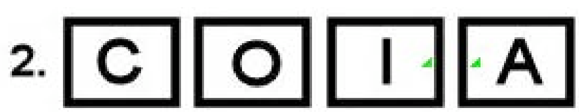

Figure 4: HappySide for 'I' and 'A' between the 'I' and 'A' sides (Green 'triangle' feedback).

Get Neighbors() (C, O, I, A) gives *HappySide = true* for *I.East* and *A.West*. This gives that I and A enters a *HappySide* mode on these sides.

The connected sides' lights on I and A are now being turned on and (green light) feedback is given to demonstrate that the *HappySides* condition on these sides have been reached. For both the tiles the *LocalHappiness* state is not yet true because it doesn't realize all the required conditions for their happiness state.

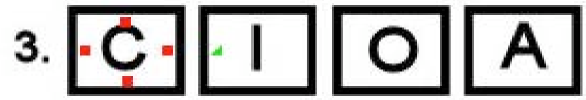

Figure 5: 'C' is LocallyHappy (Red 'square' feedback) and 'I' has one HappySide (Green 'triangle' feedback).

Figure 5 shows another try to reach the game solution. Given that the winning solution is the word sequence "CIAO", C and I are in their right position and orientation. C now reaches a complete *HappySide* state and thus is *LocallyHappy*. This *LocallyHappy* state provides the user with a (red) light output depending on the initial configuration.

Get Neighbors() (C, I, O, A) gives *HappySide = true* for *C.East* and *I.West*. These condition makes *C=LocallyHappy* and *I=SideHappy*.

The global state of the active surfaces is maintained by a broadcast of the message *UnHappy()* from those tiles that are not LocallyHappy. That puts all the tiles in a *AllUnhappy* state.

Going toward the final (and global) solution of the game, the child starts to align all the tiles in the right position by following the linear orientation of the word sequence.

Moving all the tiles correctly gives the final output and the game is solved.

getNeighbors(C, I, A, O) gives *HappySide = true* for *C.East-I.West* and *I.East – A.West* and *A.East – O.West*. This results in that all the tiles (C, I, A and O) reaches the *LocallyHappy* state.

When all tiles are in the LocallyHappy state (i.e. no *UnHappy* message is broadcasted) the tiles reach the *AllHappiness* state. The game is solved as presented in figure 6.

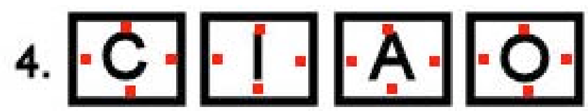

Figure 6: All tiles are LocallyHappy, this gives a complete happiness within the system (AllHappy).

4.2 Scrabble game logic

Scrabble[2] game is a popular word board game, in which two to four players score points by forming words from individual lettered tiles on a 15x15 game board [8]. The words are formed across and down in a crossword fashion, and must appear in a standard dictionary. Active Surfaces can be used in high complexity scrabble-like game in which each word crosses the others and has letters in common. In the example given we describe the composition of "H E L L O" and "L O O K" and how they can create the scrabble assemblies that appears in figure 7.

2 "Scrabble" is a registered trademark of Hasbro, Inc

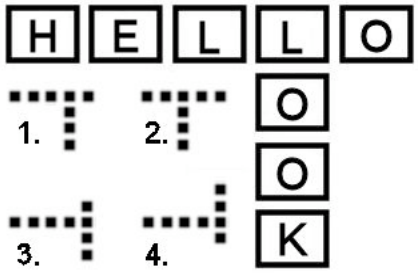

Figure 7. Solution 1 and 2 can be reached using only the tiles illustrated in the big picture, if an extra L is present, solution 3 and 4 can be reached. However, leaving a spare O tile.

Being present three L and three O, it is necessary to characterize the tiles' identity and to distinguish their position and orientation in the sequence.

Starting a new scrabble game activity, the therapist can create labels with a *SetEqual* command by using the Assembler Tile. This creates a unique *tileLabel* for the group of tiles currently attached to the assembler tile; in the example shown in figure 8, the L group and the O group have been created before the game. The therapist performs this action by the *Double touch* command on the AT interface setting the label equal in the group of tiles. Now the grouped tiles still have different IDs but share the same instructions for the connections (position and orientation of the neighbouring tiles). For instance the three L can be used in the different positions still maintaining adequate coherence in the game.

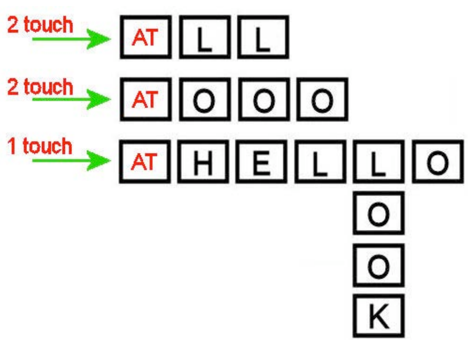

Figure 8. The Assembler Tile is used to group the tiles and create labels.

Grouping the tiles is the mean through which users can manage tiles' identities and eventually recover in case of tiles' breakdown by substituting them with new ones. When created in a group, the tiles under the same label might substitute each other and the therapist would be allowed to recover from system degradation at-hand.

In this way Active Surfaces shows a high combinatory power to support the therapist in game composition creation along her practice.

4.3 Re-assembling the game and storing

During the activity the therapist can re-attach the AT and change the game. She is now configuring a new assembly to be created along the activity with the patients. When this is done without the Browser, the game owns the same input/output profile as in the existing activity. It is nevertheless allowed to configure the tiles populating the game by differentiating her identity or positioning, just by placing them differently in relation to the others.

The assembler tile also provides the therapist with the opportunity to store and re-use previous games. When connected to the tiles along the setting up of the activity (e.g. *SetNeighbours and SetEqual*) the AT records and stores the configurations that are being used and the shapes created with the tiles. When the AT is re-connected to the MUI Browser, the stored data are uploaded to the MUI and stay available for the different therapists and trainers. The database has been created both as a memory of the past activity and as an inspirational repository of use (e.g. Best Practices collection).

5. DISCUSSION

The concept of Active Surfaces elaborates distributed game logics for the rehabilitation practice in the swimming pool.

The scenarios we developed are based on the idea of end-user composition, mobile interaction and user control and it seems very promising for our stakeholders at the hospital and at the swimming pool. In particular, users appreciate the idea of being supported by ready-at-hand and easy to program technology. The Active Surfaces provides them with the possibility to improve the day by day rehabilitation practice.

The concept elaborates on a new challenging view of construction complemented with deconstruction of physical assembly. The therapist is asked to manipulate and physically configure the tiles while the dynamic and self-configuring discovering of components occurs. This guarantees minimum skills in technology and programming for the users.

Active Surfaces provides the therapist with the possibility to adapt the technology pursuing extreme change and flexibility while keeping system stability. In that way we have situations where total control is desirable, but has to be complemented with sense making and meaning attribution of events.

Palpable computing can be seen as extending ambient computing with additional characteristics for user control. Palpable computing systems offer not only invisibility (the capacity of unobtrusively performing computing tasks in the background environment) but also visibility, that is, the capacity of making visible to users what they are doing and what they may do. Moreover, systems should offer both construction (the ability to support end-user composition of devices or services to form new devices and/or services) and also deconstruction, that is, the ability to disassemble a device or service into its constituent parts to enable understanding and manipulating of each part individually [2].

Palpable computing may constitute the framework for the definition of design guidelines for mobile interactive systems. Challenges in the shape of dichotomies as the ones described above, can serve as means through which the designer can interpret and re-define pervasive applications.

6. ACKNOWLEDGMENTS

The work regarding the game-logic and the communication between the tiles has been developed together with University of Aarhus (Denmark) and Lund University (Sweden). For their work in relation to this paper we would like to especially thank Prof. Boris Magnusson and Dr. Görel Hedin.

Our thanks also go to Laura Cardosi, Maria Grazia Burroni and the therapists at the Functional Rehabilitation Unit of the "Le Scotte" hospital, Siena, for their open-minded collaboration and continuous support during fieldwork and participatory design. The research was part of the EC IST PalCom project.

7. REFERENCES

[1] Weiser, M. The Computer for the Twenty-First Century. *Scientific American*, pp. 94-10, September 1991.

[2] Schultz, U.P., Corry, E., Lund, K.V. Virtual Machines for Ambient Computing: A Palpable Computing Perspective *OT4AmI ECOOP Workshop Position Paper* 2005.

[3] Svensson,D. Magnusson,B. and Hedin,G. Composing ad-hoc applications on ad-hoc networks using MUI *Proceedings of Net.ObjectDays, 6th Annual International Conference on Object-Oriented and Internet-based Technologies, Concepts, and Applications for a Networked World*, Erfurt, Germany, September 2005.

[4] Greenbaum, J. Kyng, M. *Design at Work*. Hillsdale, NJ: Erlbaum, 1991.

[5] Marti, P., Rizzo, A. Levels of design: from usability to experience. *HCI International 2003, 10th International Conference on Human - Computer Interaction*, Crete, Greece, Lawrence Erlbaum Associates. 2003.

[6] Gronvall, E., Marti, P., Pollini, A. Rullo, A, Active Surfaces: a novel approach for end-user composition *NordiCHI, Fourth Nordic Conference on Human- Computer Interaction* 2006.

[7] Rizzo, A., Bacigalupo, M., Scenarios: Heuristics for action *Prooceedings of XII European Conference on Cognitive Ergonomics, D.J. Reed, G. Baxter, M. Blythe (eds.)* 2004.

[8] Wikipedia http://en.wikipedia.org/wiki/Scrabble

A middleware of services for spontaneous short-range ad-hoc network of mobile devices

Antonio Sá Fernandes Palmeira Filho
Programa de Pós-Graduação
Departamento de Engenharia de Eletricidade
Universidade Federal do Maranhão
São Luís, MA, Brasil
palmeirafilho@gmail.com

Francisco José da Silva e Silva
Departamento de Informática
Universidade Federal do Maranhão
São Luís, MA, Brasil
fssilva@deinf.ufma.br

ABSTRACT

Mobile devices have evolved from simple datebooks and contacts keepers or voice devices to cutting-edge personal digital assistants, aliasing or combining on-the-fly video and audio acquisition and data processing. But give the best usage to all this new technology in our day-by-day tasks is still challenging. The large variety of devices, the increasing need for ubiquitous mobile services and the network heterogeneity are just some problems developers may find when deploying for the mobile world.

In this paper we describe a framework that has the goal to help developers to easily create applications capable of managing and sharing digital content within groups of people on a spontaneous short-range ad-hoc network. We describe the framework architecture, its services and their relationship, as well as some usage scenarios and related works.

Keywords

ad-hoc, spontaneous, middleware, mobile

1. INTRODUCTION

Nowadays it is undeniable that mobile phones are much more than just a communication device. They evolved from a simple voice transmission device to a sophisticated digital assistant, much more like a PDA, capable of acquiring, processing and storing large amount of data and sharing it via short-range radio connections or even the Internet itself. The personal nature and the increased availability of cheap yet powerful mobile phones and PDAs let us envisage pervasive environments in which we can interact with and exploit its advantages.

However, in order to have useful mobile systems, they must support and enhance various user activities and offer useful, easy to use and effective functionality. Mobile phones are now capable of render interesting content, whether it be video, audio or text. But, use it to share this content still is somewhat cumbersome [11]. Sophisticated devices have the ability to send and receive content using Bluetooth technology [8], a well defined low power shot-range radio connection. Although, choose a resource and send it via Bluetooth is just the edge of the iceberg. It is desirable that mobile systems become able to support a multitude of social interactions [2]. Almost always available and highly personal devices [9], they can easily become a important tool helping people share their experiences or let co-workers get synchronized.

The recent rapid growth of wireless technologies creates a great interest around infrastructureless wireless networks or MANETs, more commonly known as ad-hoc networks. Recently, a more pervasive and popular shot-range infrastructureless wireless network called Personal Area Network (PAN), basically supported by the Bluetooth technology, has been widely studied. In between PANs and MANETs are Spontaneous Networks [4], an infrastructureless network formed by a small group of people that come together in some collaborative activity. In those small infrastructureless networks routing issues are not a primary concern and the collaboration is basically done in a peer-to-peer manner.

The new set of features available on mobile devices and the ability to communicate via spontaneous networks support a rich multimedia interaction model that enables users to share digital assets directly from their mobile devices [13]. But the lack of a well defined development framework avoids its usage on our day-by-day tasks. Our research aims to provide a flexible, simple yet efficient middleware of services without any centralized manager, to enable programmers to easily develop content sharing applications, so users can share their digital assets more effectively. The prototype application, called Spontaneous Network Utility SNU, presents a simple way to share personal (public or private) information and resources (like videos, audios and pictures) within a group of trusted friends. Our middleware provides functions to create and manager groups of friends, scan and seamless connect to nearby devices using the Bluetooth technology (but extensible to any other networking technology), request and transfer resources, like a video stream, and personal informations like addresses, phone numbers, personal interests and so on.

This article is organized as follows: Section 2, Motivation

Scenarios and Requirements, shows some usage applications of the middleware, as well as the requirements we impose to its development; Section 3, The Middleware Architecture, illustrates how we divided the middleware, grouping services and keeping the logical structure simple and clean; Section 4, Implementation, points some of the middleware implementation characteristics and issues; Section 5, Related Work, describes relevant project related to our work; and on Section 6, Conclusion and Future Work, we wrap up our conclusions and show what we plan for the future of this work.

2. SCENARIOS AND REQUIREMENTS

Think you are in a pub, having some fun with your friends or in an airport waiting for that vacation flight. In both scenarios you may want to spend some quality time and search for new friends would be useful. You would look for people with your same age or interests; maybe see some public information about them, like their name, nickname and a personal photo. In case you find something interesting, you may want to chat anonymously until you think your new friend worth a try. You can send a picture of your neighborhood and play a hide and seek game.

Let's think you are now on a business meeting. It can be very handy if you could share some documents with a group of co-workers and at the same time use a small whiteboard to point a detail on a diagram. Perhaps, you could like to seamless broadcast a video to the meeting participants. At the end of the meeting you could use a common agenda to schedule the next reunion.

Students may share their classroom annotations, exercises and teacher's notes. They may need to quickly establish a virtual chat room to post common questions and answers, as well as save the conversation log. They could even seamless exchange jokes when watching classes [2].

All these scenarios can be possible using a personal device with the ability to discovery and communicate within the close geographic area. Our middleware intend to make all this possible by providing a simple and easy to use set of services where a collaborative application can be built on quickly and effectively.

With these scenarios in mind, we based our middleware on the following requirements:

1. *No centralized control*: our middleware is intended to be used on a spontaneous ad-hoc network, where components will collaborate in a spontaneous manner without a centralized server hooked to an infrastructured network. The absence of a centralized control imposed by the ad-hoc model force our device to act as client and server simultaneously, storing group, person and device informations as well as security data to tell what each user or group can do. Our idea is provide resource sharing without any particular infrastructure available, so users will be free to share their assets wherever they go;

2. *Easy on device resources*: the communication between devices must be fast to save battery resources, but the nearby device discovery process must be silent and transparent. The middleware needs to save groups, people (with their devices), shared resources and data information but should not use large chunks of persistent memory. Save device's resources is particular important once in general mobile devices have modest specifications and the most available space should be used by the users data, not the application itself;

3. *Network technology independency*: the usage of the Bluetooth technology is desirable but cannot be the only way to interact, once we know there are others ad-hoc network technologies, like specified on the 802.11 standard [1]. The middleware must be designed on layers in order to separate the network technology from the discovery and communication routines. Allowing multiple network technologies is important to make our solution wide available and this requirement suggests the use of services layers on the entire project;

4. *Software portability*: we believe that in order to make this middleware as much potable as possible, the usage of Java 2 Micro Edition (J2ME) is desirable [14]. J2ME is present in almost every mobile device on the marketing and its ability to work with the Bluetooth stack make it a good solution. We based our middleware on devices capable to run J2ME with Connected Limited Device Configuration (CLDC) 1.1 and Mobile Information Device Profile (MIDP) 2.0 specifications. These specifications define a set of attributes that should be, but not must be, implemented and available on all devices. Manufactures are free to decide what to implement, which let us deal with a number of devices that does not have all the functions we need. For our development, we defined that the middleware should run on an environment with the PDA Profile for J2ME (JSR 75) and Bluetooth/OBEX for J2ME (JSR 82) additional APIs.

3. THE MIDDLEWARE ARCHITECTURE

In order to be flexible, robust and has a well defined interface between the middleware and the application, we divided our solution in layers, where each one provides services to the others. Working in layers give us the ability to add or replace functionality by changing or adding layers with the same set of interfaces.

Figure 1 shows the basic schema of layers we defined. The Network Layer is attached to a specific network technology and it provides input and output functions encapsulated as method to send middleware commands and receive their answer. The Services Layer holds all available user services in our solution. It has all the interfaces used by developers when creating end-user applications (at the top of the diagram). The user application doesn't communicate directly with any component of the Network Layer, as the Network Layer doesn't send any data straight back to the user application.

3.1 The Network Layer

The **Network Layer** is actually implemented for the Bluetooth technology but, as we use a well-defined interface, we can easily extend this model using any network technology.

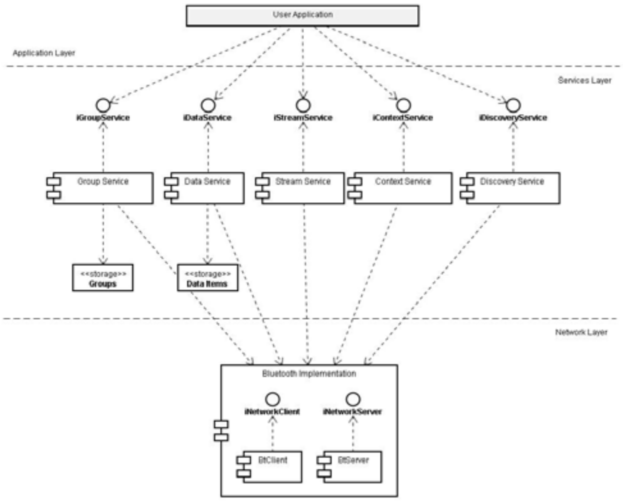

Figure 1: Component Diagram

The network layer defines a client and a server implementation, since in an ad-hoc network the device must act as both during the application execution. There are no low level methods, like send a byte through an open stream, available to the Services Layer. Instead of, the network client interface has a method to send a command (with its parameter) to a target person using its device's connection string. On the other hand, the network server interface has a method to listen for incoming connections, accept them, receive and process commands, and send the response back to the questioning device. This model encapsulates all network related job and presents just a simple interface to send and process commands on devices.

Another asset of our network layer is the fact that it is responsible for holding all the data directly associated to any network call, like scan for nearby devices, store their names, connection string and unique ids, find a desired device and make them available to any other layer. Search for public information, request a file transfer and store any recently file data transferred are examples of methods provided by the network layer.

3.2 The Services Layer

The **Services Layer** has all the available middleware services. As Figure 1 shows, this is the only layer visible to the user application. There are services to deal with group management, data share, data stream, context information and service discovery. All these services use the interfaces provided by the Network Layer and by no means they have directly access to the network medium in use.

3.2.1 The Group Service

The **Group Service** holds our notion of groups. As described on Section 2, we intend to develop a solution for a spontaneous network, which there is no server where we can synchronize any sort of data. The idea behind our group philosophy is gather people that will share the same set of security assets together but each user is completely responsible to create and manager his/her groups locally. Even though users can make a group public and let friends download its configuration, there is no centralized server or no distributed solution to store and synchronize groups. Spontaneous networks don't guarantee that once we distribute information we will be able to have it consolidated again, that's why each user stores its own groups definition.

A group is basically a set of people. The highly personal facet of mobile devices [9] let us realize that they are in general used by one user and they are barely shared between users. In our approach, a device is recognized as a person, even though we know a user may have more than just one device. In this case, we save all device's ids used by the person, so we can distinguish them. If a user creates a data item and wants to share it with some friends, he/she may create groups and set the data item security vectors as desired. For example: to share a data item with a Peter, this data item must have Peter associated with its vector of trusted people or have a group that has Peter associated with its vector of trusted groups. If a person is not listed on the vector of trusted people neither is an element of a group listed on the vector of trusted groups, this person will not see the public data item. Users can freely create and delete groups, insert and erase people in groups and use them to set security assets of available data items. Once a group is created or modified, its information is stored on the available persistent memory so the user doesn't have to re-enter it. Every time the application is started and the Group Service object is created, the group's information is loaded from the persistent memory and made available to the entire framework.

When we designed our notion of groups, we had an opportunity to use a sort of distributed algorithm for group management [12, 3]. However, all these algorithms are based on a low cost and highly available network. On mobile devices, use a network connection means spend additional power, and this is why we cannot have a connection always available. One of our requirements is save at most the available power resource so we cannot afford a solution that drains to much battery. Following this argument, we chose an approach that center the group information on its own device and where there is no option to synchronize groups.

3.2.2 The Data Service

The ability to seamless share data without a centralized control is one of our requirements. In order to achieve this goal, we have the **Data Service**, a service responsible to create, edit, erase and share data items. A data item is a simple pair of strings (item name and value or name and file path) with some security information. Following our security model, any data item can be remotely read if, and only if, it has a group that contains the questioning person or if it knows the name of the person itself. I.e., when a user search for public assets it will receive a list of available data items just for those items that match this security validation.

If the request passes through the security validation, the data item is sent to the person. As there is no server, the information is sent to the questioning device every time it is requested and only the user who created the content has a valid copy of the data. The middleware doesn't contemplate any data synchronization feature but such service can be implemented on the user application level. If the user application makes a copy of any friend's data, it will be responsible for searching for changes and updates its copy. Data items can be completely public, which means that any person can read its information. If the information is a file (storing the file path), the system simply transfers the file content to the questioning device. Save or just view the information is a matter of the user application implementation.

Data items can be used to save any personal or business information as name, address, phone numbers, emails, occupation, interesting, and so on. As well as it can store job related information, like costs of trips for a taxi cab, or even a pub menu, with beverages, meals and deserts. There is no limitation for the file sharing feature, so users can share photos, sounds or text documents, like their own resume. Users can create a sort of professional portfolio with pictures and diagrams of their work and care it with themselves all way around. The possibilities are endless and our middleware tries to make them easy and seamless.

3.2.3 The Stream Service

The idea behind the **Stream Service** is make available a text, audio or video stream. It will have the same security logic of the data item on the Data Service, but with only these three types of streams. So, if a user wants to share its camera with some friends, he/she could create a public video stream and his/her friends would be able to see on their devices the images that have been captured in real time. The same could be done for audio streams, creating a kind of walkie talkie and for text streams, like on a conference room.

Data streams are a very power consuming feature once we have to sustain a network connection during all the process. Compression algorithms are a powerful tool to save network time and keep the best real-time experience possible but we know there is a serious power limitation that we have to live with. Even though this is for sure a fun feature, it will have to be used in a very brief manner until we have more powerful batteries on the market.

3.2.4 The Context Service

The **Context Service** is designed to share personal context information, like the users mood or his/her ring tone volume (which can potentially indicate his/her availability), as well as device context information [9], like available storage space or battery life, user profile or location. This service is especially useful to help the user application decides what to do when some conditions are achieved. For example, the user application can wait until the device general volume is not muted (when the user is potentially available) to send a SMS message to the person asking to share its audio stream.

Some context information must be fed by the user, like his/her mood, but others can be read straight from the device, like its battery life or ring volume. Some context information can also be made public and shared using the same notion of groups we designed, like his/her mood, and some of them are especially useful to the user application, helping on taking some work decisions, like decrease the real-time video resolution because the compression process is drain out the battery too fast. Context information can provide special adaptive hints to the user application so it can change its behavior on-the-fly.

3.2.5 The Discovery Service

With all these services we need a way of knowing which ones are available. To fix this gap we have created the **Discovery Service**. This service searches for available services on a target device and make this information accessible to the middleware and the user application. The user application may have services by itself and it may use this service to publish them as well.

To archive this goal, the middleware needs a feature to search for available native middleware services, as well as user-defined ones. Services must implement a well-defined service interface so the middleware can identify them, read their public information (like name and id) and make them available to the user application. This interface has a set of methods to initialize the service, send commands, read their answer, processing status and error messages.

4. IMPLEMENTATION

Our implementation actual status contemplates the Network Layer with a Bluetooth implementation, and the Group Service and the Data Service (already with the file transfer method) from the Services Layer.

4.1 The Network Layer

As a way to apply network independency, the middleware uses the *Factory Patern* [5]. A factory of network objects knows how to create network servers and clients, as we have defined our network factory interface, as well as the server and client interfaces. The middleware allows programmers to easily create a network factory for each available network technology, as they need. In our experimental application, we created a Bluetooth Factory (`btFactory`) class who creates Bluetooth Servers (`btServer`) and Bluetooth Client (`btClient`) classes implementing the network server and network client interfaces. The Services Layer itself doesn't know any detail of the Bluetooth technology but knows the set of methods used to access network servers and clients and use the network factory set of methods to create the desirable network object. The application can decide which network technology use by a command line parameter, a specific auto-located device configuration or even an user selection.

The interfaces we defined for a network factory, as well as for a network server and client are shown in Figure 2. The methods declared on the `iNetworkFactory` interface (`createNetworkServer()` and `createNetworkClient()`) simply create the respectively objects but doesn't start their operation. The `iNetworkServer` method `startServer()` starts the network server using a reference to the control object (which holds the references of the middleware services) and a vector of initialization attributes (default timeouts, for instance). The `startClient()` method from the `iNetworkClient` interface starts the client with a reference to the same control object, a command (what to do), a parameter to the command (if any) and the network connection string to use (maintained by an internal vector of nearby people - the connection string has a reference to an

```
public interface iNetworkFactory {

  iNetworkServer createNetworkServer();
  iNetworkClient createNetworkClient();

}

public interface iNetworkServer {

  void startServer(
     control param,
     Vector attrib
  );

}

public interface iNetworkClient {

  void startClient(
     control param,
     String cmd,
     String parameter,
     String pconnStr
  );
  Vector nearby_people();
  person findPeople(String dname);
  void scan_nearby_people();
  void scan_ForeignPublicData(int person_pos);
  Vector get_ForeignPublicData();
  void get_FileData(String dataItemName, int person_pos);
  byte[] get_ImportedFileData();

}
```

Figure 2: iNetwork Interfaces

available channel which can be different on each Bluetooth network scan). The idea behind this model is to have a set of commands available to the network client to be processed on the network server of the target device. For instance, some of the possible commands are "`PublicGroups`", to retrieve a list of visible public groups; "`PublicData`", to get the list of available public data items; and "`getFile`", to actually receive a file content. Everything is always transferred between devices as an array of bytes, and stored on class's internal arrays if necessary.

The network layer should take care of any network specific job, such as store the information of devices in range. On the scanning for potential mates front, we have developed methods to scan for nearby people, `scan_nearby_people()`, retrieve the vector of people found , `nearby_people()`, and to find a specific person on this vector, after a successful scan `findPeople(String dname)`. On the data sharing front, there are methods to scan for public data on a nearby device `scan_ForeignPublicData()`, retrieve the available data `get_ForeignPublicData()`, as well as, request a public shared file from a remote device `get_FileData()` and retrieve its content `get_ImportedFileData()`.

Figure 3 shows a sequence diagram for a network call to retrieve the public groups shared by a user. The user application class `groupServiceForm` uses the `groupService` class method called `getPublicGroupsFromPerson()`, an implementation of the `iGroupService` interface (in details on the next section). The `groupService` using a network client, implemented by the `btClient` class (Bluetooth client implementation), uses the `startClient()` method to send the "`PublicGroups`" command to the network server implementation (`btServer`) on the target device. The network server object process the command and send the response back to the network client as an array of bytes. The network client object interprets the answer and make a call to the method `update_imported_public_groups()` of the `groupService` object to update its vector of imported groups, which calls the method `showAvailablePublicGroups()` available on the `groupServiceForm` object to show the query result.

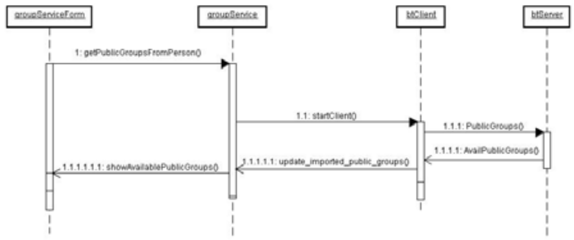

Figure 3: Network Call Example

As can be seen, there is no direct usage of input or output streams, as well as there is no low level detail of how to establish a network connection. Everything is encapsulated and detached from the rest of the implementation.

On ad hoc networks, the environment places strict limitations due to the lack of any network infrastructure. Every device must act like client and server at the same time, once we don't have a server hooked to a infrastructured network to run a middleware service that intermediate the communication. This P2P scenario on an ad-hoc network let us deal with the fact that at anytime a previously accessible device can be out of range without notice. So, in order to establish a successful connection it is necessary to scan the neighborhood for available devices before try to send any data. We cannot trust that once a device was found it will be accessible at all time. A device can be out of range in a matter of seconds, which makes unfeasible to develop communication routines that believe to have trustable connections. We need to send all needed data in on blast because the second one can be too late. Another issue that supports this "send it all in one blast" technique is that radio connection are in general too energy consuming, so keep a connection alive could drain out the devices battery so fast that it would make our solution useless.

4.2 The Group Service

Our notion of group is based on the people, but as was said on Section 3.2.1, as people can have multiple devices, we had to store a sort of ID for each device. On our test application we use the name used by the user to identify him on the network. For instance, on the Bluetooth implementation this is the friendly name used on the device. On each network implementation we can use any variable for this name, including a user parameter. This approach has two basic consequences: first, it doesn't enforce attach the user to a device, so the user can use the same name for more than one device, and the middleware will see it like one single user (even though he/she is using different devices); second, as a direct cost of the first consequence, there is a chance of two users with the same name has their security assets misunderstood. In order to fill this gap we save a unique ID for each device, like MAC address [1] or Bluetooth local address [6]. This device ID is used only when the middleware finds two or more devices with the same name and needs to distinguish between them. At this point the user may be asked to define how the middleware should treat this device, as a new one or not.

```
public interface iGroupService {

  void setGroupServiceForm(iGroupMangForm param);
  Vector groups();
  group findGroup(String gname);
  boolean addGroup(String gname);
  boolean changeGroupVisibility(String gname);
  boolean delGroup(String gname);
  int addPersonToGroup(String gname, person per,
   boolean add_if_exist);
  int delPersonFromGroup(String gname, person dv);
  void getPublicGroupsFromPerson(int personVectorIndex);
  Vector imported_public_groups();

}
```

Figure 4: iGroupService Interface

The Group Service interface called `iGroupService` can be seen on Figure 4. There are methods to retrieve, add, delete and find groups, as well as change group's visibility - groups can be set to public (`group.publicGroup = true`), where everybody can read/import its composition (name and the list of people that actually form the group), or private, where this information is hidden. As groups are formed by people (where each one can have more than one device), there are methods to deal with them, like add or delete a person to/from a group. Following the same idea, there are methods to request to get the public groups from a person and to retrieve them: `getPublicGroupsFromPerson()` and `imported_public_groups()`, respectively.

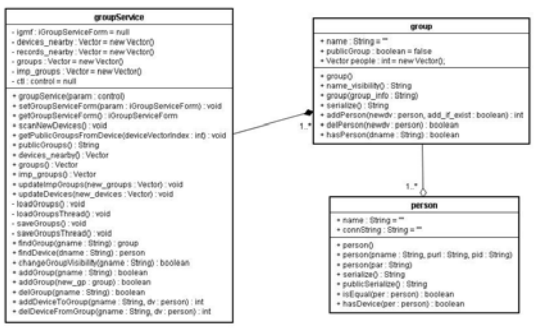

Figure 5: The Group Service Classes Diagram

Implementing the Group Service interface shown in Figure 4, there is the Group Service class shown in Figure 5. Besides all the public methods declared on the `iGroupService` interface and the respectively vectors to store their information, we have methods to asynchronously load and save groups using threads (J2ME best practice - use thread every time there is a potentially blocking procedure - in this case, access the persistent memory)[1], set and retrieve the reference to the form used to manager groups (used to tell the `iGroupServer` implementation which form object needs to receive the callback - in details on Figure 6) and generate a string with all available public groups (those with public visibility) to support an answer to the network server command "`PublicGroups`".

Figure 5 still shows the class signature for groups and people. The group class has a name, a vector of people and an indication of the public status. There are methods to serialize a group and create one from its previously serialization, as well as methods to deal with the vector of people. Simpler, the people class has a name, an url to store its connection string used to establish a network connection, a vector of devices used by the person, a serialization method, a simple constructor and a comparison method, used to verify if two people are the same.

Since on J2ME we cannot tell the environment when to update the screen (it does it when the application main thread is free), the middleware uses a callback approach in order to synchronize an asynchronous network call. On a common implementation we could just use a class to deal with the communication and another one to keep the user posted, and synchronize them using a simple `thread.join()` method. However, you cannot force the screen update before call the `join()` method, and the system doesn't update the screen until its thread is free, what happens just after the blocking call to `join()` is completed. So, in this scenario the user got not screen update in time and cannot understand what is going on in background. The only way we found to solve this problem is the usage of the callback methods. A class creates an object with the arguments needed to do the background task and just ends its method execution, waiting to receive a call to a callback method with the expected result. This is the reason why we have methods to set and retrieve the application form (which needed to have its own `iGroupServiceForm` interface) used to interact with the group service class. It is not a clean way to solve the problem but it does the job.

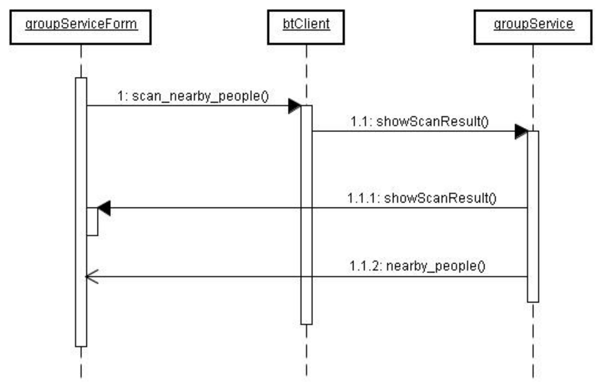

Figure 6: Callback Example

On Figure 6 we have a callback method example. When a user request a search for people on the neighborhood, the `groupServiceForm` object calls the `scan_nearby_people()` method of the `btClient` class and ends its method execution. This call will update a vector of found people. When the network client object ends its operation, it calls

[1] If we don't use threads for potentially blocking calls, the J2ME compiler always echoes a warning alert.

the `showScanResult()` method on the `groupService` object which uses the `showScanResult()` method of the form object in order to update the user screen. The `groupServiceForm` object needs the vector of found people and in order to read it, it calls the `nearby_people()` method on the `btClient` object (responsible for holding all the nearby people information). As shown, the `btClient` makes a callback to the `groupService` object which makes a callback to the `groupServiceForm` object to in fact finish the entire procedure.

4.3 The Data Service

The Data Service interface `iDataService` is shown on Figure 7. It is a straight forward service that has methods to add, delete, update and find a data item. There is also methods to retrieve all available items, as well as create a serialized string with all public data to a given person - it goes through all data items selecting those who have the given person on its access control list, and creates a string (for communication) with their serialized data. For file transfers, the service uses the network client command "`getFile`" (via the `startClient()` method of the `iNetworkClient` interface).

```
public interface iDataService {

  String publicData(String person_name);
  boolean updateItem(dataItem new_di);
  boolean addItem(dataItem new_di);
  boolean delItem(String iname);
  Vector get_items();
  dataItem findItem(String iname);
  void setDataServiceForm(iDataServiceForm fparam);

}
```

Figure 7: iDataService Interface

Figure 8 shows the class diagram for the `dataService` class which implements the `iDataService` interface. As happens on the `groupService` class, besides all the public methods declared on the `iDataService` interface and the respectively vectors to store their information, there are methods to asynchronous load and save data and a method called `get_LocalFiles()` to read the directory content in order to support the creation of data item that represents files.

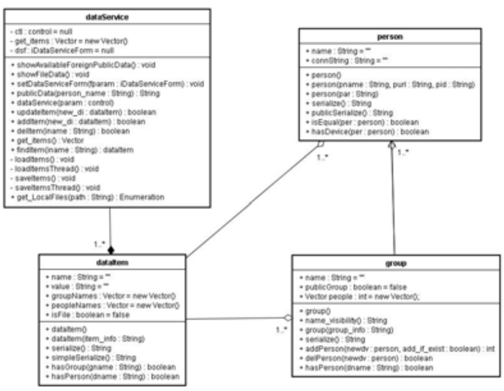

Figure 8: The Data Service Classes Diagram

Still on Figure 8 is shown the data item class signature we've created. The data item it a pair of name and value, which can be a simple string or a file path (when the data item is a file). Following the same security logic we used for groups, the data item has a vector of trusted groups and a vector of trusted people that work like an access control list. There is a constructor method that uses a previously serialized data item to create a new one, as well as methods to verify if the data item vector of trusted group has a given group (`hasGroup()`) or if the data item vector of trusted people has a given person (`hasPerson()`). These methods are used when the network server receives a command to retrieve the public data items for a questioning person ("`PublicData`" command). The service needs to run though all data items selecting those who the questioning person is added to the access control list (it is present on the vector of trusted person or it is an element of a group present on the vector of trusted groups).

5. RELATED WORK

The MIRES - an Information Exchange System for Mobile Phones project [7] aims to provide a mobile resource management system that plays the role of a data sharing manager: data resources created on the mobile device reside at the server side and are selectively downloaded to the mobile clients. When users want to share the resource items, he/she tells MIRES which, when and with whom resource should be shared, and the system takes care of the data synchronization, versioning control and data distribution exclusively on the server side. As we require no server interaction, our middleware is completely serverless and by no means we have any kind of centralized controller. The data share is done entirely on the mobile device and synchronization is a matter treated by the user application. We drive our solution to a spontaneous network where people can join and left the neighborhood at any time.

The PGWW, Personalized Group Wide Web project [10] aims to provide a flexible, simple, and efficient mechanism to enable next-generation cell phone users to share their digital assets. It is based on the WWW model, where all the resource is created and managed on the mobile device. It can share different types of media, including some contextual information, and providing access control policies. It uses a set of network services to allow the resource sharing although the user can control everything from the mobile device. The network services act like a mediating agent creating a view of shared resources and caching the mobile data avoiding further interactions. As happens with the MIRES project, the PGWW rely on a infrastructured network with a server, what is the opposite of our goal. However, the ability to control everything from the device, share any kind of resource and the usage of access control policies consist on a great similarity with our middleware.

Aiming the on-growing smart phone world, the *Context-Phone* project [9] is a set of four interconnected modules providing an entire new phone environment with context-aware applications. It was developed to run on off-the-shelf mobile phones using Symbian OS and it comes with replacements for almost all phone internal applications, such as the Contacts and Call Log. The idea is add context information like location, mood and phone usage to the internal applications. So the new ContextContacts application can say not only the budy phone number and address, but its actual phone profile, ring tone volume, vibrating status, and even its geographic location. As we can see, this project is pretty much for the Symbian smart phone world. It is against our goal to run on any mobile device with J2ME, but its idea of seamless share context information, as well as the fact that it doesn't require a server to run, is an undutiful similarity with our project. However, there is no data share besides the context information and by no means there is any resource share like video or audio streams.

6. CONCLUSION AND FUTURE WORK

Applications that take advantage of emerging short-range radio technologies and allow peer-to-peer collaborative resource sharing will be in our not so distant future. This paper describes a middleware of services designed specially to speed up the development of collaborative peer-to-peer applications on ad-hoc short-range spontaneous networks of mobile devices. It provides a set of services that encapsulates common issues that rises when developing to spontaneous networks, like group management, resource and data share, and file transfers. It is written in Java, a language known by its portability and power. Through this middleware, developers have an option to quickly deploy powerful applications able to make our social interactions richer and funnier.

The future work of this middleware includes the support of text, audio and video streams (the Stream Service), the support of context information (the Context Service), and the development of some testing application, able to use the services we created and fulfill the scenarios we illustrated.

7. REFERENCES

[1] I. S. Association. 802-11:1999/amd 1:2000(e) Standard Specification, June 2003. http://standards.ieee.org.

[2] R. Beale. Supporting social interaction with smart phones. *PERVASIVE Computing IEEE CS and IEEE ComSoc*, pages 35–41, 2005.

[3] G. Coulouris, J. Dollimore, and T. Kindberg. *Distributed Systems: concepts and design.* Addison-Wesley, 2000. 3rd edition.

[4] C. Elliott and B. Heile. Self-organizing, self-healing wireless networks. *IEEE Int'l Conf. on Personal Wireless Communications*, pages 355–362, 2000.

[5] E. Gamma, R. Helm, R. Johnson, and J. M. Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software.* Addison Wesley Professional, Oct 1994. ISBN: 0201633612, 416 pages.

[6] M. Inc. Java api for bluetooth wireless technology (jsr 82), Sep 2005. Specification Version 1.1.

[7] Q. Li, X. Li, J. Zhai, and L. Wenyin. Mires - an information exchange system for mobile phones. *ACM Symposium on Applied Computing*, pages 1196–1200, 2004. City University of Hong Kong.

[8] R. Mettala. Bluetooth protocol architecture. Technical report, The Official Bluetooth Membership Site, Aug 1999. Doc. 1.C.120/1.0.

[9] M. Raento, A. Oulasvirta, R. Petit, and H. Toivonen. Contextphone: A prototyping platform for context-aware mobile applications. *PERVASIVE Computing IEEE CS and IEEE ComSoc*, pages 51–59, 2005. University of Helsinki and Helsinki Institute for Information Technology.

[10] M. Roman, N. Islam, and S. Shoaib. A wireless web for creating and sharing personal content through handsets. *PERVASIVE Computing IEEE CS and IEEE ComSoc*, pages 67–73, 2005. DoCoMo USA Labs, San Jose, CA.

[11] B. N. Schilit and U. Sengupta. Device ensembles. *IEEE Computer Society, Computer*, pages 56–64, Dec 2004.

[12] A. S. Tanenbaum and M. van Steen. *Distributed Systems: principles and paradigms.* Prentice-Hall, 2002.

[13] R. Want, T. Pering, G. Danneels, M. Kumar, M. Sundar, and J. Light. The personal server: Changing the way we think about ubiquitous computing. *Proc. 4th Intl Conf. Ubiquitous Computing*, June 2002. Intel Research, Santa Clara, CA.

[14] M. J. Yuan. Access web services from wireless devices. http://www.javaworld.com, Aug 2002. Wireless Java Session, Java World, Fueling Innovation.