MUM 2003

Proceedings of the 2nd International Conference on
Mobile and Ubiquitous Multimedia,
10–12 December, 2003
Norrköping, Sweden

organized by

Linköpings universitet and the Santa Anna IT Research Institute

In cooperation with
ACM,
SIGCHI,
SIGGRAPH,
SIGMOBILE

Edited by

Mark Ollila and Martin Rantzer

Published for ACM Press by
Linköping University Electronic Press
Linköping, Sweden, 2003

ACM PRESS

The Association for Computing Machinery
1515 Broadway
New York New York 10036

Notice to Past Authors of ACM-Published Articles

ACM intends to create a complete electronic archive of all articles and/or other material previously published by ACM. If you have written a work that was previously published by ACM in any journal or conference proceedings prior to 1978, or any SIG Newsletter at any time, and you do NOT want this work to appear in the ACM Digital Library, please inform permissions@acm.org, stating the title of the work, the author(s), and where and when published.

ACM ISBN: 1-58113-826-1

Distributed by Linköping University Electronic Press
Printed by UniTryck, Linköping, 2003
Additional copies may be ordered from:
Linköpings Universitet
Linköping University Electronic Press,
SE-581 83 Linköping, Sweden
E-mail: ep@ep.liu.se

Grant of Permission

The Association of Computing Machinery, Inc. (ACM) hereby grants to Linköping University Electronic Press, located at http://www.ep.liu.se/, permission to mount and serve digital copies of the ACM copyrighted Proceedings of the 2nd international Conference on Mobile and Ubiquitous Multimedia (December 10–12, 2003, Norrköping, Sweden). ACM grants this permission without charge. ACM understands that Linköping University Electronic Press wishes to provide open access this publication without any restrictions and ACM hereby grants Linköping University Electronic Press permission to do so. ACM asks that Linköping University Electronic Press display a link to http://www.acm.org/pubs/copyright_policy/ with the following message in association with the title of this volume:
"© ACM Press 2003. Linköping University Electronic Press gratefully acknowledges ACM Press for granting permission to make this work freely available from our servers."

Deborah Cotton
Copyright & Permissions
ACM Publications
1515 Broadway, 17th floor
New York, NY 10036
212.869.7440 ext. 652
Fax: 212.869.0481
permissions@acm.org

Table of Contents

Welcome to MUM 2003 in Norrköping!

We meet again, for the second international conference on Mobile Ubiquitous Multimedia (the first one was in Oulu, Finland.) As always it is exciting to see what we can do to evolve both the conference and the area as such, with this year comprising of keynotes, papers, demonstrations, tutorials and a workshop. Just as with the conference the area of Mobile Ubiquitous Multimedia is now leaving the infancy and taking its first steps in to general applications.

To reflect, we need only look at the acronym of MUM, and see we have the terms Mobile Ubiquitous Multimedia. We need to look at these words backwards to understand the historical context of what MUM is about.

Multimedia, a term that is often misunderstood, can be defined as some form of application or transmission that combines media of communication (text and graphics and sound etc.). It is also a research area that has existed during the rise and fall of the Internet bubble with the first ACM conference in Multimedia occurring in, what seems an eternity, 1994.

With "Ubiquitous" we have the "being or seeming to be everywhere at the same time." This is an active research area with many researchers from cross disciplinary fields working together.

The final word we have is "Mobile". In this age of mobility, mobile research is a growing research area. With "Mobile" we mean the capability of moving or of being moved readily from place to place. By adding Mobile to Ubiquitous Multimedia, we enter the realm of truly exciting research across many disciplines.

The once so bold vision of ABC (Always Best Connected) is becoming something we can take for granted and use as a foundation for new and exciting services. Because it is in the services we will see the most exciting developments. Sure there is more to do within the areas of infrastructure but given the rule of Pareto we should be able to do "80% of what is possible by using only 20% of what is available".

So it is only natural that the papers of the second conference deal more with the "what to do" than "how to" do it. We are quickly moving from what is possible to what is practical and then beyond what is imaginable. So maybe the theme for next year of the conference will be "More Unusual Magic".

The goal of the MUM 2003 conference is to provide an international forum for presenting recent research results mobile ubiquitous multimedia, and to bring together experts for a fruitful exchange of ideas and discussion on future challenges.

The heart of the technical program are keynote presentations and tutorials by leading experts, whom we wish to warmly thank for accepting our invitation. We wish to thank the program committee and contributing reviewers for putting together a strong program,

which, by spanning from networking and human computer interaction to applications and case studies provides a comprehensive view of the field. Last but not least, special thanks are due to the organizing committee for making the conference possible.

The main sponsor of the conference is Santa Anna IT Research Institute AB, a part of the National Swedish IT Institute, a meeting point for information technology researchers from the University of Linköping and the Local IT industry. Other major sponsors are Nokia, Microsoft, Ericsson, Norrköping Visualization and Interaction Studio, Norrköping Council, and the Department of Science and Technology at Linköping University.

We would like to express our gratitude and warm welcome to the keynote speakers, lecturers of the tutorials, authors of contributed papers, and other participants. We wish you a most pleasant stay in Norrköping. Finally, we thank ACM, and particular, SIGCHI, SIGMOBILE, and SIGGRAPH for their in cooperation support.

Martin Rantzer
Chair, Program Committee

Mark Ollila
Chair, Organizing Committee

MUM 2003 Program Committee

MUM 2003 consisted of experts in the field from all over the world. We thank them for their comments and reviews.

Berglund, Erik (Demo Chair) Linköping University, Sweden

Brown, Barry Glasgow University, UK

Carlshamre, Par, Linköping University, Sweden

Chalmers, Alan, Bristol University, UK

Crisler, Ken, Applications Research group at Motorola Labs, USA

Doermann, David, The University of Maryland Institute for Advanced Computer Studies (UMIACS), USA

Dykstra-Erickson, Elizabeth, Kinoma Inc, USA

Ebert, David S., School of Electrical and Computer Engineering, USA

Eriksson, Henrik, Linköping University, Sweden

Holmlid, Stefan, Linköping University, Sweden

Holmquist, Lars-Erik, Viktoria Institute, Sweden

Höök, Kristina, Swedish Institute of Computer Science, Sweden

Juhlin, Oskar, Mobility Interactive Institute, Sweden

Kuutti, Kari, University of Oulu Finland

Nadjm-Tehrani, Simin, Linköping University, Sweden

Niemela, Eila, VTT, Finland

Ojala, Timo, MediaTeam, Oulu University, Finland

Ollila, Mark, (Local Chair) Linköping University, Sweden

Pulli, Kari, Nokia, Finland

Rantzer, Martin (Program Chair), Swedish Defence Research Agency and Santa Anna IT Research Institute

Rauterberg, Matthias, Technical University Eindhoven, Netherlands

Shahmehri, Nahid, Linköping University, Sweden

Steinhage, Axel, Infineon Technologies AG Munich Germany

Veijalainen, Jari, University of Jyväskylä, Finland

MUM 2003 Organizing Committee

Bakos, Niklas, Linköping University, Sweden

Berglund, Erik (Demo Chair), Linköping University, Sweden

Carlshamre, Pär, Linköping University, Sweden

Erlandsson, Birgitta, Santa Anna IT Research Institute

Henrysson, Anders, Linköping University, Sweden

Hjalmarsson, Jonas, Linköping University, Sweden

Holmlid, Stefan, Linköping University, Sweden

Ollila, Mark, (Local Chair) Linköping University, Sweden

Rantzer, Martin (Program Chair), Ericsson AB and Santa Anna IT Research Institute

Tubbin, Erika, Linköping University, Sweden

Interaction on the Go

Mark Billinghurst
Human Interface Technology Laboratory (New Zealand)
mark.billinghurst@hitlabnz.org

The WIMP (Windows, Icon, Menus, Pointers) metaphor is one of the most successful user interface paradigms ever developed. Even though the mouse was first demonstrated nearly 40 years ago, Englebart's invention is still the dominant tool for computer interaction.

However, as computing moves from the room scale into the hand and onto the body there are increasing opportunities for more innovative forms of human computer interaction. This is particularly true with mobile devices where users do not have the luxury of a large screen, full sized keyboard, or even a pointing device. Luckily, exponential advances in computer processing, graphics, networking and storage have enabled a wide range of other user interface techniques and devices. We are now entering a post-WIMP era where interface designers have a wide range of approaches to choose from.

Three emerging interface trends are particularly suited to mobile multimedia devices; Augmented Reality (AR), Perceptual User Interfaces (PUI) [Turk 2000] and Tangible User Interfaces (TUI) [Ishii 97]. AR interfaces superimpose virtual imagery over the real world, so that both reality and virtual reality are seamlessly blended together. PUI use cameras, and other sensing devices to give computers some of the same perceptual capabilities of humans. Finally, TUI bridge the worlds of bits and atoms by enabling the user to interact with digital information by manipulating real objects.

Each of these research fields is important its own right and all explore innovative ways to interact with computers. TUI techniques are used to provide intuitive input, AR interfaces support enhanced display, and PUI provide expert reasoning that connects input to display.

However, in many cases interfaces in these individual areas introduce artificial seams and discontinuities into the workspace. Seams are spatial, temporal or functional constraints that force the user to shift among a variety of spaces or modes of operation [Ishii 97]. For example, the seam between computer word processing and pen and paper makes it difficult to produce digital copies of handwritten documents.

Natural seamless interaction is important for mobile devices. When interacting on the go, users need the least amount of distraction and the most intuitive ways of interacting with information. However, seamless interaction does naturally occur in the overlap of AR, Perceptual and Tangible User Interfaces (fig. 1). Research in this area explores how to enhance interaction with virtual imagery by using machine perception and cognition, and real object manipulation.

The use of real objects to manipulate virtual content is often referred to as Tangible Augmented Reality. This is an exciting new area of interaction design possibilities for mobile devices. Tangible AR interfaces provides true spatial registration and presentation of 3D virtual objects anywhere in the physical environment, while at the same time allowing users to interact with this virtual content using real objects. In this way the interface is moved from the screen space into the familiar real space.

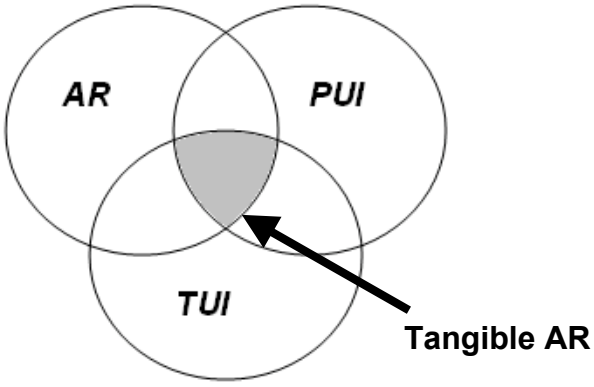


Fig. 1. *The convergence of AR, PUI and TUI metaphors*

The ability to seamlessly interact with the real world is important for mobile devices. PUI techniques can be used to provide the devices with awareness of the user's location and state. Context cues such as these can be used to customize the information presented back to the user. AR techniques can then overlay information in such a way that it doesn't interfere with the users actions in the real world.

There are several advantages of Tangible AR interfaces for mobile devices. First, they are *transparent interfaces* that provide for seamless two-handed 3D interaction with both virtual and physical objects. Users can manipulate virtual objects with the same input devices they use in physical world – their own hands. Tangible AR allows *seamless spatial interaction* with virtual objects anywhere in their physical workspace. The user can pick up and manipulate virtual data as easily as real objects, and arrange them on any working surface, such as a table. The digital and physical workspaces are therefore continuous, naturally blending together. Finally, the *physical form-factor* of mobile devices suggests how they are to be used. The device itself can be used as an intuitive interface object facilitating natural interaction.

Tangible Augmented Reality blends elements of AR, Perceptual and Tangible interfaces. Technologies in this area do not separate actions in the physical environment from the digital domain and so support seamless interaction. In this way they enable us to move beyond WIMP interfaces and enhance interaction in the real world. It is metaphors such as this that will be necessary for supporting interaction on the go.

References

Ishii, H., Ullmer, B. Tangible Bits: Towards Seamless Interfaces between People, Bits and Atoms. In proceedings of CHI 97, Atlanta, Georgia, USA, ACM Press, 1997, pp. 234-241.

Turk, M., Robertson, G. Perceptual User Interfaces *Communications of the ACM*, March 2000.

Pretending to See the Future

Kristoffer Åberg
Sony Ericsson Mobile Communications AB
kristoffer.aberg@sonyericsson.com

The relationship of computers to us humans has changed radically from mainframes to personal computers. From a user's perspective, telephony has long had a more moderate development, but lately the advent of the mobile phone has had profound impact on many people's everyday life and communication. Now the next, third wave of ubiquitous computing is gradually coming upon us, with precursors in the form of embedded and mobile computing.

The mobile multimedia device is one particular result of technical convergence between the telephone and the personal computer. What do you get when you cross a computer with a telephone? Is it a computer you can make calls with? Is it a telephone you can play games with? Or does the resultant hybrid have emergent properties that cannot be traced directly to its parentage? These are examples of present and near-future issues for the user experience of mobile and ubiquitous multimedia, where the mobile multimedia device is in turn an important part of a larger system.

Interaction design is the most recent addition to the design disciplines, claiming ancestry partly from socially oriented computing disciplines, most notably Human-Computer Interaction, and partly from older, more firmly established design disciplines such as architecture, graphic design and industrial design. Interaction design is about design of the intended use of interactive, digital artifacts. As interaction designers we strive for positive innovation where people as well as technology are different from today.

However, history is rife with examples where people's actual use of an artifact goes counter to the intentions of the designer. History is likely to repeat itself in the future as computing is becoming ubiquitous and moves from the office into the everyday social, physical world, especially considering the flexible qualities of the interactive material. This calls for a renewed perspective on the roles of designers and the people formerly known as users. It will further affect interaction design process and practice.

As interaction designers in the mobile multimedia industry, we aim to take advantage of the results produced by the research community and apply it to our daily practice. Although important work is produced by researchers, several constraints inherent in a commercial context prevent much of its application in the daily work of the interaction design practitioner. To bridge the gap between theory and the design of consumer products, we increasingly enter into partnerships with selected research institutes and universities to produce immediately applicable results, recently in the areas of collaborative and inter-disciplinary design. Armed with new ways of working we will be better prepared for upcoming, possible futures.

Microsoft and Mobility – Today and in the Future

Jonas Persson

Jonas Persson, since May 2003, is Developer and Platform Director at Microsoft and a member of the Swedish Management Team. Jonas is responsible for the current and future development technologies in Sweden and handles relations with companies, Universities and Partners across the country. Jonas is also responsible for sales and marketing of development products /.NET in Sweden. Jonas has also been Sales Manager within the Server Area and Manager of Mobility at Microsoft EMEA. Jonas Persson started at Microsoft in 1997. Before stating at Microsoft Jonas completed his PhD in Chemistry at Uppsala University. Jonas is also Halo Champion within Microsoft.

Jonas likes to spend his spare time with his children. He also likes fishing, being in the outdoors and enjoying a glass of vintage wine from his private collection. Jonas also reads a lot, especially history.

Practical Considerations of Mobile Game Design

Ernest Adams

This keynote addresses some of the practical considerations surrounding mobile gaming from the perspective of the game designer. Mobile games have a number of advantages and limitations not found in other forms of video gaming. In addition to the usual issues surrounding the small form factor of the device, and the features offered by networking, the designer must address such things as the personal safety and security of the user, the circumstances in which the game may be played (on foot, in a vehicle, etc.), and the most appropriate means of generating revenue from a small game. While seemingly mundane, these and other factors influence the designer's creative choices, and it is better to understand them in advance than to create an innovative game which proves to be unplayable or unsalable for practical reasons. The keynote ends with a discussion of the features we may expect to see in future mobile devices and some ways they could affect game design.

SmartRotuaari – Context-aware Mobile Multimedia Services

T. Ojala[1], J. Korhonen[1], M. Aittola[1], M. Ollila[2,1], T. Koivumäki[3], J. Tähtinen[3] and H. Karjaluoto[3]

[1]MediaTeam Oulu, University of Oulu, Finland
[2]Norrköping Visualization and Interaction Studio, Linköping University, Sweden
[3]Department of Marketing, University of Oulu, Finland

Abstract

This paper presents the SmartRotuaari service system, which motivated by business and customer surveys provides a diverse set of consumer applications ranging from rapid and highly personalized mobile direct marketing to information, communication and payment services. We present results from the first field trial to show that SmartRotuaari provides a functional framework for large-scale field trials for the purpose of empirical evaluation of technology, new services, customer behavior and business models in real end user environment. Our work is based on a seamless cooperation between the various players in the R&D and business networks. To stimulate this cooperation we also extend our work to management research of value creating networks.

1 Introduction

Technological advances and improving financial viability of wireless broadband networks, middleware components and powerful versatile mobile devices are bringing a new dimension of mobile multimedia into the application domain. Whereas the conventional applications mostly assume stationary or fixed users, more and more emphasis is placed on mobility, the need for people to stay connected while moving around. One application domain, which is expected to greatly benefit from mobile multimedia, is mobile commerce or m-commerce for short.

In this paper we present the SmartRotuaari service system, which is operational at the city center of Oulu, the well-known "silicon valley" in Northern Finland (Rotuaari is the name of the walking area at downtown, hence the name of the service system). SmartRotuaari comprises of a wireless multiaccess network, a middleware for service provisioning, a web portal with content provider interface (CPI) and a collection of functional context-aware mobile multimedia services: service directory, map-based guidance, mobile ads, personal communication and presence, personalized news, mobile payment and Time Machine Oulu, a dynamic interactive 3D model of historical Oulu. The design of the business driven services is based on our own extensive survey of Oulu-based companies, while the design of the user driven services is motivated by existing surveys of location-based services.

The novel contribution of this work is a range of functional context-aware mobile multimedia services available in real end-user environment and their evaluation with a large number of test users. To our best knowledge we are unaware of another service environment, neither commercial nor academic, which would provide equally comprehensive set of consumer applications for businesses and mobile users. The most similar efforts include the eStreet project in Luleå, Sweden, and the Elisa mobilemall in Helsinki, Finland, which are briefly introduced below. For brevity we omit discussion on less similar installations, which for example provide location-based mobile multimedia information services, e.g. Lancaster GUIDE [8], Genoa Aquarium [1], and San Francisco Exploratorium [14].

Although not documented in the academic community, the eStreet project was one of the most successful pioneers of context-aware m-commerce services [12]. The eStreet project started off with simple SMS-based personalized direct marketing, which then evolved into more advanced services including cell id -based positioning, as well. The project had up to 2500 registered test users. One of the service providers in the project was the local McDonald's, which became the most successful store of the chain in Sweden, after 25% of the test users responded to their advertisement. The eStreet contributed to the foundation of a larger pilot framework known as Testplats Botnia [21].

The Elisa mobilemall located at the Arabia shopping center in Helsinki was introduced in September 2002 [11]. It includes WLAN coverage throughout the three-story shopping center, dynamic positioning of the user, map-based guidance to stores and products, a pool of PDA's test users are welcome to borrow and multimedia content provided in form of web pages.

This paper is organized as follows. Section 2 discusses the three-folded motivation of our work: SmartRotuaari service system, the evaluation of technology, consumer behaviour and services in large-scale field trials, and the underlying value creating R&D and business networks. Section 3 describes the SmartRotuaari service system in detail. Selected results from the first field trial are presented in Section 4. The early results of modelling the formation of the SmartRotuaari R&D network are discussed in Section 5. Section 6 concludes the paper.

2 Motivation

The inability of the mobile service providers to offer consumers sufficient added value has been among the main reasons for the slow adoption of mobile data services and m-commerce. To figure out what consumers really consider as added value with respect to mobile services is a challenging problem. As suggested by Woodruff [27], consumers' values are inherent and cannot be fully determined by the seller or producer. What is considered valuable also changes from one individual to another. This point is also stressed by Parasumaran [20], who states that values should be studied in different segments separately.

What adds to the difficulty of finding out what the really valuable mobile services and offerings are in different consumer groups is the fact that for the majority of consumers mobile Internet services are new and hence they do not really know their needs. As noted by Bagozzi and Dholakia [3], consumers' values are related to their goals, objectives and decision-making and thereby the evaluation of new services and offerings is difficult.

The theoretical base of the questionnaire used in the empirical study is in the two most widely applied theories which have been used to explain the adoption of ICT technologies, technology acceptance model (TAM) proposed by Davis [6] and the theory of planned behavior (TPB) by Ajzen [2]. Technology acceptance model predicts and explains the likelihood of technology use by stating that perceived usefulness and perceived ease of use are the primary determinants in technology acceptance behavior. It is a general level model which is capable of explaining user behavior across a broad range of end-user computing technologies and user populations. The key feature of TAM is to provide a basis for tracing the impact of external factors on internal beliefs, attitudes and intentions [7].

In the TPB, behavior is determined as a function of compatible intentions the perceived behavioral control. When studying consumer behavior with respect to new technologies such as mobile devices, the assumption that an individual is not in complete control over his/her behavior is very realistic, and therefore any possible effects of impaired control should be taken under study. In TPB the perceived behavioral control is expected to moderate the effect of intention on behavior. The intention is further determined by attitudes and subjective norms. Attitude toward a certain behavior is the degree to which performance of the behavior is positively or negatively valued, and it is determined by behavioral beliefs. Subjective norm is the perceived social pressure from e.g. friends or family members to engage or not to engage in a behavior.

We further make the use of the model of internet consumer satisfaction proposed by Matthew K. O. Lee in [24]. It provides an extensive framework for studying consumers' behavior in web environment. One of the main implications of Lee's model is that customer retention (which in the case under study could be interpreted as continuing usage of the mobile device) is determined by customer satisfaction. Customer satisfaction is in turn determined by logistic support, i.e. the manner in which the shipments of the purchased goods are handled, customer support, pricing of the goods, trust in the online vendor and different web shop related properties. These properties include different privacy and security related issues such as transaction safety, the quality of information, and the operational and navigational properties of the store such as speed of operation, system reliability and the ease of use. Although the model is developed primarily to explain consumer behavior in web environment, several features such as navigational properties, the relevance and the timeliness of the content and the offerings and trust in the service provider, can also be assumed as significant variables in explaining behavior in mobile environment.

SmartRotuaari is motivated by both the needs of companies (i.e. mobile service providers, technology providers, retailers etc) and consumers as end users of mobile services. The former have been identified via a tight collaboration with the local business actors at downtown Oulu. The motivation is straightforward: the companies have the most comprehensive knowledge about their business, in other words the "market pull". However, some of the companies, especially smaller retailers, are not necessarily aware of the possibilities offered by the new technology, in other words the fabled "technology push".

To strive for fruitful encounter of the above mentioned technology push and market pull, the project extends from technology research to management research. The interests of the business community are a vital issue in the commercial success of the innovative mobile services. To identify and satisfy these needs we work towards understanding and developing underlying value creating networks. By value creating networks we refer to both R&D networks that produce innovations and the business networks that commercialize the innovation, i.e. produce and market the innovation to end users (both consumers and other companies). In this scope we focus both on R&D activities in networks (e.g. [22]) and strategic business networks (e.g. [18]).

Our goal is three-fold: to model the dynamics of value creating networks (i.e. different stages of the process), to distinguish the

factors affecting the dynamics of value creating networks, and to model how the different factors influence the stages of the network development. The goal will be pursued via a qualitative, follow-up study, which will follow the developments of both the R&D network and the business network. Based on this real life data, the study combines theoretical knowledge into a description of the dynamics of value creating networks. Previous research has mostly concentrated in studying existing networks (see e.g. [9][16]) and thus this research fills the gap in studying networks longitudinally, producing real-time data.

The management research component has been initiated in form of a survey of business needs, joint workshops and one-to-one discussions. The survey was conducted at downtown Oulu in fall 2002. One of the objectives of the survey was to quantify the local firms' capabilities and skills in using computers, the Internet and the WWW. Further, we wanted to identify the tasks that the businesses found relevant in their daily customer service.

The questionnaire was sent to 209 companies in three business segments, namely retailers, restaurants and cafes, and other service providers. We received completed questionnaires from 57 companies resulting in a response rate of 27%. 80% of the firms were small ones, having a maximum of 15 employees. Additional statistics of the respondents: 80% uses computers daily, 75% have an Internet access, 65% uses Internet daily, 60% does not regard themselves as skilful computer users, 80% advertises their products or services, 80% of those who advertise use at least sometimes external help in implementing the advertising. From the survey, we can conclude that even though most of the respondents use computers daily, they do not regard themselves as computer wizards. This underlines the fact that any content provider tools offered to the businesses have to be as simple and easy to use as possible.

For the purpose of quantifying the end users' needs, SmartRotuaari and its services are exposed to test users in field trials. The SmartRotuaari service system provides a functional framework for large-scale field trials, which have several goals: evaluation of technology and services in real end-user environment with real users, evaluation of service usability and end-user experience, in-depth analysis of customer behavior in electronic service environment, and evaluation of candidate business models underlying this type of service system.

Data on user experience is collected in different forms, including questionnaires, interviews, monitoring and logging. The SmartRotuaari service system automatically logs data upon usage, for example locations and routes of the test users, service events and CPI interactions. Approval of the logging of data is one of the conditions test users and service providers have to comply with. All information is time stamped and stored into a database for further analysis. Having this data available, we can by means of data mining, for example, search for behavioral patterns such as typical walking routes at the downtown and how receiving a mobile ad may affect them. Having identified these kinds of routines we can then utilize them in service provisioning.

3 SmartRotuaari service system

SmartRotuaari comprises of a wireless multiaccess network, SmartWare architecture for service provisioning, a web portal with content provider interface and SmartServices, a collection of functional context-aware mobile multimedia services.

3.1 Multiaccess wireless connectivity

Wireless connectivity is currently provided in form of the Rotuaari WLAN (IEEE 802.11b) and the Octopus GPRS network. We are ourselves building the Rotuaari WLAN, which at the time of conducting the first field trial comprised of the 11 access points illustrated in Fig. 1. By November 2003 the Rotuaari WLAN had expanded to 20 access points covering the downtown and the market area. The Octopus GPRS is the network component of Octopus, the open innovation, development and testing environment for mobile applications in Oulu [18]. Up to 3000 mobile devices can be simultaneously connected to the Octopus platform. During year 2003 the Octopus environment has been upgraded to provide EDGE coverage at selected locations. Third network component will be provided by Bluetooth beacons at selected locations.

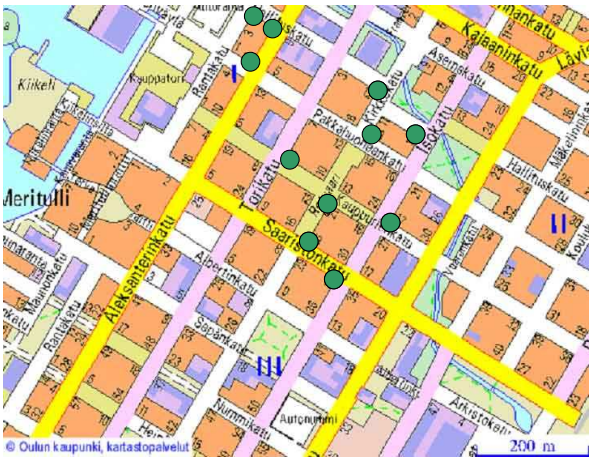


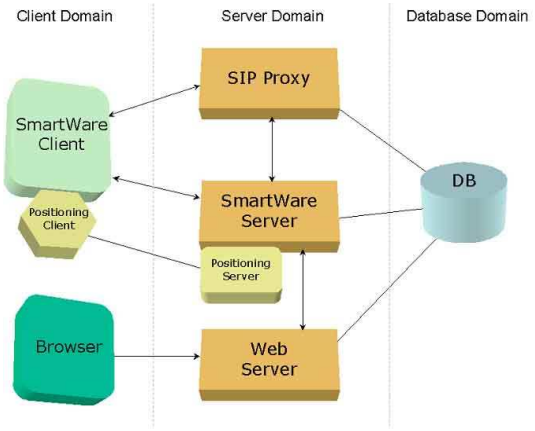
Figure 1. Locations of the 11 access points in the Rotuaari WLAN.

3.2 SmartWare architecture

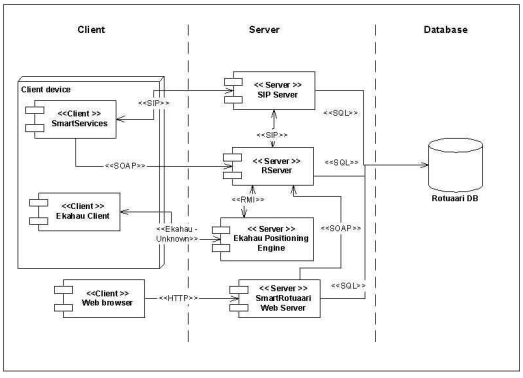
The services are provisioned with the SmartWare architecture illustrated in Fig. 2. The main components are:

- **SmartServices Client**; PersonalJava 1.2 compliant Java Application. The client is modular and extendible so that the existing services are abstracted from each other to make addition of new services as straightforward as possible. The client is designed to be used with a pen or a similar pointing device in a PDA, instead of the conventional mouse driven desktop approaches. Client communicates with the Rotuaari Server using SOAP over HTTP over TCP, and with the SIP Server using SIP over TCP.

- **SmartWare Server**; Offers SOAP interface to the clients for accessing the multimedia content of the services. It also controls the IP Gateway, so that registered users can access web content from the public Internet. The server includes SIP Server with SIP Registrar, SIP Proxy and SIP Presence server components, which form the infrastructure for instant messaging and real-time presence management. Presence notifications are used to convey all context information related to users and services. For example, when a new mobile ad is triggered, those users, which shall receive the ad, have their profile changed. This event triggers a presence notification, which is delivered to every interested party.



(a)



(b)

Figure 2. SmartWare architecture: (a) main components; (b) communication protocols.

- **Positioning Server**; Ekahau Positioning Engine [10], which conducts real-time WLAN-positioning based on a "location fingerprint" recorded beforehand and WLAN signal strength measurements obtained from the Positioning Client. The accuracy is 5-10 meters, when signals from three or more access points are available.
- **Database**; Relational database (MySQL) for storing all system data such as user information, ads, places and logs of users' and content providers' actions.
- **Web Server (http://www.rotuaari.net)**; Facilitates web based access and control of the service system. It provides interface for the general public to sign up as test users and manage their personal profile, CPI for service providers and administrative tools for managing user and service provider accounts.

One of the design principles has been to use open standards, which include:

- **HTTP** (Hyper-Text Transfer Protocol) for communicating with the Web Server.
- **SIP** (Session Initiation Protocol) for instant messaging and distribution of presence information.
- **SOAP** (Simple Object Access Protocol) for client-server communication.
- **SQL** (Structured Query Language) for database access.
- **XML** (Extensible Markup Language) for storing information such as user data, places and mobile ads.
- **WGS 84** (World Geodetic System 1984) coordinate system for describing locations of users, places, etc.

The SmartWare architecture facilitates utilization of following context information in service provisioning:

- **Time.** Time range, which indicates when the service is active.
- **Location**. Absolute and relative locations of the user and/or a service provider. For example, a service can be triggered if the user is within a specific distance of a service provider. Location of the user can be provided by GPS, WLAN-positioning or manual entry.
- **Weather**. Real-time weather observation is obtained from a local weather station, which is accessible via Internet. Two attributes of the observation can be currently used, temperature and wind speed. They are categorized to a set of non-overlapping labelled ranges, six for temperature (freezing → heat) and seven for wind speed (still → hurricane).
- **User profile and presence status**. Various detailed user information (age, marital status, education, occupation, income) is entered upon registration, together with any of the 23 different personal interest categories selected by the user. Also, the user's presence status, availability, can be utilized. Further, the user can also set his/her mood to any of the seven predefined values (hungry and/or thirsty, partying, looking for company, etc.) when logged into the system.

3.3 SmartServices prototype services

The service system includes a collection of functional prototype services: service directory, map-based guidance, mobile ads, personal communications, Time Machine Oulu, personalized news and mobile payment. The services are provided via a Java-based client software, which has been developed for the Pocket PC environment. Samples of the graphical user interfaces of the various services are illustrated in Fig. 3. The SmartServices "desktop" shown in Fig. 3(a) provides access to individual services.

Service directory provides access to a database of places, which can refer to a shop, public office, bus stop etc. The user can search places via free form text queries or pre-defined categories, as illustrated in Fig. 3(b). Once a particular place has been found, the user can ask for map-based guidance, for example. The user can also add the place to his "MyPlaces" collection of important places, which can later be accessed quickly. New places can be easily added to the database via the CPI.

Map-based guidance provides visualization of the location of a place, also relative to the user's location. A place can refer to any entry in the service directory, a 'buddy', or a location of personal interest specified earlier by the user. The service system contains maps of the whole area of city of Oulu in four different resolutions. Currently, the maps are stored and presented as bitmaps, which have been clipped beforehand. The user can scroll the map on the screen in all four principal directions with the pointer. Fig. 3(c) illustrates map-based guidance, showing the locations of the user (red dot) and a shop.

Mobile ads is a specially designed service for personalized customer service and customer relationship management, which were identified by the Oulu downtown businesses as the most important aspects of their customer service. A mobile ad is a SMIL 2.0 compliant multimedia message, which is authored and activated by a service provider and received by a customer, assuming that the conditions set for the mobile ad to appear (trigger) are met. The various conditions that can be used for triggering an ad are described in Section 3.2. Mobile ads as the one illustrated in Fig. 3(d) are created with the easy-to-use CPI, which facilitates authoring and activation of a new ad in a couple of minutes. Effectively, from a service provider's point of view mobile ads provide means for conducting rapidly highly personalized, low-cost direct marketing.

Personal communications is supported in form of peer-to-peer and group chat, which in the current version are implemented as simple text-based chat as illustrated in Fig. 3(e). The user can maintain a list of 'buddies' and invite them to a chat. Further, the user can set his presence status ('mood') to any of the predefined alternatives. The user can choose whether his location and/or presence status are shown to his 'buddies' and other users.

Time Machine Oulu [21] provides opportunity to travel both in time and spatial space simultaneously. The service builds dynamically from a database an interactive 3-D VRML2 virtual model of historical Oulu at the user's current location in the designated year as illustrated in Fig. 3(f). The database contains exact scientifically validated data of buildings in historical Oulu. The current database covers years 1822-1882 and is being extended towards the 21st century. The application allows for moving around in the virtual model using different viewpoints and accessing the buildings for additional information, e.g. the owner of a particular building at a given time. The service is realized using Cortona's browser, which supports VRML2.

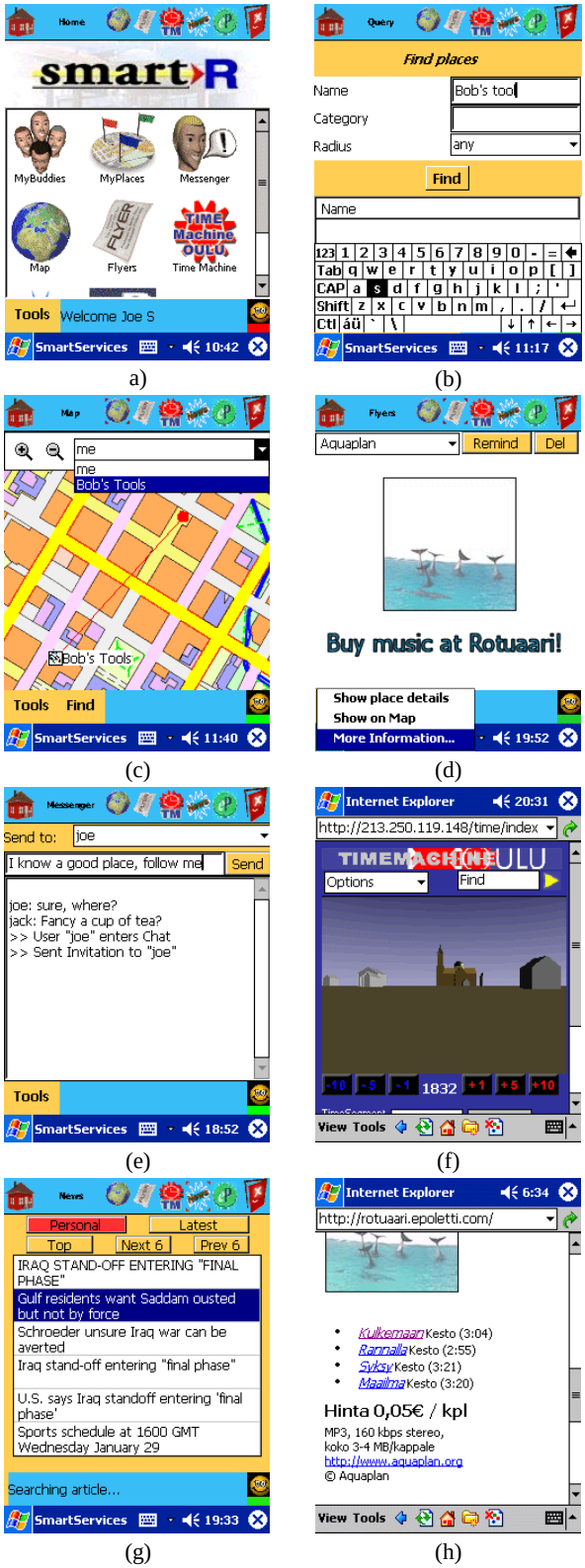


Figure 3. Samples of the graphical user interfaces of the prototype services: (a) "desktop"; (b) textual query into the service directory; (c) map-based guidance; (d) mobile ad; (e) personal communication; (f) Time Machine Oulu; (g) personalized news; (h) mobile payment.

Personalized news provides personalized access to a news feed as illustrated in Fig. 3(g). By using the '+' and '-' buttons the user can designate whether he finds the current article interesting or not. Personalization is carried out with the Leiki Targeting personalization engine [17], which based on the user's feedback updates the user's profile and ranks the incoming news so that those matching the user's profile are provided first.

Mobile payment facilitates micropayment for on-line content with real money. This is an important detail, for any service pilots providing chargeable services for artificial money are bound to produce unreliable results on customer behavior. The mobile payment is realized with the ePOLETTI service, which monitors the HTTP traffic and upon recognizing a link to a chargeable content redirects the traffic to the payment server [12].

4 Empirical evalution: Field trial #1

In this section we present the setup and results of the first field trial, which was conducted between 29 August and 30 September 2003. We have previously conducted smaller scale user evaluations of the SmartLibrary, the location-aware library service realized on top of the SmartWare architecture [1], and the TimeMachine Oulu service [21].

4.1 Setup

The field trial comprised of three major functions: deployment and maintenance of the service system, establishment of an office to coordinate the field trial, and collaboration with companies to obtain real-life up-to-date content in form of mobile ads.

Deployment of the service system was initiated by choosing the services to be evaluated: service directory, map-based guidance, mobile ads and TimeMachine Oulu. The mobile ad service was configured to allow the service provider to use following optional context conditions in triggering and delivering a particular ad (we call the set of conditions defined by the service provider the ad profile): activity range, time, age (seven ranges: custom (m-n), children (0-12), teenagers (13-15), youth (16-19), adults (20-39), middle-aged (40-64), seniors (65-119)), mood (seven categories), personal interests (23 categories), temperature (six ranges) and wind speed (seven ranges).

The ranges and categories were mutually non-exclusive, i.e. the service provider was able to designate any combination of them. If all conditions set in the ad profile were fulfilled, together with the receiving range set by the user (i.e. the user was within the designated distance from the service provider sending the ad) and the user allowed the reception of ads, the ad was delivered to the user's device.

The various conditions in the ad profile facilitate precise definition of the target group. However, the larger number of conditions is set in the ad profile, the sparser becomes the parameter space, when the numbers of ads and users are fixed. Since we had a limited number of active ads available in the service system, we removed personal interest categories from the ad profile at the halfway point of the trial, to guarantee that each test user received ads during his/her test period.

We allocated a pool of HP iPAQ's with expansion packs and WLAN cards to be loaned to the test users. We configured the iPAQ's shortcut buttons to directly open SmartServices client and an Internet browser. We also attached small label stickers to the buttons to increase accessibility. The maximum loan period was set to two hours, though longer periods were granted upon request.

(a)

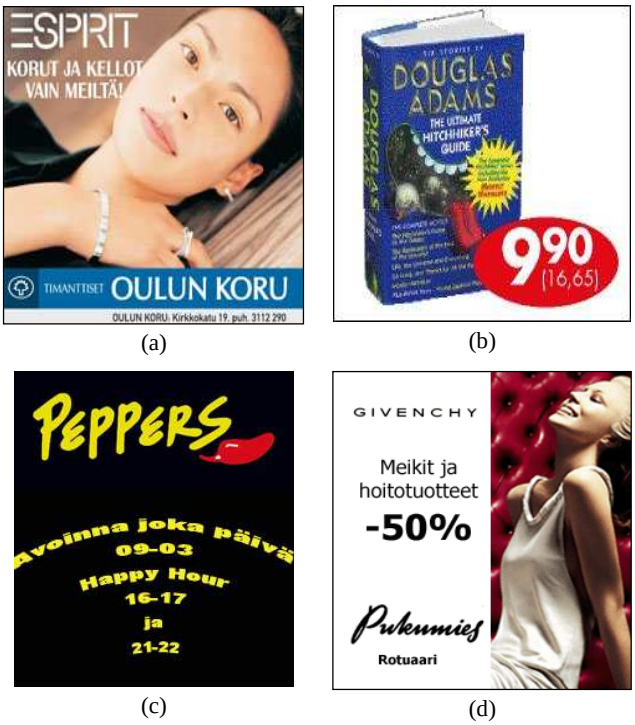
(b)

Figure 4. (a) Field trial office at the Rotuaari pedestrian street; b) a test user is signing up.

The field trial was coordinated from an office established in a small hut placed at the very heart of the pedestrian street Rotuaari (Fig. 4). The office was open for six hours daily from Monday to Saturday. The office was staffed with at least two researchers, who persuaded passers-by to sign up as test users, helped test users in creating a user profile and in using the iPAQ and services, and collected feedback via a questionnaire and occasional interviews. Each test

user was awarded with a voucher to a nearby café after the test session.

We recruited 18 local companies to serve as content providers, to produce for free mobile ads promoting their products and services. It was an encouraging experience that none of the companies we approached rejected the invitation. The companies were presented with the possibility to either produce the ads themselves using the CPI at the web portal or order the ads with a paper form from the 'advertising agency' offered by the project. To support the companies in their content creation we offered them few ad templates, which were produced by a professional advertising agency, one of the partners of the project.



(a) (b) (c) (d)

Figure 5. Sample mobile ads: (a) jewelry store advertising a jewelry brand; (b) discount offer from a bookstore; (c) bar advertising its happy hour; (d) cosmetics discount ad.

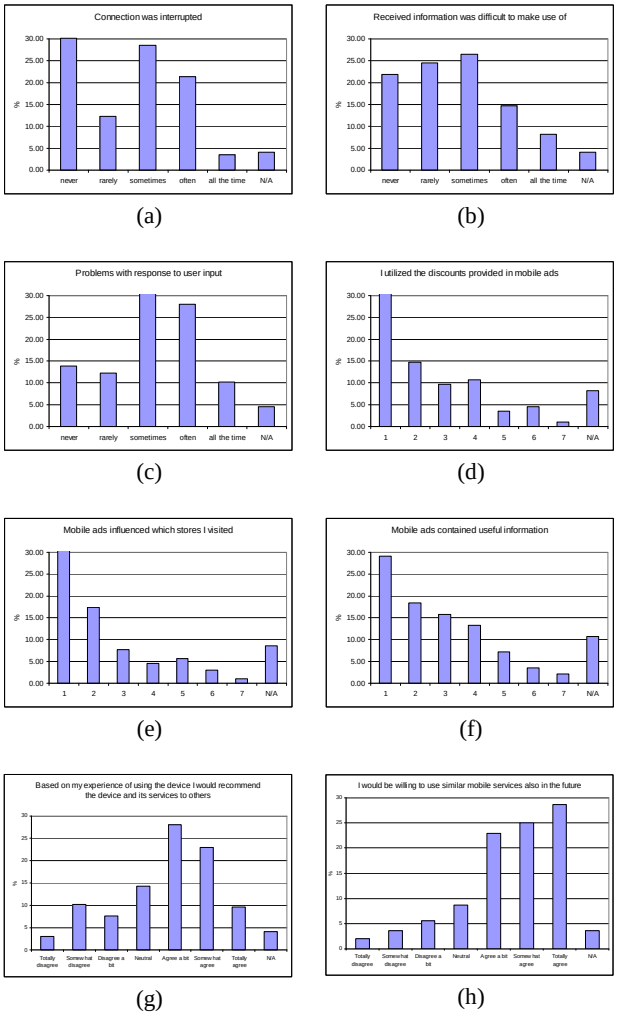
The final outcome was that five companies produced the ads themselves, seven companies relied on our 'advertising agency' and six companies did not produce any ads during the field trial after all. Their inactivity was compensated by few companies, which had several places of businesses for which they produced ads. Fig. 5 illustrates some of the mobile ads created by the companies.

The main restriction of the field trial was the limited coverage of the Rotuaari WLAN (see Fig. 1) and lack of indoor coverage. This resulted in test users losing the network connection upon leaving the coverage or entering stores, which in turn spelled usability problems.

4.2 Results based on questionnaires

We first extract some statistics of the questionnaires returned by the test users. It should be noted that not all test users filled in the questionnaire after returning the device. The questionnaire was very extensive, totalling 219 individual questions grouped into 31 categories. The questions provide comprehensive data both in terms of the background of the user and the experience of the mobile device and services evaluated in the field trial. The questionnaire was probably too extensive, as several test users complained about it being too long and requiring too much time.

There were 196 respondents in total, of which 66% were male and 34% female. The age distribution of the respondents was: under 18 3.1%, 18-24 38%, 25-34 41%, 35-49 14%, 50-65 3.1% and over 65 1.0%. 43% of the respondents were students, 17% executives and 16% workers. 86% of the respondents lived in the Oulu region, 11% elsewhere in Finland and 2.1% abroad. The reason for their visit to downtown was: running errands 33%, shopping 30%, work 10%, leisure 9.4%, tourism 2.6%, other 15%.



(a) (b) (c) (d) (e) (f) (g) (h)

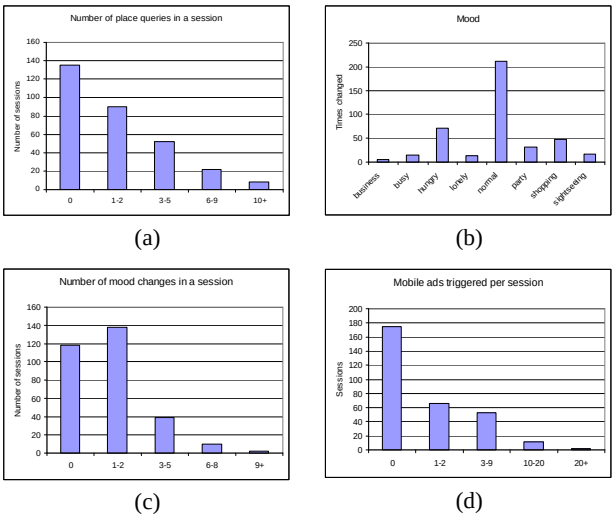
Figure 6. Some statistics compiled from the questionnaires.

The charts in Fig. 6 illustrate few selected statistics compiled from the questionnaires (presenting all interesting charts would require dozens of pages). Charts in Fig. 6(a-c) address the technical functionality of the device and the services with respect to connection stability, information usefulness and response. Charts in Fig. 6(d-e) reflect the test users' experience of mobile ads, when asked to provide their feedback on scale 1 (totally disagree) – 7 (totally agree).

The graphs support our own observation of the limited added value provided by the bulk of the mobile ads, which were content to just provide the name and street address of a store, for example. This emphasizes the importance of training the companies to use this type of a new marketing tool in a manner, which provides some benefit to the recipient. Despite the occasional technical problems and limited informative value of mobile ads the overall user acceptance of the device and services is encouraging, as illustrated in Fig. 6(g-h).

4.3 Results based on log data

Charts in Fig. 7(a-c) illustrate sample statistics compiled from the log data automatically recorded by the service system. In total there were 307 sessions, which was defined to be constituted if a user login was followed by at least one service event (location change, place query, triggered mobile ad, etc.) within 20 minutes of the login.

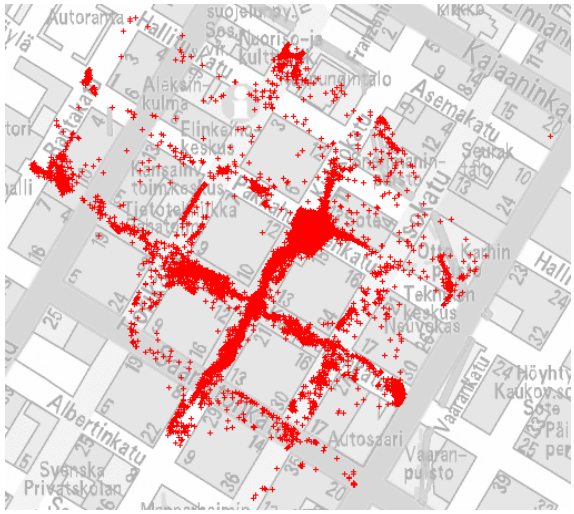


(a) (b)

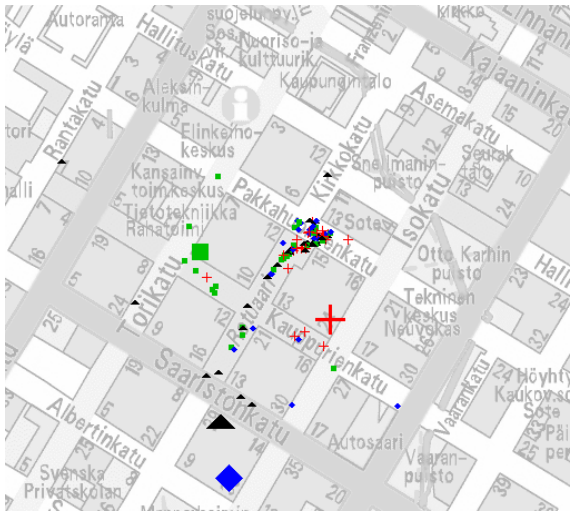
(c) (d)

Figure 7. Some statistics compiled from the log data.

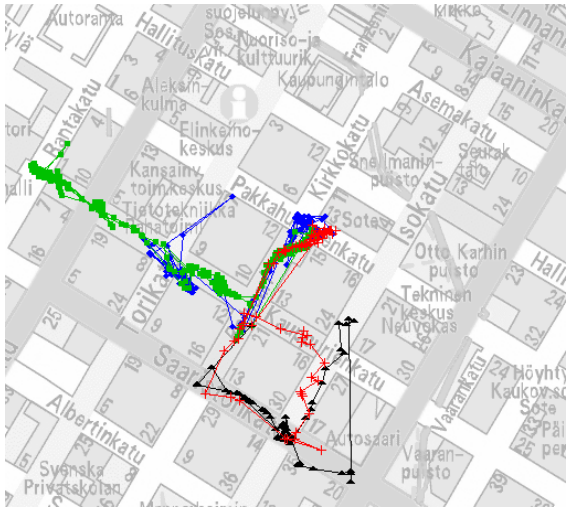
Fig. 7(a) reveals that almost half of the sessions did not include a single place query, while the other half of the sessions accumulated 589 place queries in total. Fig. 7(b) shows how many times a particular mood was selected: predictably normal mood was the most common among the 413 mood changes. Fig. 7(c) illustrates that over 60% of the sessions had at least one mood change. Fig. 7(d) shows that over half of the sessions did not include the reception of a single mobile ad.



(a)



(b)



(c)

Figure 8. Illustrations of location recordings in the log data and location-based service events.

Fig. 8(a) shows all location recordings found in the log data, which correspond to the limited coverage of the Rotuaari WLAN, excluding few erroneous location estimates produced by the Positioning Server. Fig. 8(b) shows the locations where four different mobile ads were received. The shape and color of the tiny markers designate the company sending the ad which in turn are marked with large triangle, square, circle and cross symbols. As expected, most ads are received in the neighborhood of the field office, where test users started their session. Fig. 8(c) displays the routes of four different test users. The long "leaps" on a route may be due to a lost connection or the user turning the PDA on and off.

5 R&D network of SmartRotuaari

In this section we will very briefly describe the first results of the research, which focus on modelling the first stage of the R&D network of SmartRotuaari, i.e. its formation (for a detailed description, see [15]). In [15], a process refers to a sequence of events or activities, which describe the development over time [24]. In addition, a process is considered through a teleological perspective (see [24], [25]), which applied in the focal case, views the network as a purposeful and adaptive entity, having a jointly preferred end state towards which it reaches. A process consists of a multiple streams of activities and may be limited by the network's recourses and environment. Moreover, the actions of the actors may also change the goal and thus the outcome of the process cannot be known in advance.

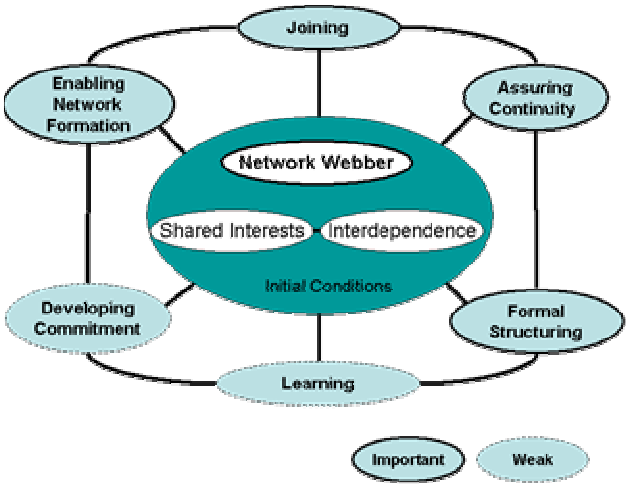


Figure 9. The process model of the engineered R&D network formation of SmartRotuaari [15].

As illustrated in Fig. 9, the process model of engineered R&D network formation of SmartRotuaari depicts two main elements; the initial conditions and the cycle of sub-processes. Third important element is part of the initial conditions, i.e. the role of the network webber. The initial conditions influence each other as well as the network formation. The cycle of sub-processes contains six intertwined series of activities, which together form the network formation process. Initial conditions influence each sub-process and each sub-process may also influence other sub-processes. The intensity of a certain sub-process as a part the network formation process can vary from low to high. The higher the intensity, the more important the sub-process is.

Based on the empirical data collected during the formation of the R&D network of SmartRotuaari, we are able to show that the sub-processes may take place simultaneously during a longer time period. Moreover, sometimes the formation process may even return to previous sub-process forming loops within the process. This view is very different from the existing research, which treats formation process as stages, following each other like steps in a ladder.

6 Discussion

This paper presented the current status of the SmartRotuaari service system, together with results from the first field trial. The results show that SmartRotuaari provides a functional research framework for prototyping and empirical evaluation of context-aware mobile multimedia services, customer behaviour and business models in real end user environment.

The main technical drawback in the current implementation of the SmartRotuaari service system is the monolithic Java client, which limits the selection of applicable mobile devices to those equipped with the required Java functionality (JVM), effectively laptops and high end PDA's. Addition of new services also requires modifications to the client software. Therefore, we are currently redesigning and implementing the client side according to the 'web services' paradigm so that services are provided via a web browser. This allows using the services also with the new smartphones, which are equipped with XHTML browsers and TCP/IP stacks. More importantly, the new architecture will facilitate seamless expansion of the service system with independent third party services. Further, we will employ services developed by our industrial partners.

In the scope of empirical evaluation, in the upcoming field trials a particular exercise we look forward carrying out is correlating the qualitative data obtained from the test users with the quantitative log data recorded automatically by the service system. The motivation is to see to which extent the test users' reporting ("I used the news service most") corresponds to the log data ("Downloading of music videos was the most often used service").

Acknowledgements

The financial support of the National Technology Agency of Finland, the GETA Graduate School in Electronics, Telecommunications and Automation, the Infotech Oulu Graduate School and the Academy of Finland is gratefully acknowledged. The authors wish to thank the numerous organizations and individuals, whose invaluable collaboration has made this work possible.

References

[1] Aittola M, Ryhänen T & Ojala T (2003) SmartLibrary - Location-aware mobile library service. Proc. Fifth International Symposium on Human Computer Interaction with Mobile Devices and Services, Udine, Italy, 411-416.

[2] Ajzen I (1991) The theory of planned behavior. Organization Behavior and Human Decision Processes 50: 179-211.

[3] Bagozzi RP & Dholakia U (1999) Goal setting and goal striving in consumer behavior. Journal of Marketing 63:19-32.

[4] Bellotti F, Riccardo B, de Gloria A & Margarone M (2002) User testing a hypermedia tour guide. Pervasive Computing 1:33-41.

[5] Bornträger C, Cheverst K, Davies N, Dix A, Friday A & Seitz J (2003) Experiments with multi-modal interfaces in a context-aware city guide. Proc. Fifth International Symposium on Human Computer Interaction with Mobile Devices and Services, Udine, Italy, 116-130.

[6] Davis FD (1989) Perceived usefulness, perceived ease of use, and user acceptance of information technology. MIS Quarterly 13:319-340.

[7] Davis FD, Bagozzi RP & Warshaw PR (1989) User acceptance of computer technology: a comparison of two theoretical models. Management Science 35:982-1002.

[8] Davies N, Cheverst K, Mitchell K & Efrat A (2001) Using and determining location in a context-sensitive tour guide. IEEE Computer 34(8):35-41.

[9] Doz Y, Olk P & Ring P (2000) Formation processes of R&D consortia: which path to take? Where does it lead? Strategic Management Journal 21:239-266.

[10] Ekahau Positioning Engine (2003) http://www.ekahau.com/products/positioningengine/.

[11] Elisa mobilemall (2003) http://www.efodi.com/mobilemall/ineng.html.

[12] ePOLETTI (2003) http://www.epoletti.com.

[13] eStreet (2003) http://www.estreet.lu/.

[14] Fleck M, Frid M, Kindberg T, O'Brien-Strain E, Rajani R, Spasojevic M (2002) From informing to remembering: Ubiquitous systems in interactive museums. Pervasive Computing 1:13-21.

[15] Heikkinen M & Tähtinen J (2003) The formation processes of an R&D network: a case study. Third International Conference on Electronic Business, Singapore, to appear.

[16] Hellström T & Jacob M (1999) Evaluating and managing the performance of university-industry partnership: from central rule to dynamic research networks. Evaluation 5(3):330-339

[17] Leiki Targeting (2003) http://www.leiki.fi/products.leiki.

[18] Möller K, Rajala A & Svahn S (2002) Strategic business nets – their types and management. Journal of Business Research 55.

[19] Octopus (2003) http://www.mobileforum.org/octopus.

[20] Parasuraman A (1997) Reflections on gaining competitive advantage through customer value. Journal of the Academy of Marketing Science 25(2):154-162.

[21] Peltonen J, Ollila M & Ojala T (2003) TimeMachine Oulu - Dynamic creation of cultural-spatio-temporal models as a mobile service. Proc. Fifth International Symposium on Human Computer Interaction with Mobile Devices and Services, Udine, Italy, 342-346.

[22] Peters L, Groenewegen P & Fiebelkorn N (1998) A comparison of networks between industry and public sector research in material technology and biotechnology. Research Policy 27:255-271.

[23] Testplats Botnia (2003) http://www.testplatsbotnia.com.

[24] Turban E, Lee JK, King D & Chung MH (2000) Electronic Commerce: A Managerial Perspective. Prentice all, Upper Saddle River.

[25] Van de Ven A (1992) Suggestions for studying Strategy process: a research note. Strategic Management Journal 13:169-188.

[26] Van de Ven A & Poole M (1995) Explaining development and change in organizations. Academy of Management Review 20(3):510-540.

[27] Woodruff RB (1997) Customer value: The next source for competitive advantage. Journal of the Academy of Marketing Science 25(2):139-154.

An Architecture for Distributed Spatial Configuration of Context Aware Applications

Martin Wagner Gudrun Klinker

Technische Universität München, Institut für Informatik
Boltzmannstr. 3, 85748 Garching bei München, Germany*

Abstract

This paper discusses an architecture for spatially distributed storage of contextual configuration information in ubiquitous computing environments. Based on the assumption that we want to integrate arbitrary mobile clients in ubiquitous computing environments, we derive the requirements for the spatial distribution of data, transparent access to context aware configuration data, and separation of context estimation algorithms. We developed a highly distributed architecture that fulfills these requirements. Using DWARF, we implemented and successfully demonstrated a mobile demonstration setup that incorporates all key concepts of our architecture.

CR Categories: H.5.1 [Information Interfaces and Presentation]: Multimedia Information Systems—Artificial, augmented, and virtual realities; C.2.4 [Computer-Communication Networks]: Distributed Systems—Distributed applications

Keywords: Augmented Reality, Context Awareness, Ubiquitous Computing, Distributed Data Storage

1 Introduction

This paper deals with an architecture for spatially distributed storage of contextual configuration information in ubiquitous computing environments. *Ubiquitous Computing* [Mattern 2001; Weiser 1991] aims at providing computers invisibly at all places one may go to such that access to information gets ubiquitous.

In particular, we explore the possibilities of Augmented Reality based applications in ubiquitous computing environments. *Augmented Reality* (AR) is a new technology that seamlessly integrates virtual information in a user's physical environment. Using a Head-Mounted Display (HMD), the user's view of the world is augmented with virtual objects that are spatially fixed in relation to the physical objects. For example, a refrigerator AR application would display virtual objects that depict the content of the refrigerator or objects that are missing and should be refilled on the display, thus giving the impression to the user that the door of the refrigerator is transparent. When the user moves or turns his head, the virtual objects are displayed so that the user has the impression that they are "attached" to the refrigerator. Hence, one of the core requirements of AR is the correct three-dimensional registration of the user's viewing direction and all relevant objects in real time.

Many AR applications handle registration by a combination of several trackers, each specialized for different situations (e.g., tracking the user outdoors, tracking the general location of the user indoors, precise tracking of the user's head motion) [Klinker et al. 2000]. All these devices need to be configured. *Configuration data* is data that a generic software or hardware device needs to work correctly in a ubiquitous computing environment. Hence, an optical tracker worn by a mobile user needs descriptions of markers in the current room to correctly compute the room's position relative to the user; a tracker mounted in a room needs information about the properties of a marker on the user to register the user's position relative to the room. Moreover, many ubiquitous computing applications have the requirement to accomodate arbitrary *mobile clients* potentially composed of multiple, configurable devices.

The usual approach of storing all configuration data in a central database turns the environment into a complex monolith that is hard to understand for both developers and users. A strictly central data structure is contradictory to the intention of ubiquitous computing to provide usually simple applications that are specifically tailored to the who, where, when and how. In this paper, we propose a highly distributed architecture that stores the configuration data where it belongs to. We think that every device in a ubiquitous computing environment should be responsible for its own configuration. This yields more flexible applications in context aware environments, as the environment can be partially reconfigured dynamically and with minimal side effects to the rest of the system. By this assumption, the architecture we propose naturally allows to incorporate arbitrary new mobile clients worn by users that come to ubiquitous computing environments they have never been to before.

Although this architecture results from the need to configure mobile tracking devices, it is suitable to store all types of configuration information that have a clear mapping on spatial entities. These entities may be rooms as well as mobile users carrying their own information with them on their *mobile clients*.

This paper is structured as follows. Section 2 illustrates the problem using a visionary scenario. Section 3 derives requirements for our architecture from this. Section 4 presents the core concepts of the DWARF framework that allow us to implement a distributed configuration. The key ideas of this architecture are described in section 5. We implemented the scenario with a mobile marker-based tracker and describe some details of this implementation in section 6. Section 7 discusses work related to ours. Section 8 concludes this paper with proposals for future enhancements.

2 An Example Scenario

The following scenario illustrates the typical features of a ubiquitous computing environment.

A user, Joe, walks around an intelligent building (see figure 1). The building is equipped with many sensors registering what happens inside. All sensors are integrated in a ubiquitous computing environment providing access to information and applications at

*e-mail: {wagnerm,klinker}@in.tum.de

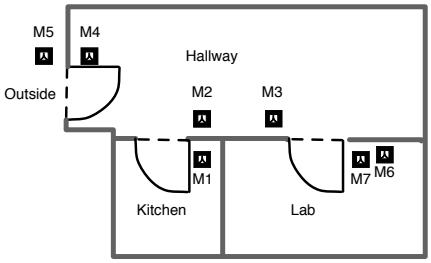


Figure 1: Example scenario.

every point in the building. Most users of the building wear mobile clients, a set of computing devices having widely varying functionality and configuration, depending on the user's personal preferences. Joe is wearing a mobile Augmented Reality system consisting of computers that access the environment using a Wireless Local Area Network (WLan), a head mounted display, an input device on his hand and a camera that detects markers for tracking Joe's current position. A prototype of this mobile client is depicted in figure 5.

Many applications are available in the building, and most are tied to specific rooms. For example, the refrigerator in the kitchen has a virtually transparent door, the table in the AR lab is augmented with shared views of three-dimensional construction plans Joe is currently working on. The AR system enables him to collaborate on these with some remote colleagues.The hallway provides short speech output about what is behind all the doors. To execute the matching applications at every point in time, the mobile client has to know which room Joe is currently in. This is handled by optical tracking software analyzing the video stream from Joe's camera. For example, the inner side of the kitchen door carries a marker with the semantics "User leaving the kitchen". We do not assume that the mobile client has any knowledge about the building, as we would like to incorporate arbitrary new users. As such, our environment provides the semantical information correlating with certain markers automatically to the mobile clients.

Let us assume Joe is in the hallway. His mobile client has a service that can play arbitrary sounds and gets therefore connected to the hallway's "What's behind the door?" application. Whenever either the environment or Joe's mobile client realize that he is standing in front of a door, the application gets notified and sends an audio stream to Joe's mobile client. If Joe now leaves the hallway and goes to the lab, his audio service gets disconnected from the hallway application and connected to the collaboration program in the lab. By this dynamic reconfiguration, Joe is now able to listen to his colleagues' comments on his work using the same mobile client services as in the hallway. If Joe had another service allowing streaming video in his mobile client, he could see his colleagues as well.

3 Requirements for Distributed Spatial Configuration

Ubiquitous computing environments are typically made out of many small applications using a small number of basic components and a large number of users and sensors. Both applications and sensors need to be partially reconfigured very often, as the environment serves many users with different background, preferences, access rights and capabilities of their mobile clients. The simple approach of using a central static configuration database for our setup has several major drawbacks. On the user level, the whole environment would have to be treated as a single complex application, making it hard for both developers and end users to understand the effects of changes in the configuration. This may easily break existing applications if the central system is adapted to new applications. Moreover, the centralized approach introduces a single point of failure.

Configuration information that only is of interest in certain spatial regions is organized best according to these regions rather than in a centralized structure. This facilitates keeping the data up to date and allows simple partial reconfiguration. In addition, it seems reasonable to let every user with a mobile client store all configuration information that the environment needs to correctly communicate with the user at this mobile client. This strategy allows users to dynamically change their clients, depending on personal preferences and the availability of hardware or software components, without the need to tell the ubiquitous environment of these changes before actually using the reconfigured client. Our architecture supports spatial distribution of configuration data and automatic reconfiguration of both the environment and the mobile clients.

This leads to the following requirements on a distributed configuration architecture.

Context aware configuration data. In our work, we follow Dey's [Dey and Abowd 2000] separation of context in *location, identity, activity* and *time*. Our architecture should support configuration based on all these contextual attributes. Location is obviously handled by the spatial distribution, as is partially the identity of the user. Some configuration may depend on the current time of day, for example, an optical tracker may have some parameters telling it the current lighting situation. Finally, the user's current activity, in our example the application he currently uses, may influence the choice of tracking accuracy.

Transparent access to configuration data. Components that need to be configured should not have to bother about where to get the necessary data. The architecture should support dynamic reconfiguration based on the current context. This should happen transparently to the components getting configured. The transparency may be used to centralize some logically separate configuration information on a single database server for security or reliability reasons.

Separate context estimation component. If the mobile client and the ubiquitous infrastructure have to work together to provide its users full access to all information available, they need to communicate using clear protocols. This holds especially true for those parts of the system that derive the current context out of sensor data. For example, an optical tracker has to get information about its environment to determine the user's current location. If the environment is equipped with new trackable features that can only be handled by an algorithm not available to the mobile client beforehands, we either have to implement this algorithm at the mobile client, dynamically load executable code to the mobile client or have to send the mobile client's video image to a stationary compute server. In all three cases, we must encapsulate the tracking algorithm in a component that takes video images and provides location information. The same holds true for all other components deriving context of the current situation.

4 DWARF

The *Distributed Wearable Augmented Reality Framework* [Bauer et al. 2002b] is a research platform that explores the possibilities of Augmented Reality based applications in distributed ubiq-

uitous computing environments. The framework contains services for tracking, visualization of three-dimensional data, calibration of objects, multimodal input, and modeling of user tasks. Several systems have been built so far [Bauer et al. 2001; Echtler et al. 2002; Klinker et al. 2002; MacWilliams et al. 2003], consisting of between 10 and 50 services.

The DWARF services are accessed through a CORBA-based middleware. On each network node of a DWARF system, there is one *service manager*. There is no overall central component. The service manager controls its local services and maintains descriptions of them. Each service manager cooperates with the others in the network to set up connections between services. A DWARF *service* is the basic building block of a running system. It either encapsulates a hardware device like a tracker, performs some reusable functionality like controlling a taskflow or handles some application-specific task. Each service is running within a separate operating system process or thread. Its functionality is described in terms of *abilities*, the functionality it requires from other services is described in terms of *needs*. Needs and abilities are matched using *connectors*.

An *ability* is the abstract description of a service's functionality, e.g. location data for optical trackers. A service can have multiple abilities, e.g. the tracker may track multiple objects simultaneously. Abilities are typed, an optical tracker delivers `PoseData` for location information.

A *need* describes the functionality a service requires from its counterparts to be able to work. Again, a service may have multiple needs. The optical tracker needs a stream of video data and descriptions of markers to be able to find these markers in the video image. Needs are typed, too, and a need can only be satisfied by an ability of the same type.

A *connector* is a description of the communication protocol, e.g. shared memory for video data, CORBA object references or CORBA notification events for event-based communication of location data.

Attributes enhance the description of an ability. The optical tracker may therefore specify which object it tracks, using e.g. `Thing=UserHead`. Needs can be refined using *predicates*, e.g. `(&(User=Joe)(Room=Lab))`. When matching needs with abilities, the service managers ensure that the ability's attributes satisfy the need's predicate. Attributes may be specified for the entire service, in this case all abilities of that service have these attributes.

```
<service name="OpticalTracker">
    <need name="video" type="VideoStream">
        <connector protocol="SharedMemory"/>
    </need>
    <need name="marker" type="MarkerData"
          predicate="(&(Room=*)(User=*))">
        <connector protocol="ObjectReference"/>
    </need>
    <ability name="poseData" type="PoseData"
             isTemplate="true">
        <attribute name="Room"
                   value="$(markerData.Room)">
        <attribute name="User"
                   value="$(markerData.User)">
        <connector protocol="NotificationPush"/>
    </ability>
</service>
```

Figure 2: Sample XML description of an optical tracker service having two needs of type *VideoStream* and *MarkerData* and an ability of type *PoseData*. If the need for *MarkerData* gets satisfied by another service with attributes *Room* and *User*, the optical tracker offers the *PoseData* ability with the same attributes.

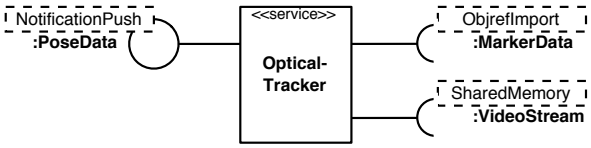


Figure 3: Sample UML description of an optical tracker service. Needs are depicted as semicircles, abilities as circles.

Each DWARF service installed on a system may either describe itself at startup to the service manager or use an XML file format to give the service manager the possibility to start and stop the service on demand. Our example of an optical tracker is shown in figure 2. A UML-based notation is shown in figure 3.

MacWilliams and Reicher describe in [MacWilliams and Reicher 2003] how a service's ability may change at runtime according to how its needs are satisfied. In our example, if and only if the optical tracker gets `MarkerData` for user *Joe* and the room *Lab*, it can provide an ability of type *PoseData* with the attributes `User=Joe` and `Room=Lab`. As such, the optical tracker has a *template* ability of type `PoseData` that can run in multiple instances depending on the available configurations.

5 Automatic Context Aware Configuration

The DWARF framework provides the basic building blocks for our distributed configuration architecture. In this section, we describe this architecture based on a simple example setup consisting of the following components. A UML description of the setup is shown in figure 4. This rather simple setup is prototypical for most ubiquitous computing applications.

Video Grabber. A digital camera is read out and the resulting video image is sent to other components.

Optical Tracker. This component takes a video image and marker descriptions detailing the marker's appearance as well as their 3D location in a room. It then finds the marker in the image and reconstructs the camera's current position [Tsai 1987]. In our setup, the user is wearing the camera, therefore the tracker yields the user's current position. Finally, this component has the need for contextual changes and can therefore be told the current context.

Application. There is one instance of this component for each application, such as the transparent refrigerator door discussed in section 2. Applications may either run on the user's mobile client or in the ubiquitous environment. A typical application takes the location information from the tracker and displays augmentations to the user's view using the mobile client's rendering component [MacWilliams et al. 2003].

Context Estimation. This component analyzes the location data from the tracker and triggers context switches whenever it concludes that a user has moved to a new room. Note that this component may exist in multiple instances, each specifically adopted to a certain contextual situation.

Configuration Data. This component actually stores configuration data. There is one instance of this component for each contextual state. As the marker descriptions of the tracker are nothing more than configuration information, it is this component that reconfigures the optical tracker.

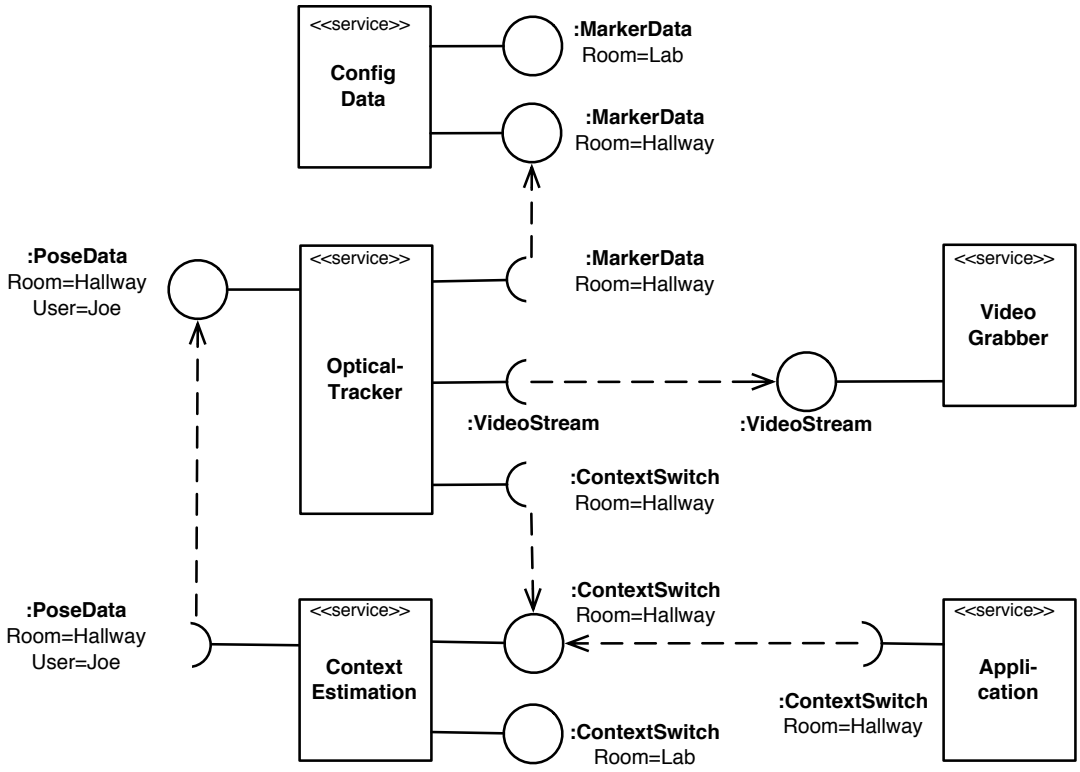


Figure 4: UML model of the mobile setup. Needs are depicted as semicircles, abilities as circles. Note that needs may satisfy two other service's abilities (*ContextSwitch*). Both the *ContextEstimation* and the *ConfigData* services offer abilities for two contextual situations.

Context is modeled using DWARF attributes and predicates. According to [Dey and Abowd 2000], we split context in *identity, location, activity* and *time*. We assume that *Joe* is moving around the *Lab* at *9 a.m.* doing *NothingSpecial*. As such, the optical tracker's need for marker descriptions should have the predicate `(Room=Lab)` and its ability to send location data should have the attributes `User=Joe`, `Room=Lab` and `Time=9am`. Our optical tracker does not care about Joe's current activity, so we leave out this attribute.

Let us now discuss how the automatic configuration proceeds. We assume the user starting in the *Hallway* and moving to the *Lab* (see floor plan in figure 1). The transition from the contextual state `Room=Hallway` to `Room=Lab` is triggered by the marker *M3*, therefore the optical tracker is configured initially to recognize marker *M3* (and some others). Once the user is in the lab, the optical tracker has to be configured in a way that allows it to recognize markers *M6* and *M7* in order to detect other context switches. The flow of events is as follows:

1. The user enters the building's hallway with his mobile setup. He starts the *Video Grabber*, *Application* and *Optical Tracker* components. This may also be performed automatically (see [MacWilliams and Reicher 2003]).

2. Initially, the optical tracker's need for marker data has the predicate `(Room=Hallway)`. As there exists a *Configuration Data* service with a matching attribute, the DWARF service manager connects both. The configuration service gives the information necessary to detect markers *M2*, *M3* and *M4* to the optical tracker.

3. The service manager recognizes that there exists a *Context Estimation* component that has the need for marker data with the predicate `(Room=Hallway)` and the ability to trigger contextual changes, attributed again by `Room=Hallway`. The component matches the optical tracker in its current state and gets therefore connected.

4. The user now moves towards the door and has marker *M3* for quite a while in his camera's viewing frustum. This information is sent to the context estimation that concludes that the user must have left the hallway and went to the lab. It therefore tells the optical tracker (and all other components of the mobile setup that may be interested) that there was a change in context.

5. The optical tracker service changes its service description at the service manager, setting the marker data need's predicate to `(Room=Lab)`.

6. The service manager disconnects the hallway's configuration and context estimation services from the optical tracker. In consequence, the optical tracker unloads all configuration data it received from these services. As there is a new configuration service available in the lab with an attribute `Room=Lab`, the optical tracker is connected with the new configuration and context estimation services.

7. Again, the optical tracker gets information about markers, this time *M6* and *M7*, and outputs location information to the lab's context estimation component until the latter concludes that the user left the room again.

Up to now, we always assumed that the user has an initial contextual state. This state is either provided by explicit user input, i.e. "I am at the TUM campus." or by using the same configuration mechanisms with special attributes.

For each network broadcast area, there must be one configuration and context estimation service that provides the initial state. For example, if a user walks towards a campus building and enters its WLan area, the current `Room` predicate should be set to `(Room=global)`. The campus building may now provide a context estimation service with matching attributes that interprets the user's current GPS information in order to trigger contextual switches at the user's mobile client, such as setting the `Room` attribute to `Room=FMIBuilding`.

6 Implementation Status

The architecture just described was implemented within a larger project, ARCHIE (Augmented Reality Collaborative Home Improvement Environment), evaluating some new AR technologies for collaborative support of architectural planning. Further information about ARCHIE can be found on our web site[1].

The system runs on several stationary and mobile computers running Linux (see figure 5). The components are implemented in C++ and Java. We use an iBot IEEE 1394 camera that delivers its data via the *libdc*[2] library. The optical tracker uses the AR Toolkit library[3], an easy to use marker-based tracking library. More information on how we incorporate AR Toolkit in our setup can be found in [Wagner 2003]. The application consists of a simple *Speaker* service that plays prerecorded texts matching the current location. Context estimation is rather simplistic, once the system realizes that a single or multiple markers have been "seen" several times by the user's camera, a context switch is triggered. As such, our system allows the user to implicitly change the context by moving to another room. The configuration data is stored by a service wrapping a MySQL database offering separate abilities for every configuration context stored. We successfully demonstrated the context aware configuration architecture incorporating multiple instances of the *Context Estimation* and *Configuration Data* services presented in this paper.

7 Related Work

There exist several systems supporting ubiquitous computing environments. The Gaia project [Hess and Campbell 2003; Román et al. 2002] introduced the concept of *Active Spaces*, ubiquitous computing environments similar to our spatial entities. It proposes a context aware filesystem that stores the user's data. In contrast, our system stores the user's data on the user's mobile client. Project Aura [Garlan et al. 2002] uses the coda file system [Satyanarayanan 2002] to allow a mobile user nomadic file access. Again, the data storage solution is central. The Ninja system [Gribble et al. 2001] uses distributed data structures to provide the high load that a central storage solution has to handle. Riché and Brebner propose a user-centric replication mechanism [Riché and Brebner 2003] to speed up access to contextual information, however, this approach is based on a centralized data storage as well. The NEXUS project [Hohl et al. 1999] aims at providing an open platform for context-aware applications with a special focus on location. The CANU subproject [Bauer et al. 2002a] proposes to obtain model data by the next network node in a mobile ad-hoc network and is therefore similar to our basic assumption of spatially organized data. However, the spatial model is still represented as a single central graph and does not support the separation of context estimation according to the current spatial situation.

The configuration problem of trackers with large tracking areas has been treated by Reitmayr and Schmalstieg using an XML based configuration framework called OpenTracker [Reitmayr and Schmalstieg 2001], later this work was extended to allow reuse of configuration information [Kalkusch et al. 2002]. Although using OpenTracker is an easy way of configuring setups of many tracking devices, the framework does not support dynamic reconfiguration of trackers. The Bat system [Addlesee et al. 2001] provides building wide location tracking, but suffers from a central data storage, not allowing dynamic addition of mobile clients.

The configuration data in our architecture is structured along the four dimensions of context defined by Dey [Dey and Abowd 2000]. Lieberman and Selker [Lieberman and Selker 2000] point out that context aware applications can simplify the interaction with computers without reducing functionality. This simplification can be done via automatic configuration as discussed in this paper. The GUIDE Project's data storage architecture [Efstratiou et al. 2001] discusses strategies to resolve ambiguities in contextual information that could be incorporated in our architecture, whereas Dearle et al. propose an architecture for global smart spaces [Dearle et al. 2003], leading to universally available information that may be enhanced locally by ubiquitous computing environments as discussed in this paper.

8 Conclusion and Future Work

In this paper, we presented an architecture for distributed spatial configuration of context aware applications that allows spatial distribution of data, separation of configuration data according to its contextual use, transparent access to the data and an encapsulation of algorithms estimating the context. This architecture is built on the DWARF framework, thus incorporating a wide array of already existing other components for location tracking, 3D rendering and modeling of data necessary for Augmented Reality.

Although we discussed our concepts only for a rather simple marker-based tracking setup, we think it is well suited for all mobile systems acting in ubiquitous environments. However, clear interfaces need to be defined to allow flexible yet simple configuration for other areas than tracking.

While the ideas presented in this paper allow the spatial separation of data, it is left to the application to define spatial entities. In our setups single rooms proved well, for other applications we might have to choose larger or smaller contextual entities. An interesting question is how to structure context in a way that allows a "natural" design of new applications. Problems arising include how to provide architectural support for restructuring legacy information and whether it is possible to let the system learn context boundaries automatically. A similar area of future work lies in developing concepts on how to actually store the spatially organized data. For security or reliability reasons, it might make sense to store all spatially organized configuration data of a building in a single database system, if the network infrastructure in this building is dense and reliable.

Up to now, we have not systematically investigated security and privacy issues, although storing all user data at the user's mobile client will serve as a good starting point for implementing privacy policies.

Finally, we might incorporate some of the storage facilities discussed in related work in order to allow efficient access not only to local but to global data as well. In our concept, global data is a context device that has no `Room` attribute and is therefore available

[1] `http://www.augmentedreality.de`

[2] `http://sourceforge.net/projects/libdc1394/`

[3] `http://www.hitl.washington.edu/research/shared_space/download/`

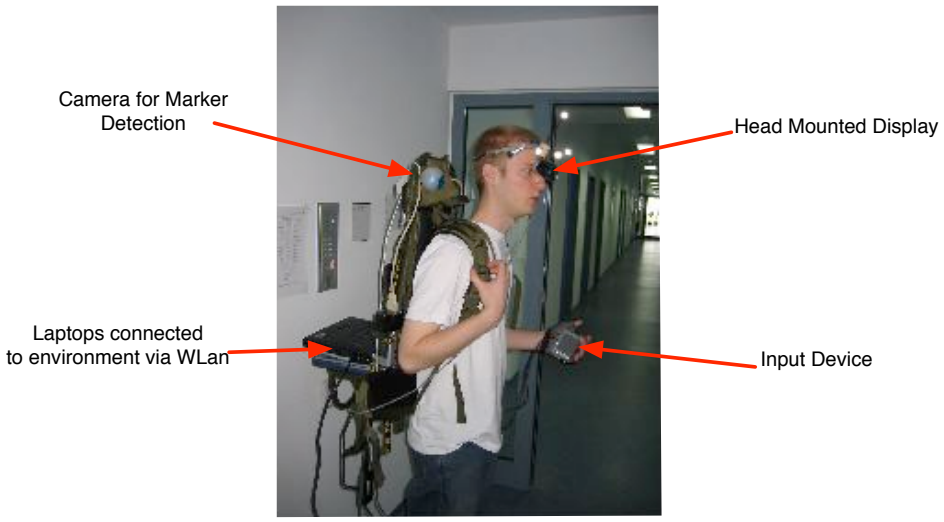


Figure 5: The mobile client.

everywhere. To access such information efficiently, caching and prefetching services should be added.

Acknowledgments

The work described in this paper was partially supported by the High-Tech-Offensive of the Bayerische Staatskanzlei.

The authors would like to thank all people involved in the ARCHIE project, in particular Felix Löw and Marcus Tönnis, and all members of the DWARF project for their collaboration on the framework. Special thanks to Allen Dutoit, Thomas Reicher and Christian Sandor for helpful comments and valuable dicussions.

References

ADDLESEE, M., CURWEN, R., HODGES, S., NEWMAN, J., STEGGLES, P., AND WARD, A. 2001. Implementing a Sentient Computing System. *IEEE Computer* (August).

BAUER, M., BRUEGGE, B., KLINKER, G., MACWILLIAMS, A., REICHER, T., SANDOR, C., AND WAGNER, M. 2001. Design of a Component-Based Augmented Reality Framework. In *Proceedings of the International Symposium on Augmented Reality*.

BAUER, M., BECKER, C., AND ROTHERMEL, K. 2002. Location Models from the Perspective of Context-Aware Applications and Mobile Ad Hoc Networks. *Personal and Ubiquitous Computing 6*, 5–6 (December), 322–328.

BAUER, M., BRUEGGE, B., KLINKER, G., MACWILLIAMS, A., REICHER, T., SANDOR, C., AND WAGNER, M. 2002. An Architecture Concept for Ubiquitous Computing Aware Wearable Computers. In *International Workshop on Smart Appliances and Wearable Computing*.

DEARLE, A., KIRBY, G., MORRISON, R., MCCARTHY, A., MULLEN, K., YANG, Y., CONNOR, R., WELEN, P., AND WILSON, A. 2003. Architectural Support for Global Smart Spaces. In *Proceedings of International Conference on Mobile Data Management*.

DEY, A. K., AND ABOWD, G. D. 2000. Towards a better understanding of context and context-awareness. In *Workshop on the What, Who, Where and How of Context-Awareness, affiliated with CHI 2000*.

ECHTLER, F., NAJAFI, H., AND KLINKER, G. 2002. FixIt. In *Demonstration at the International Symposium on Augmented and Mixed Reality (ISMAR 2002)*.

EFSTRATIOU, C., CHEVERST, K., DAVIES, N., AND FRIDAY, A. 2001. An Architecture for the Effective Support of Adaptive Context-Aware Applications. In *Proceedings of International Conference on Mobile Data Management (MDM 2001)*, Springer, K.-L. Tan et al., Eds., vol. 1987 of *LNCS*, 15–16.

GARLAN, D., SIEWIOREK, D., SMAILAGIC, A., AND STEENKISTE, P. 2002. Project Aura: Toward Distraction-Free Pervasive Computing. *IEEE Pervasive Computing 1*, 2.

GRIBBLE, S. D., WELSH, M., VON BEHREN, J. R., BREWER, E. A., CULLER, D. E., BORISOV, N., CZERWINSKI, S. E., GUMMADI, R., HILL, J. R., JOSEPH, A. D., KATZ, R. H., MAO, Z. M., ROSS, S., AND ZHAO, B. Y. 2001. The Ninja Architecture for Robust Internet-Scale Systems and Services. *Computer Networks 35*, 4, 473–497.

HESS, C. K., AND CAMPBELL, R. H. 2003. A Context-Aware Data Management System for Ubiquitous Computing Applications. In *Proceedings of the 4th International Conference on Mobile Data Management*.

HOHL, F., KUBACH, U., LEONHARDI, A., ROTHERMEL, K., AND SCHWEHM, M. 1999. Next century challenges: NEXUS – an open global infrastructure for spatial-aware applications. In *Proceedings of the Fifth Annual International Conference on Mobile Computing and Networking (MobiCom)*.

KALKUSCH, M., LIDY, T., KNAPP, M., REITMAYR, G., KAUFMANN, H., AND SCHMALSTIEG, D. 2002. Structured Visual Markers for Indoor Pathfinding. In *The First IEEE International Augmented Reality Toolkit Workshop*.

KLINKER, G., REICHER, T., AND BRUEGGE, B. 2000. Distributed User Tracking Concepts for Augmented Reality Applications. In *Proceedings of the International Symposium on Augmented Reality*.

KLINKER, G., DUTOIT, A., BAUER, M., BAYER, J., NOVAK, V., AND MATZKE, D. 2002. Fata Morgana – A Presentation System for Product Design. In *International Symposium on Aumgented and Mixed Reality ISMAR 2002*.

LIEBERMAN, H., AND SELKER, T. 2000. Out of context: Computer systems that adapt to, and learn from, context. *IBM Systems Journal 39*, 3&4.

MACWILLIAMS, A., AND REICHER, T. 2003. Decentralized Coordination of Distributed Interdependent Services. *IEEE Distributed Systems Online* (June). Accepted for publication as Middleware Works in Progress Paper.

MACWILLIAMS, A., SANDOR, C., WAGNER, M., BAUER, M., KLINKER, G., AND BRUEGGE, B. 2003. Herding Sheep: Live System Development for Distributed Augmented Reality. In *Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR)*.

MATTERN, F. 2001. The Vision and Technical Foundations of Ubiquitous Computing. *Upgrade 2*, 5 (October), 2–6.

REITMAYR, G., AND SCHMALSTIEG, D. 2001. OpenTracker – An Open Software Architecture for Reconfigurable Tracking based on XML. In *Proceedings of the ACM Symposium on Virtual Reality Software & Technology (VRST)*.

RICHÉ, S., AND BREBNER, G. 2003. Storing and Accessing User Context. In *Proceedings of the 4th International Conference on Mobile Data Management*.

ROMÁN, M., HESS, C. K., CERQUEIRA, R., RANGANATHAN, A., CAMPBELL, R., AND NAHRSTEDT, K. 2002. Gaia: A Middleware Infrastructure to Enable Active Spaces. *IEEE Pervasive Computing 1*, 4, 74–83.

SATYANARAYANAN, M. 2002. The evolution of Coda. *ACM Transactions on Computer Systems 20*, 2 (May), 85–124.

TSAI, R. 1987. A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses. *IEEE Journal on Robotics and Automation 3*, 323–344.

WAGNER, M. 2003. Configuration Strategies of an AR Toolkit-based Wide Area Tracker. In *Proceedings of The Second IEEE International Augmented Reality Toolkit Workshop*.

WEISER, M. 1991. The computer of the twenty-first century. *Scientific American* (Sep.), 94–100.

Empirical Evaluation of User Experience in Two Adaptive Mobile Application Prototypes

Leena Arhippainen
University of Oulu, P.O. Box 3000,
90014 University of Oulu, Finland
leena.arhippainen@oulu.fi

Marika Tähti
University of Oulu, P.O. Box 3000,
90014 University of Oulu, Finland
marika.tahti@oulu.fi

Abstract

Today's applications such as ubiquitous systems are more and more aware of user's habits and the context of use. The features of products and the context of use will affect the human's experiences and preferences about the use of device. Thus, user experience in user-product interaction has been regarded as an important research topic in the mobile application design area. The purpose of this paper is to clarify how user experience can be evaluated in adaptive mobile applications. The user experience evaluations were performed through interviews and observation while test users were using PDA-based adaptive mobile application prototypes. As a result, this paper presents the analysis of the test methods for further user experience evaluations.

CR Categories: J.m [Computer Applications]: Miscellaneous; Experimentation; Human Factors

Keywords: Adaptation, HCI, mobile device, user experience evaluation

1 Introduction

In the recent years, the use of different mobile products such as mobile phones and Personal Digital Assistant (PDA) devices has increased rapidly. Moreover, ubiquitous computing has become a popular topic in research and design areas. Nowadays, systems are more and more aware of their context of use. [Dey and Abowd 1999; Weiser 1991]

In order to be useful, ubiquitous applications need to be designed so that the user's needs and preferences and the context of use have been taken into account [Consolvo et al. 2002]. However, the evaluation of pervasive computing systems and their influences on users is quite difficult because the evaluation will require analysis of real users in a real context [Bellotti et al. 2002]. In addition, in continuous interaction research, test users should have a fully operational, reliable, and robust tool [Bellotti et al. 2002]. Evaluation with an incomplete prototype will not give a realistic test result. Nevertheless, preliminary tests in early phases of product development are necessary to perform in order to achieve information about the end user's preferences and needs.

In the recent years, in the Human-Computer Interaction (HCI) research area the capturing of user experience has been seen as an important and interesting research issue. In general, user experience has been captured with techniques like interviews, observations, surveys, storytelling, and diaries among others [Johanson et al. 2002; Nikkanen 2001]. However, in the HCI research area the understanding of user experience and its evaluation has not been established. One reason for this may be shortcomings in the definition of user experience and its relation to usability issues. Also, future proactive environments and adaptive mobile devices bring new aspects to the field of user experience research.

The aim of the paper is to study how user experience can be evaluated in adaptive mobile applications. User experience research and its methods are briefly presented in Chapter 2. Adaptive mobile prototypes and user experience evaluations are described and methods analyzed in Chapter 3. The results of the paper are presented in Chapter 4. Finally, the research is concluded and further work discussed in Chapter 5.

2 User Experience and Research Methods

Basically, user experience refers to the experience that a person gets when he/she interacts with a product in particular conditions. In practice, there are numerous different kinds of people, products and environments that influence the experience that interaction evokes (Figure 1). The user and the product interact in the particular context of use that social and cultural factors are influencing. The user has the following aspects: values, emotions, expectations and prior experiences, among others. Also, the product has influential factors, for example, mobility and adaptivity. All these factors influence the experience that user-product interaction evokes. [Dewey 1980; Forlizzi and Ford 2000; Hiltunen et al. 2002]

Moreover, in order to investigate user-product interaction, researchers need to determine the nature of a product. The type of the product will affect the research methods and targets. For example, user experience studies of web sites [Garrett 2002] emphasize visual issues whereas research of hand-held devices needs more attention on issues such as size, weight and mobility. Likewise, the evaluation of ubiquitous computing environments emphasizes different factors of a product and may thus require different methods for investigating user experience. In addition, the target use group needs to be defined before developing or testing prototypes; for instance, if the device will be put to public use and the users are not very familiar with computers, the interface should be simple and clear [Bellotti et al. 2002].

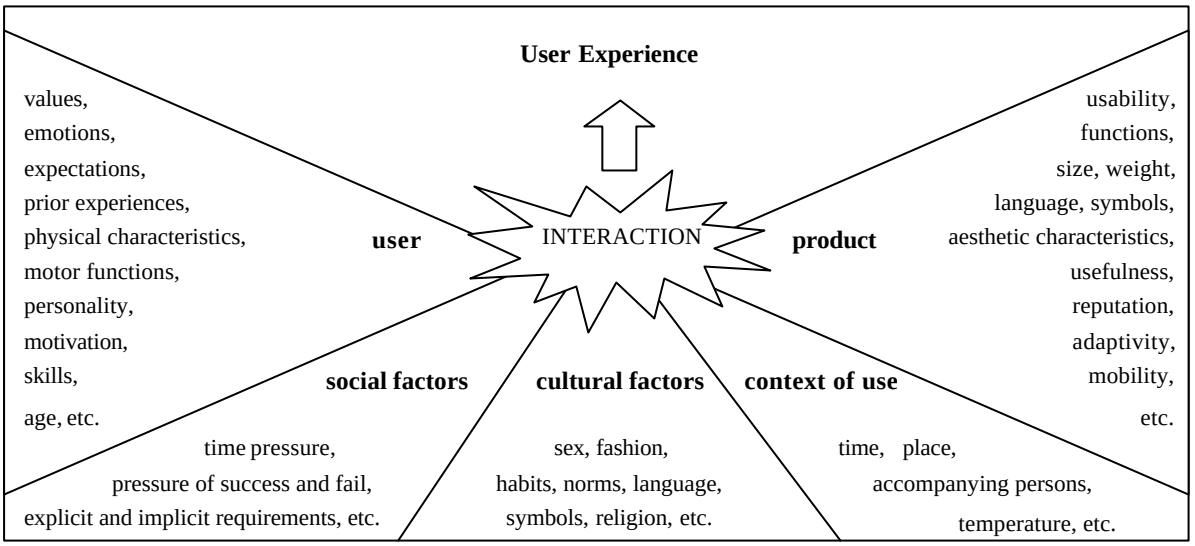


Figure 1.User experience forms in interaction with user and product in the particular context including social and cultural factors

There are several methods in the user experience research area that have been used for capturing experiences, for instance interviews, observation, surveys, diaries, storytelling and prototyping [Nikkanen 2001]. In long-term use, surveys, diaries [Palen and Salzman 2002] and storytelling have been regarded as an effective way to get information about user experience. That is because the user can express some of his/her experiences in a written form. Stories are ways to organize and remember experiences and they enable humans to communicate experiences in different situations to the particular people involved [Forlizzi and Ford 2000]. On the other hand, observation is a suitable method to gather user experience from non-verbal expressions. This is important, because the user may not be aware of his/her experiences or be capable to express them verbally. Buchenau and Fulton Suri [2000] have developed a method called Experience Prototyping for simulating experiences of different situations. The method allows designers, clients or users to "experience it themselves" rather than just witness a demonstration of someone else's experience. [Buchenau and Fulton Suri 2000]

Ubiquitous environments bring new aspects to user experience research. One reason for that is that environments and systems, according to Mark Weiser's vision [1991], should be invisible to the user; however, it should be possible to evaluate the interaction with the system. User experience in this kind of challenging environments and systems have been evaluated by interviews and observations in different ways [Bellotti et al. 2002; Johanson et al. 2002; Fleck et al. 2002].

Bellotti et al. [2002] have utilized different methods in their evaluations. In the first evaluation, they used two different questionnaires: a complete version and a short version. One year later, in the second evaluation, they performed ethnographic observation and qualitative and quantitative measurements [Bellotti et al. 2002]. Johanson et al. [2002] have developed interactive workspaces (iRoom) and performed some experiments of human-computer interaction. In these experiments they have utilized open participatory meetings with different sets of groups, e.g. expert and student project groups [Johanson et al. 2002]. Fleck et al. [2002] have developed an electronic guidebook for an interactive museum, called Exploration. In this museum, they have performed informal user studies by observing users with and without technology. Moreover, subjects were interviewed after the use of the prototype [Fleck et al. 2002].

3 User Experience Evaluations

In this study, user experience has been evaluated in two different adaptive mobile applications. Both the evaluation cases are presented and their weaknesses and strengths are discussed. Interview was selected as the method, because the amount of test users was quite small; this made it possible to observe user during the interview, and gave an opportunity to make the evaluation flexible. Observation was selected for gathering user experience from non-verbal expressions because the user may not be aware of his/her experiences, or be capable to express them verbally. Also, these methods were well suited for the test situation and resources. The interviews and observations have been analyzed from the user experience point of view, i.e. how well these methods suit user experience research.

3.1 The First Evaluation

The first evaluated prototype (Figure 2) is a context-adaptive application and it runs in a Personal Digital Assistant (PDA). This mobile device can localize the user by using WLAN (Wireless Local Area Network) positioning. In addition, the application can learn the user's habits on choosing the phone profile (normal, silent, loud) in a particular room by using routine learning algorithms.

Figure 2. The first adaptive mobile application prototype.

The context-aware application prototype can ask the user if he/she wants the profile to adapt automatically. The user can set the profile of his device and can make notes, for example mark a location of the meeting in the calendar of the prototype. Based on the context information of the user's location, the device can remind the user of the appointed meeting so that the user has enough time to get there, for example five minutes before the meeting. In addition, the user can add notes to the meeting place, and thus the prototype will give reminders of those notes in the meeting.

3.1.1 Test Scenario

The user experience of these applications' features was evaluated with the following predefined scenario:

- The user is using his/her mobile device in a corridor at his/her workplace. The application reminds the user that he/she has a meeting in one minute in a particular meeting room.
- The user enters the meeting place and the application automatically adapts into silent mode.
- When the user is in the meeting room, the application reminds him/her about any notes that he/she has made earlier. The note can be for example that the user must remember to bring forward some important things in the meeting.

This test scenario was created so that it enabled testing of the main adaptive features of the prototype. Moreover, it made it possible to evaluate user experience in the real environment. The first evaluation was conducted in an office-type environment. The test environment consisted of one big corridor and several rooms along it. One of the rooms was the meeting room where the user was going. The test was performed during the workday, and consequently there were some passers-by.

3.1.2 Test Methods

In this case study, the user experience evaluation techniques were interviews before and after the use of the prototype and observation during the use and the interviews. Users were asked to "think aloud" during the test. Interviews and observations were tape-recorded. The interview questions were developed on the basis of literature reviews [Bellotti et al. 2002; Dewey 1980; Forlizzi and Ford 2000; Hiltunen et al. 2002]. The questions concerned the user's prior experiences and present emotions, the prototype's mobility and adaptivity, context of use, and social and cultural factors. The interviews were organized so that one of the researchers asked questions and the other made notes, even though the interview was also recorded. The clerk also asked additional questions. Both of the researchers observed the test user during the interviews.

The observation was selected in order to get some information about the user's emotions and experiences, which he/she may not be able to describe him/herself. The observation focused on the user's facial expressions and behavior in general. During the test the observations about the user's gestures and behavior were written down. The whole evaluation including interviews took between 20 and 30 minutes.

The prototype was tested with three test users. The aim was to collect different user experiences, so two of the test users were familiar with computers and they had some background in information processing science. One of the users was quite inexperienced with computers and had a background in the humanities. No more test users were used because the aim of the first evaluation session was to collect preliminary information of the suitability of interviews and observations for user experience research.

3.1.3 Analysis of Test Methods

The study illustrated that interviews and observations are appropriate methods for evaluating user experience in user-product interaction. The documenting of the gestures and facial expressions during the test was slow and hard to combine with the particular action. In addition, perhaps some of the gestures were missed. Thus, in order to catch user experience from emotions and facial expressions, observations as well as interviews are better to record onto videotape.

Goal definition. The user experience evaluation elicited several improvements for the further tests. For example, a goal for the test has to be predefined, because user experience includes so many different aspects. In the test plan these factors have to be defined and a decision made about which information is needed to be captured; for example, whether experience relating to mobility issues is needed to be gathered, or also adaptivity and the context of use. Moreover, the test situation and atmosphere have to be as natural as possible for the test user because it will affect the kind of user experience that is formed.

Interview. It is important that the questions related to user experience are very simple so that the interviewee can understand them easily. Questions should not be strictly directed to user experience issues; for example, the interviewer should not ask, "Did the context of use affect the experiences that arose with the use of the application?" Instead, a better way is to ask "Can you tell something about this test situation? How did you feel about it?" In addition, the order of questions may affect how the interviewees understand the questions and this will influence user experience. For example, the interviewee should not be prompted by asking some questions about user experience before it is a topical issue. This is quite a challenge for user experience researchers because they have to find a balance for when to ask questions and when to expect the user to tell about his/her experiences freely. Nevertheless, it could be difficult for the user to express his/her personal experience verbally.

One of the interesting findings was that if the user was handling the product during interviews he/she may play with the product and not concentrate carefully on the interview questions. On the other hand, when the user has got a product in his/her hands he/she is more interested in discussing it and can perhaps express his/her opinions and experiences about the device better, because after familiarization he/she knows the device better.

3.2 The Second Evaluation

In the second evaluation, the adaptive application is implemented into a PDA-based prototype and it has an adaptive map-based interface (Figure 3). In this application, the map can be positioned according to the user's present location by using WLAN positioning.

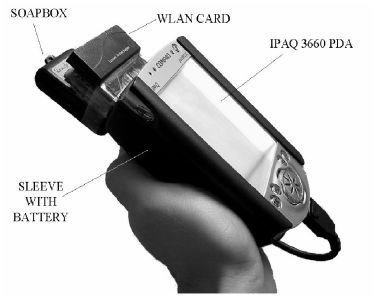


Figure 3. The second adaptive mobile application prototype [Rantakokko 2003].

Via the compass feature, the map can be rotated automatically according to a user's orientation. The user can zoom in and out on the map by moving his/her hand under the prototype's proximity sensor. The prototype zooms in when the user puts his/her hand closer to the sensor and vice versa. Scrolling the map is implemented by utilizing an accelerometer. Thus, the user can scroll the map by tilting his/her hand in four directions (forward, backward, left and right). In addition, the prototype's map includes objects and halos the aim of which is to help the user in finding different places and objects faster and more easily. With the halos the user can estimate the distance to a destination. [Rantakokko 2003]

3.2.1 Test Scenario

The test scenario was developed so that it was appropriate for the test environment. The environment was a home laboratory, which consisted of a kitchen, a living room, an office room and a hall. However, it was not appropriate to use the map-based prototype in the home environment. Instead, it is more sensible to evaluate the features of the prototype in some bigger context, for instance a health centre. So, we renamed the test environment more appropriately, e.g. living room was now waiting room.

The tests were performed with the following predefined scenario and test cases:

- **Test case 1:** The user is in the health centre and is given the gesture- and context-sensitive control device for finding the necessary places and objects. The user gets acquainted with the features of the prototype:
  - Positioning: the user identifies him/herself from the screen of the prototype.
  - Compass: The user uses the compass by holding the prototype in front of him/her and turning simultaneously.
  - Zooming: The user zooms by altering the distance from the bottom of the device to other objects.
  - Scrolling: The user tilts the prototype.
  - Service selection: The user clicks the icons and recognizes objects.
  - Halo: The user identifies the distance of objects through the arcs of halo circles.
- **Test case 2:** The user is sitting in the living room and waiting to see a doctor. He/she needs to go to the toilet and will use the prototype's feature to find out where the toilet is located.
- **Test case 3:** After going to the toilet, the user needs to wait for the doctor for a moment. He/she wants to change the channel on the TV, but first he/she has to find the remote control. He/she will use the prototype for that.

3.2.2 Test methods

The improvement ideas from the first evaluation were taken into account in this second evaluation. Hence, interviews and observations were recorded with a video recorder. Before the actual tests a pilot test was performed and it confirmed that the test scenario and cases are appropriate.

At the beginning of the test, the users' backgrounds as well as their familiarity with mobile applications were determined. Interview questions were updated from the first evaluation and they concerned the user's prior experiences and expectations, the prototype's mobility and adaptivity, context of use and social factors.

This evaluation was conducted in a laboratory, because the tested prototype required a particular WLAN environment in order to operate. The user experience evaluation was carried out with ten test users. The purpose was to get user experiences from different kind of users. Thus, half of the users were experienced and had been using PDA devices a lot or at least a little, or they had a background in technology. The other half had never used a device like that and came from different fields of occupation. In each evaluation, the whole test situation including interviews took approximately an hour.

In addition to the interview and observation, the questionnaire and user instructions for the prototype were sent to the test users afterwards (a few days later). The purpose of the questionnaire was to clarify the user's experiences about the prototype and its features as well as the whole test situation. This gave the users the possibility to think about the test more carefully than just in the test situation.

3.2.3 Analysis of test methods

The second study also supports the view that interviews and observations are methods that can be used for evaluating user experience, because a lot of the user's thoughts, experiences and emotions can be captured. Nevertheless, these methods are not enough in order to get a deeper knowledge of the user's emotions and experiences. The evaluation of user experience will need more capable ways to catch user experience.

Video recording. Although the selected methods seemed to be good, some observations and problems arose. For example, tests were videotaped in front of the users, and this perspective gave information about the user's facial expressions such as eyebrow movements. However, when the user's head was down while he/she was watching the device, it was difficult to see all the facial expressions (Figure 4). Another problem was videotaping when the user has to move between different rooms (Figure 4). He/she could turn his/her back to the camera. The biggest problem in the video recording was that the screen of the device was not recorded at all. Information about what happened on the

screen was captured via the user's "thinking aloud" and the moderator standing alongside the test user and watching the screen.

The use of video recording elicited some new questions for user experience research: Does video recording have an influence on the user's behavior and user experience formation? Do these possible impacts influence user experience? Can we collect all emotions only by using video recording? For example, one test user said that she is very nervous, but that was impossible to interpret from her facial expression and gestures. So, this user experience (emotion) was captured via interview – not by observing.

Interview. The first evaluation showed that user experience questions should be formulated in a particular way, so that that the user can understand them easily. In this second evaluation, interview questions were made easier and "the user's language" was used. However, some of the test users regarded the questions as difficult. Does this mean that it is difficult for the user to verbally express his/her experiences? Is some visual expression an easier way for the user to express his/her emotions and thoughts?

Test situation. Organizing the test elicited several problems as well. Firstly, the test premises were too small in order to test some features of the prototype, for instance the benefit of positioning. Secondly, the test included too many features for the users to use the prototype and learn all functionalities. The test could be organized so that it is divided into two sessions where a part of the features will be tested first and the rest of them a little later (a week or two). Thus, the learning of the use

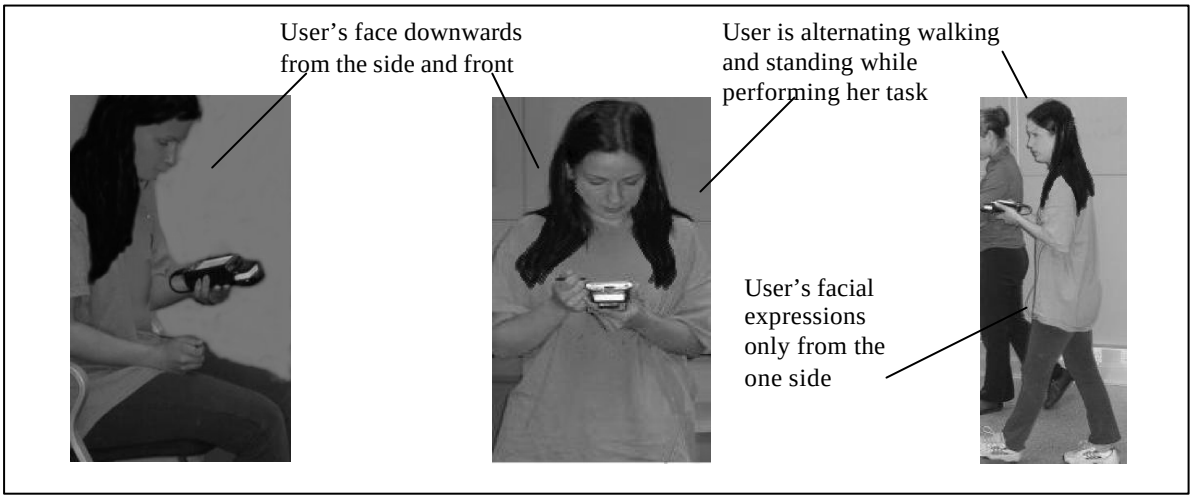


Figure 4. Capturing the user's facial expressions, gestures and body movements in interaction with the prototype.

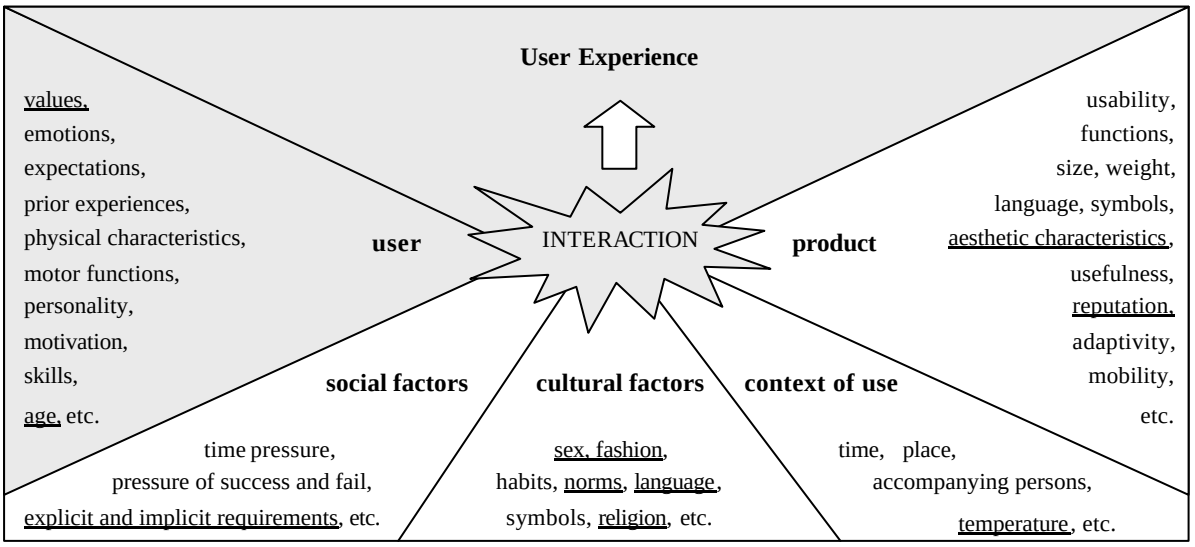


Figure 5. User experience factors captured via interviews and observations.

of the application can be also evaluated. From the viewpoint of user experience research, there were too many (five) tests in one day. Thus, researchers had to keep a schedule in order to be ready for the next test person. The tests were planned to be conducted one after the other. This strict schedule may have certain influences on user behavior and thus user experiences.

4 Results

This chapter is divided into two parts. Firstly, the benefits and challenges of the interview and observation methods from the viewpoint of user experience research are summarized. Secondly, the suitability of interviews and observations for user experience research is discussed.

4.1 Benefits and Challenges

Interview is a good method for user experience evaluation, because then the test situation can be like a "chat session" with the test user. It gives the possibility to create a calm and nice atmosphere in test situation. This is also an easy way to get information about the user's background (age, education), prior experiences, expectations and motivation, etc.

Table 1. User experience factors captured via interviews and observations.

Factor	Int.	Obs.	Comment
User			
values,	[]	[]	NE
emotions,	[]	[x]	- Difficult for user to express emotions verbally.
expectations,	[x]	[x]	- Interview gave this information very well, also observation gave information about user's appearance (shy, enthusiastic)
prior experiences,	[x]	[x]	- Good and bad experiences about technical devices. - Use of device and understanding of symbols.
physical characteristics,	[x]	[]	- Small hand vs. big device.
motor functions,	[]	[x]	- Affected user experience (how well can use the product)
personality,	[x]	[]	- Personality affects storytelling (-> user experience results as well). It also affects observation (gestures).
motivation,	[x]	[x]	- Motivation and lack of motivation were noticed.
skills,	[x]	[]	- Skills influenced user experience (use of different equipments like compass).
age	[]	[]	NE
Product			
usability,	[]	[x]	- Usability issues were not interviewed/tested, however observation elicited that it affects user experience.
functions,	[x]	[x]	- Interviews and observation gave a lot of information about the functions of the device.
size,	[x]	[x]	- Size of device vs. size of user's hand and use of both hands.
weight,	[x]	[]	- Weight affected use and thus user experience.
language,	[x]	[x]	- Affected understanding of the device, and thus user experience in a negative or positive way.
symbols,	[x]	[x]	- Affected understanding of the device, and thus user experience in a negative or positive way.
aesthetic characteristics,	[]	[]	NE
usefulness,	[x]	[]	- Affected user experience in a positive way.
reputation,	[]	[]	NE
adaptivity,	[x]	[x]	- Hard for the user to explain this factor, so the observation gave more information about it.
mobility	[x]	[]	- Was regarded as an obvious and positive aspect.
Social factors			
time pressure,	[x]	[x]	- Users leaned on the moderator a lot.
pressure of success and fail,	[x]	[x]	- Users explained a lot why they cannot use the device.
explicit and implicit req.	[]	[]	NE
Cultural factors			
sex,	[]	[]	NE
fashion,	[]	[]	NE
habits,	[]	[x]	- Users compared features to a magnifier (zooming) and a glass of water (scrolling).
norms,	[]	[]	NE
language,	[]	[]	NE
symbols,	[x]	[x]	- Symbols were familiar from other contexts.
religion	[]	[]	NE
Context of use			
time,	[x]	[x]	- Influenced user experience.
place,	[x]	[x]	- Interview and observation gave different information about its influences on user experience.
accompanying persons,	[x]	[x]	- No influence on user experience.
temperature	[]	[]	NE

However, there are some interesting challenges for the interviewers to clarify. Firstly, questions related to user experience should be formulated very carefully so that the users can understand them easily. Secondly, usually the user can express his/her opinions about a device and its characteristics, but verbally describing his/her feelings about the device is more difficult. In that kind of a situation, the interviewer can try to "read between the lines" when the user speaks about his/her experiences. Nevertheless, this challenge may require using some other methods as well.

Observation also gave information about user experience. However, researchers need to interpret the user's facial expression, body movements and gestures carefully, because the personality of the user will affect how they behave. For example, one test person said that she is very nervous, but her outward appearance was really calm. Moreover, humans make gestures very differently, for instance while one moves his or her eyebrows a lot, the other can move his/her eyes only a little.

These two user experience evaluations elicited that a comprehensive observation will require video recording. In the first evaluation, video recording was not used, and thus only some facial expression was captured. However, the second evaluation was video recorded but still some challenges occurred. The first thing in video recording in user experience research is that it must not influence the user and his/her experiences. This is an interesting challenge. However, in order to collect the user's facial expressions, gestures and actions on the screen, the video recording should be organized from different perspectives, for instance, from the front of the user's face, the top of the screen and a little bit farther away so that the user is in the picture. In order for the observation to be reliable, a tool or a method for interpreting different gestures and emotions is required.

4.2 Suitability for user experience research

The picture (Figure 1) presented in Chapter 2 illustrates what different factors affect user experience in user-product interaction. In evaluations, some factors can change; for instance, in the user experience evaluation presented in this paper, the user was one part that changed. The device, social and cultural factors and the context of use were the same. Consequently, when the user changes, interaction and user experience change as well (grey areas) (Figure 5).

User experience factors can be captured via interviews or observations on a particular level. Factors, which did not appear in the evaluations, are underlined in the picture (Figure 5) and marked as NE (Not Emerged in the evaluations) in the table (Table 1). However, this paper does not deny that those factors could not be captured via interviews and observations.

The evaluations elicited that some user experience factors can be gathered via both of the methods. For example, the user can comment on the product's functions and say that they are easy to understand and learn. However, when he/she uses product, the observer can perceive that he/she uses it in the wrong way. On the other hand, observation does not always bring out the user's emotions properly, and thus interview can reveal the true emotions more easily. Hence, interviews and observations can give different information about the same factor, and thus give a more comprehensive view to user experience. This paper presents what user experience factors were captured via interviews and observations (Table 1).

5 Conclusion

The purpose of this paper was to define how user experience can be evaluated in adaptive mobile applications. In general, the capturing of user experience is quite difficult, because there are so many different factors in user-product interaction (Figure 1). For the evaluation, those factors should be clarified and a goal for the test defined in a test plan. This may help make the evaluation more systematic.

Both the examinations illustrated that interviews and observations are appropriate methods for capturing user experience (Table 1). However, this study confirmed that several methods need to be used in order to evaluate user experience. In addition to the interviews and observations, researchers will need more efficient ways to get information about the user's emotions and experiences, concerning for example collection and interpretation of body gestures and facial expressions. In order to collect authentic emotions, the test situation should be organized so that is as natural as possible. As further research, more user experience evaluations will be done for different adaptive mobile devices, using different methods.

References

BELLOTTI, F., BERTA, R., DE GLORIA, A. AND MARGARONE, M. 2002. User Testing a Hypermedia Tour Guide. *IEEE Pervasive Computing*, 33-41.

BUCHENAU, M. AND FULTON SURI, J. 2000. Experience Prototyping, *in Proceedings of the DIS 2000 seminar, Communications of the ACM*, 424-433.

CONSOLVO, S., ARNSTEIN, L. AND FRANZA, B. R. 2002. User Study Techniques in the Design and Evaluation of a Ubicomp Environment. *In the Proceedings of UbiComp 2002*, LNCS 2498, Springer-Verlag, Berlin, 73-90.

DEWEY, J. 1980. Art as Experience, New York: Perigee, (reprint), 355.

DEY, A. K. AND ABOWD, G. D. 1999. Towards a Better Understanding of Context and Context-Awareness. *GVU Technical Report*. GIT-GVU-99-22. Georgia Institute of Technology.

FLECK, M., FRID, M., KINDBERG, T., O'BRIEN-STRAIN, E., RAJANI, R. AND SPASOJEVIC, M. 2002. From Informing to Remembering: Ubiquitous Systems in Interactive Museums. *IEEE Pervasive Computing 1/2*, 17-25.

FORLIZZI, J. AND FORD, S. 2000. The Building Blocks of Experience: An Early Framework for Interaction Designers, *in Proceedings of the DIS 2000 seminar, Communications of the ACM*, 419–423.

GARRETT, J. J. 2002. The Elements of user experience. User-centered design for the web. New Riders, 208.

HILTUNEN, M., LAUKKA, M. AND LUOMALA, J. 2002. Mobile User Experience, *Edita Publishing Inc.* Finland, 214.

JOHANSON, B., FOX, A. AND WINOGRAD, T. 2002. The Interactive Workspaces Project: Experiences with Ubiquitous Computing Rooms. *IEEE Pervasive computing 1/2*, 67-74.

NIKKANEN, M. 2001. Käyttäjän kokemusta kartoittavien tutkimus- ja suunnittelumenetelmien käyttö tuotekehitysprosessissa. Licentiate's degree. University of Helsinki, 102.

PALEN, L. AND SALZMAN, M. 2002. Voice-mail Diary Studies for Naturalistic Data Capture under Mobile Conditions, *CSCW*, New Orleans, Louisiana, USA, November 16-20, 87-95.

RANTAKOKKO, T. AND PLOMP, J. 2003. An Adaptive Map-Based Interface for Situated Services, *in proceedings of the Smart Objects Conference*, Grenoble, France.

WEISER, M. 1991. THE Computer for the 21st Century. *Scientific American* 265(3), 94-104.

A Platform Independent Image and Video Engine

David Doermann[1], Arvind Karunanidhi[1], Niketu Parekh[1], Ville Rautio[2]

[1]Laboratory for Language and Media Processing, University of Maryland, College Park, USA
doermann@umiacs.umd.edu, arvind@umiacs.umd.edu, nkparekh@umiacs.umd.edu
http://www.umiacs.umd.edu/lamp

[2]Department of Electrical and Information Engineering, University of Oulu, Finland
wilse@ees2.oulu.fi
http://www.mediateam.oulu.fi

ABSTRACT

This paper describes progress on the development of a platform independent media engine for mobile devices. The effort is part of the CAPNET (Context Aware Pervasive NETworks) project and focuses on providing a multimedia presentation and analysis service as part of a larger mobile services architecture. We present a high level overview of design, provide a general description of the functionality of various components such as MediaAlerts, MediaCapture, MediaStorage and MediaProcessing and describe several applications in progress, which utilize media engine components.

1 Introduction

Current mobile applications make limited use of video as an information rich content that can be analyzed, manipulated, indexed and retrieved. Typically, content is in the form of individual images or video clips, possibly containing metadata created by hand. Often they are used as multimedia "messages", without any further consideration of how the content can be used or reused for other purposes. On desktop computers, users are becoming more accustomed to being able to edit and integrate image and video content into standard communications via email and WWW services. As the proliferation of camera and multimedia devices gain momentum, users will demand merging of these same capabilities in their mobile environments. We will expect to be able to easily add text, construct albums, exchange media clips and retrieval media from various remote repositories. Although somewhat cumbersome in their design, many of these applications are now beginning to appear.

At the same time, the research community is focusing extensively on the automated analysis of images and video from a wide variety of electronic sources including newscasts, surveillance, home video, web sources, and digital camera stills, as well as digitized or scanned archival material, newspapers and magazines. As these technologies mature, there will be a demand for integrating them into everyday use, and ultimately into mobile applications. Tasks such as automatic summarization and abstraction of sports video have been reported as a convenient way to deliver content efficiently to bandwidth- and display-limited mobile devices [3]. Similarly, document image analysis has been used to extensively analyze and reflow images of documents for display on devices with limited screen real-estate [9]. Providing such capabilities will ensure a path to integrating many types of multimedia not traditionally suited for mobile devices, in a way that users will ultimately demand.

Even more exciting is the possibility of integrating analysis of content captured with these devices. When face recognition technology matures, we could imagine taking a picture with our devices, passing it to a server, and asking questions such as "Who is that person? I have met them somewhere before". If our database is constrained to known acquaintances, for example, or to the people one would expect to see at a given University, the problem becomes feasible. New usage paradigms are emerging for mobile devices configured with cameras behaving as input tools for fax transmission, bar code recognition and optical character recognition. For example, various groups have successfully demonstrated the ability to integrate image capture, OCR and translation of Chinese signs on a PDA [2].

Over the past two years, the University of Maryland, in cooperation with the CAPNET (Context-Aware Pervasive Networks) Project at the University of Oulu, Finland, has been designing and implementing a set of components, which facilitate the rapid development of mobile multimedia analysis and management capabilities. The goal is to be able to provide an

integrated method to expand and rapidly prototype various pervasive applications. In MUM2002, we presented an overview and motivation for multimedia processing [7]. Recently, we have focused on implementing a number of key components including capture, storage/retrieval, media messaging and alerts and remote image and video processing. In this paper we will highlight the current architecture, and described the components fundamental to the Media Engine. The potential for customized multimedia information sources to be both "produced by" and "delivered to" users in a pervasive environment opens up numerous new and exciting applications. Our goal is to develop a framework and associated toolkit to facilitate seamless media processing.

2 Background

The CAPNET program creates a foundation for new information and communication technologies and new business models for mobile services. The research makes use of key ideas of ubiquitous computing and focuses on service profiling, software technology, and content rich environments. The program provides an innovative architecture that allows us to experiment with adaptive applications, and to prototype and test business scenarios.

2.1 CAPNET Architecture

The CAPNET architecture is based on the need for the user to interact with a mobile device, while making use of distributed network centric services.

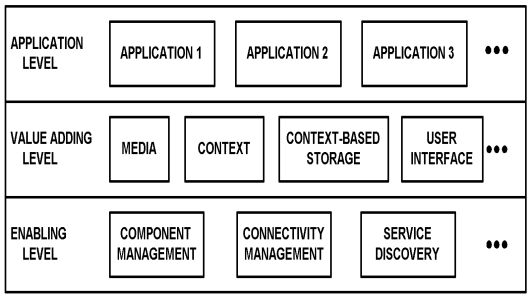


Figure 1: CAPNET Architecture Components

To facilitate operations and interoperability over a wide variety of configurations, the architecture must be adaptable. To accomplish this, it divides services into engines that implement a set of components that are necessary to support functionalities in each domain (Figure 1). These engines currently include *component management, connectivity, service discovery, context, storage, user-interface* and of course *media*, with the first three being required engines. The media engine framework is heavily dependent on the required engines. The component management engine handles direct configuration of the components for other engines while the connectivity engine handles messaging between components. Service discovery components are used for locating resources and services in the CAPNET space based on context information, service meta-data, user profiles, etc.

Each engine is dynamically reconfigurable and divided into a set of static components and dynamic components. The static components implement the basic functionality of the engine, with one of the static components being considered a core component and implementing essential capabilities. The dynamic components are downloadable and depend on the network, device configuration, and user preferences. Overall the components act independently, and rely on messages to communicate. More information about the general CAPNET program can be found at [8].

2.2 Mobile Media

As the next generation of networks make good on promises to provide seamless realization of mobile multimedia, service providers and application developers will look for new ways of providing multimedia content. In particular, frameworks such as Capnet will seek to abstract many of the lower level problems of delivery, device management, manipulation, and compression away for the application. As we have seen in PC environments, media support is being integrated at the operating system level; abstracting many of the challenges and making issues like device management transparent to application developers. Our media engine will attempt to abstract many of those same services for mobile multimedia.

2.3 Related Work

Although many groups have described approaches to the development of multimedia applications, it is essential that our work be integrated with the more general CAPNET component architecture and make use of its services. The approach allows us to build components that will be useful and extensible, but not necessarily interact smoothly with other standards.

Bates et al [5] describe a framework for the construction of mobile multimedia applications, which supports dynamic creation of media objects and movement of media objects from one terminal to another. Our media engine framework also supports storage and retrieval components and processing of media, which seems to be absent in their current framework. Media objects are quite similar in both the approaches except that the metadata is tightly integrated with the media. Low-level communication facilities are employed in their framework while we use xml-rpc communication standards and serialization to transfer media objects.

Another conceptual framework called the Situated Computing Framework [4] has been designed for mobile devices to access rich multimedia services and it provides a smart delivery service of multimedia content which calculates the optimal delivery sequence based on frequency of service request,

priority, type of media and type of output device. The remote invocation process running on the mobile devices is realized using Distributed Component Object model which makes it platform dependent. Our framework tries to solve this problem by using Java and XML - remote procedure calls.

A programming framework [6] has also been designed for quality aware mobile multimedia applications that can describe a high level application with quality requirements and controls of adaptation. The ubiquitous multimedia component model employed by this programming framework has a slight edge over our architecture since it uses QoS (Quality of Service) descriptors which will prove invaluable in case of applications like real-time video streaming.

3 Media Engine

The MediaEngine seeks to provide a seamless integration of multimedia services for end user applications, while maintaining its platform independent architecture. To do this, the system must be dynamically reconfigurable, so, for example, services that cannot run locally because of limitations of the device, constraints of the network or demand of the user, can run on the network or on a server. The MediaEngine, therefore, must provide basic capabilities, along with the logic necessary to dynamically reconfigure.

Like other CAPNET engines, the Media Engine is divided into a set of static components, containing the media core, and dynamic components, which can be adapted at runtime to provide the additional functionality. The Media Core is always present when there is any media engine related functionality present. Dynamic components are loaded on-demand, based on needs expressed by the client application.

All of the components are managed within the Capnet architecture, and hence it is not necessary to implement additional capabilities such as storage, or client server communications. The server side components are responsible for managing services that are used by individual clients, as well as shared services, that have performance requirements that cannot be satisfied by the client. The client components interact with the server by requesting services through a message passing interface.

Unlike other work, the media engine is being designed to provide not only multimedia download and display but also capture and delivery, storage, retrieval, local and remote processing. To realize these services, we have split the media engine into six components. The core components consist of MediaObject and Media Metadata while the dynamic components are comprised of MediaCapture/Playback, Media Storage/Retrieval, Media Processing and Media Alerts/Messaging. Figure 2 shows the overview of media engine architecture. In the remainder of this section, we provide an overview of the functionality of each component, and how they are used by end-user applications.

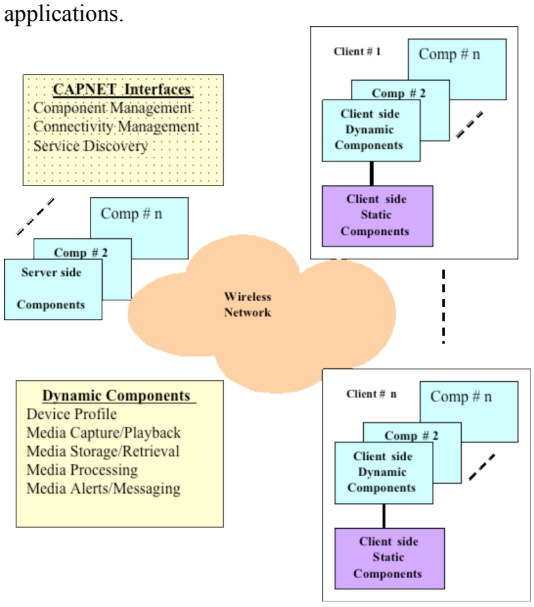


Figure 2: MediaEngine Overview

3.1 Core Media Component

The core component is the static part of the media domain area runtime structure. The core component defines the media objects and functions to create and manage them, including start, stop, reinitialize, etc. All dynamic components handle media as objects that are defined in the core of the domain area. Media core is always composed of the MediaObject component and a related MediaMetadata component.

3.1.1 MediaObject

A MediaObject defines the structure of media content and contains data along with the media metadata component. In case of composite media, the media object contains the data of all media elements, SMIL 2.0 (Synchronized Multimedia Integration Language) based presentation of the spatial, temporal and navigational relations of different media elements and metadata of each media element. Spatial relations define how media elements are spatially related to each other (Element A is on the left of element B). Temporal relations define how media elements are temporally related after each other (Element A is played after Element B). Navigational relations define how media elements are linked to each other (Element A could contain a hyperlink, which would take media playback into state where Element B is being played). Elements in media objects can be text, audio, image or video. Applications are needed to pass data between media components and they handle media objects in serialized form and can be deserialized at the receiver end. They do not have to understand any of the

internals of the media objects making it device independent.

3.1.2 MediaMetadata

The MediaMetadata component defines the metadata that is created for each media element and maintains information about the properties of the media object. Having the metadata of the media object is necessary for components that will manipulate the object, such as storage and retrieval components. A graphical description of the constituent parts of the metadata component is shown in Figure 3.

The MediaObject's metadata is split into four major elements: MediaObjectType, MediaObjectProperties MediaObjectCreation and ChangeHistory. Each element is further divided into sub-elements and contains more information about the media object. The MediaObjectType categorizes the media content while the MediaObjectProperties describe the characteristics of the media content. Media ObjectProperties include the color, orientation, duration, encoding rate, bits per pixel, key frame interval, type of codecs etc, necessary to render or process it. The MediaObjectCreation sub-elements include the device source, camera type, resolution, media source and date/time of creation. The

3.2 Dynamic Media Components

Dynamic Components are components which can be adapted at run-time to provide additional optional functionality. Dynamic components are loaded on-demand based on the needs of the application using the mechanisms provided by the service discovery and component management. Some of the dynamic components are constrained to run only on the server-side.

In the current implementation, we provide basic capabilities necessary to demonstrate the system. The DeviceProfile provides a platform and hardware specialization mechanism to allow easy reconfiguration on a new device. MediaCapture and Playback provide basic support cameras and rendering respectively. MediaStorage and Retrieval provide a way for end user applications to transparently access multimedia repositories. Finally, MediaAlerts and MediaMessages provide a way for pervasive applications to communicate with users and for users to share multimedia content either downloaded to or captured by the device.

3.2.1 DeviceProfile

As a result of technological advances and new

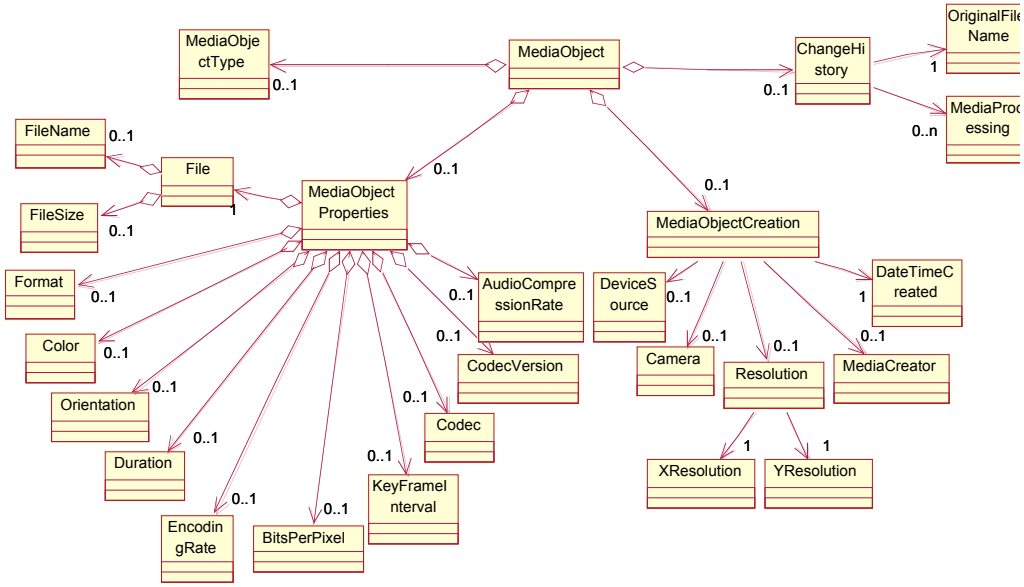


Figure 3: Media Object and Metadata Schema

ChangeHistory provides a container for information about the changes made on the media object (such as being processed remotely with some set of image processing algorithms, and has the actual file name along with the media processing type as its sub-elements.

manufacturers, the mobile device market has diversified, which makes the mobile devices more difficult to manage from the application service provider's perspective. In order to solve the problem, device information should be readily available so that the application knows the capabilities of the device and can tailor the content based on that information. For example, the application can automatically scale

the image based on the device display size and resolution. Hence, a device profile component is required which maintains a centralized repository of device specific information for various mobile devices. The device profile is a CC/PP (Composite Capabilities / Preference Profile) XML file, which contains the display resolution, display depth, supported media formats and characteristics of the mobile access network. This component provides APIs to parse the device profile information from the XML file using MinML and be used by the server to configure dynamic components for download.

3.2.2 MediaCapture and Playback

The media engine offers a media capture and playback component for applications to capture and play media content. The device must have the necessary native hardware and drivers to access it. In case of unsupported or missing capture devices, initialization of this component will fail and exit gracefully. The capture component queries the device profile component before initializing to see whether the device has the capability. The selection of an appropriate driver is completely transparent for the components and applications using it. Media Capture has a video preview component with which the application can utilize it before the image/video capture. Video capture requires the appropriate codecs to be downloaded dynamically to the device. For video capture on the PDA, we are currently using Microsoft Portrait. Portrait provides us with a compressed video stream broken up into frames which we can store locally, upload to a server, or upload to the server and transcode for delivery to other devices. The Media capture supports the Flycam Compact Flash camera, Pretec camera and the HP Pocket camera.

Video playback currently uses a native rendering component for standard encoded media such as MPEG or AVI, and we are developing a rendering component for either custom encodings or data we can easily decode such as MSPortrait.

3.2.3 MediaStorage and Retrieval

The Media Storage component offers media storage and retrieval capabilities for other components and applications. This function provides an extra layer of context-based storage, offering media specific functionality in an easy to use and transparent way.

The MediaStorage component indexes media objects using their metadata, which allows more flexible queries. This component assumes that the context-based storage is available for use from the device that it is being executed on. The client component can store several media objects by providing a MediaObjectList in a vector format. Similarly, a query can be formulated by the client to retrieve media objects all at once which returns a vector of serialized media objects.

The component also provides temporary storage on the local device and brings up a graphical user interface to browse media files stored locally and/or remotely and the ability to retrieve media files matching specific metadata. The application makes calls to the storage functions, and media is transferred and stored without regard to network, transfer protocol, location of the server-side storage or the type of the repository.

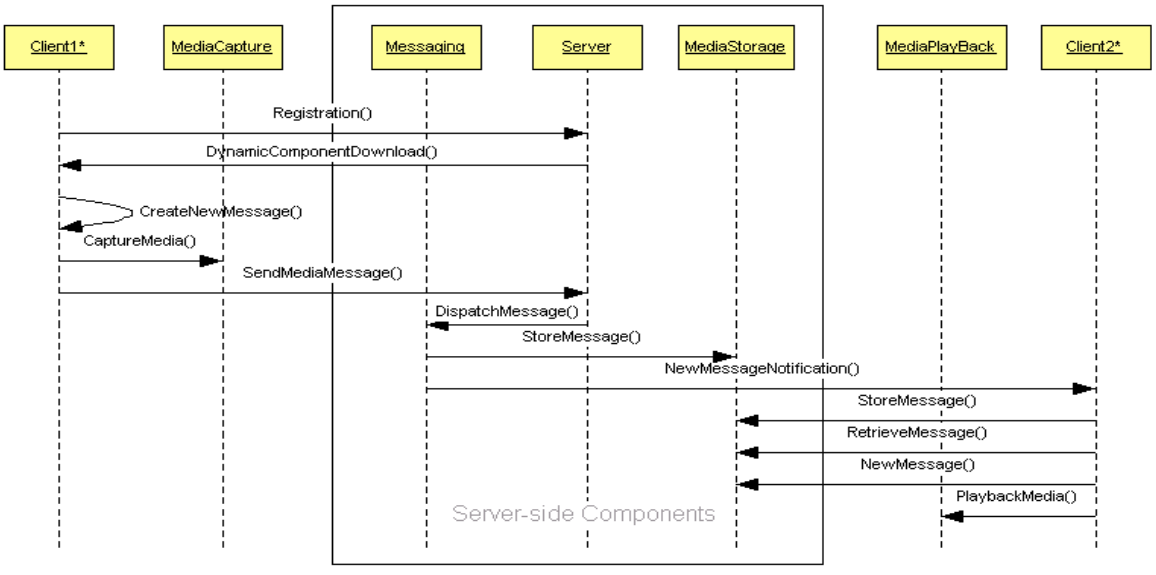


Figure 4: Example of Component Interaction for Media Engine

3.2.4 MediaAlerts and MediaMessages

The MediaAlert component provides a push-type media object delivery mechanism for the components and applications. This component borders on the level of an application and provides an essential service for pervasive applications. Components and applications that want to receive push-type content must subscribe to the MediaAlert component to receive it. A MediaAlert component can be used for two different purposes:

1. Pushing media alerts to components and applications.
2. Delivering media messages between components and applications.

The first is intended primarily for third party applications, such as advertising tools, in pervasive environments. The second should be though of as a way to deliver multimedia messages between clients.

Media alerts are generated when some event occurs and they are delivered to all components and applications that have subscribed to receive media alerts. Components and applications use a media alert to deliver messages to components that have subscribed by providing the media object and recipient's identification.

Components and applications can also invoke different events on the media alert component. This allows a flexible infrastructure where new alert generating components can be written easily. Clients subscribe for media alerts and they receive notification of an event, if they have subscribed to events with a unique identification. Clients can discover different types of event generating components with a service discovery component.

The functionality of the media alert component is asynchronous. If a client is not available, the media alert component will queue the messages and alerts that the client component would get if it were available.

In case of MediaMessaging, the components that wish to receive media messages subscribe to the sender of the message, which can forward messages to the receiver by using this component. Figure 4 illustrates an application using the media messaging component and its interaction with the other media engine components.

3.2.5 MediaProcessing

This component extends image-processing and manipulation functionalities, downloaded dynamically to the device. It currently includes basic image processing techniques (sharpening, blurring, edge detection, image transforms) as well as advanced image analysis and applications such as text detection and recognition, bar code recognition etc. Some of the media processing components involve the use of native code and achieving platform independence for these is an arduous task. Currently, we have those media processing components compiled for a specific platform and the appropriate components are downloaded based on the information specified in the device profile. If the device does not have the required resources to do advanced media processing, the client is provided with an alternative to do the media processing on the server side and return processed media. The goal is to set up Media Processing as a service oriented component, and make use of service discovery and the network centric processing to provide a wide range of processing capabilities.

4 Implementation

The media engine is designed to thrive in a heterogeneous wireless environment, which may have different configurations of processor architectures and platforms. We have implemented all components using Java. As mentioned previously, the engine has both static and dynamic components, which interact with each other using a generic message format. The server has a dedicated request handler for each component (e.g. MediaAlertHandler, MediaProcessingHandler, MediaStorageHandler), which processes each request coming from its clients and sends the response back from the server. The messages (requests and responses) between client and server components are asynchronous. If the client has to send a new request, it does not need the previous request. This asynchronous message exchange over wireless between client and server is achieved using a generic message format where each message is identified using a unique target component identifier. Our current demonstration environment for the media engine uses an iPAQ on the Windows CE platform, fitted with 802.11b wireless LAN card and a pocket camera. Following paragraphs discuss the implementation details of each component.

The MediaStorage component implements a simple database application where a user can store a media object from a mobile device to the media server (in BLOB format on MySQL database server) or retrieve a media object from the server and view it using the remote file browser facility. The remote file browser allows user to retrieve and/or view remotely stored media objects based on different attributes such as owner, resolution, last modified date/time etc.

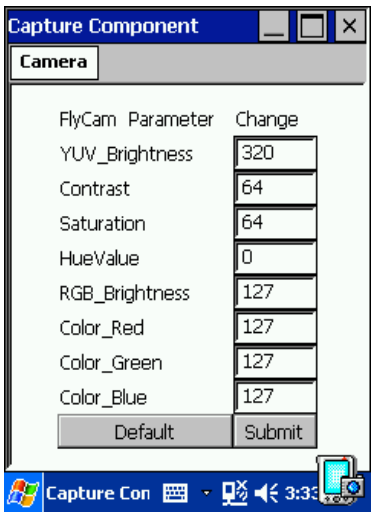


Figure 5: UI for the camera parameters

The MediaCapture component provides a user interface, which allows the user to invoke the various camera functionalities such as initializing the camera, capturing image/video, setting parameters etc. This component is capable of controlling a native API based camera as well as Java API based cameras. To control the native API based cameras, it uses Java Native Interface (JNI). This component has a viewfinder with which we can see the object and capture the image in a bitmap format.

As shown in Figure 5, the MediaCapture component allows the user to modify various camera settings such as brightness, contrast, saturation and hue. Considering the fact that most of the imaging devices for PDAs support native interface rather than java interface at the driver level, MediaCapture component is designed to provide a common interface to invoke native camera application using JNI, as well as camera applications supporting the java interface. The captured images are in Device Independent Bitmap form, which can be dynamically adjusted to adapt properly on any display device. In case of video capture, the video is encoded with low bit rate encoder and stored locally on the device.

The MediaAlerts and MediaMessaging components provide a framework which can be used by third party applications (such as video surveillance) to distribute alerts, facilitate advertising, or to share media between users. On the server side, it keeps the latest information about the subscribers for a particular type of alert. When the application generates an alert message, this component retrieves subscriber information from the database and distributes the alert message to all the subscribers. To exchange media content between users directly, it refers to the registered users information maintained in the database and forwards the message to the recipient of the message. The advantage of this mechanism is that even if the two users exchanging media are part of isolated networks, the MediaAlerts and Messaging component, can transfer information end to end.

5 Applications

We are currently using the media engine to implement a number of use case and application scenarios both inside and outside of the Capnet Architecture.

5.1 MediaToolkit

MediaToolkit is a PDA based demonstration environment for the Media Engine. It is set up as a downloadable client module, with server side video processing for storage and retrieval, media analysis, and remote messaging. The system will provide a WWW interface for registration. The user selects the desired capabilities through the WWW interface and the server builds in the capabilities simulating a fee for configuration based approach and allows the user to download only the components (including drivers, etc) they need.

When executed, the interface will first provide basic capture capabilities from a menu. The drop down interface provides video preview followed by either image or video clip capture. The video clip can be stored locally or remotely, sent as a media message or set of a variety of remote processing applications on the server. Figure 6 shows the types of media processing that can be done with the media toolkit.

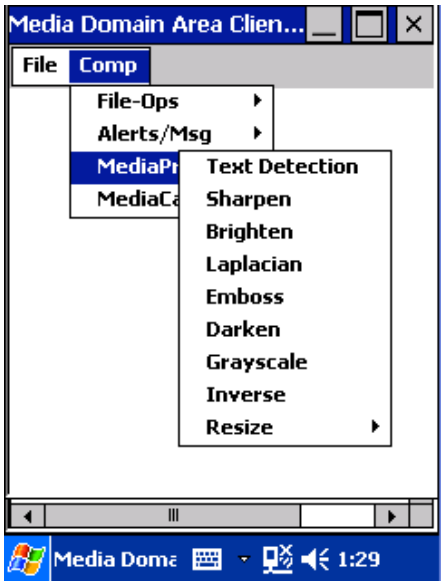


Figure 6: Media Toolkit UI

5.2 Remote Surveillance

We are also integrating our system with existing video surveillance applications. One such application is a remote video activity analysis system. Currently, a "driver" application provides a background subtraction module and a primitive classification of content [1]. The surveillance module is set up so that it records video and keyframes from periods where there is significant activity and uses simple profile measures to classify content as being a person, vehicle or unknown. The content is stored on the server.

The clients can subscribe to various surveillance alerts and then retrieve video or keyframes from the event. The frequency of the alerts and the detail provided can also be configured.

6 Extensions and future work

The focus of this system has been to provide a framework for media processing. The goal is to provide the basic capabilities so that users can focus on building applications, and not on the underlying implementation. We are in the process of integrating a number of other capabilities that will be essential for general pervasive applications. Video streaming will be added and integrated with media messaging, and alerts. In the immediate future, the service will be implemented by calling a third party server and client. Although the current system supports media objects, streaming is seen as an important, yet realizable feature which has received a great deal of attention.

The second area of future work is enhanced media processing. A great deal of work at Maryland has focused on image and video processing, including face detection and recognition, recognition of gestures, mosaicing, image stabilization, etc. The ability to provide those capabilities for media obtained from mobile devices is clearly a significant challenge. Yet as technology matures, demand for such capabilities will increase.

7 References

[1] T. Horprasert, D. Harwood, and L.S Davis. A Statistical Approach for Real-time Robust Background Subtraction and Shadow Detection . ICCV Frame Rate Vision Workshop, 1999.

[2] J. Zhang, X. Chen, J. Yang and A. Waibel. A PDA-based Sign Translator Proceedings of the 2002 International Conference on Multimodal Interfaces (ICMI), October, 2002

[3] A. Ekin and A. M. Tekalp A generic event model and sports video processing for summarization and model-based search Handbook of Video Databases (ed. Borko Furht and Oge Marques). CRC Press 2003.

[4] T. Palm, G. Schneider and S.Goose A situated computing framework for mobile and ubiquitous multimedia access using small screen and composite devices. Proceedings of the eighth ACM international conference on Multimedia, October 2000.

[5] J.Bates, D.Halls, J.Bacon A framework to support mobile users of multimedia applications Special issue on mobile computing and system services, December 1996.

[6] D.Wichadakul, X.Gu, K.Nahrstedt A programming framework for quality-aware ubiquitous multimedia applications Proceedings of the tenth ACM international conference on Multimedia, December 2002**.**

[7] D. Doermann, A. Karunanidhi, N. Parekh and V.Rautio Video Analysis Applications for Pervasive Environments Mobile Ubiquitous Multimedia, December 2002.

[8] CAPNET Context Aware Pervasive Networking (http://www.mediateam.oulu.fi/projects/capnet)

[9] Breuel, T. M. Janssen, W. C. Popat, K. Baird, H. S. Paper to PDA. International Conference on Pattern Recognition, August 2002

User- Centred Design of a Mobile Football Video Database

Alyson Evans[1]
Motorola Research Lab

Abstract

The amount of multimedia content that is available to the general population has steadily increased in recent years and this brings with it the challenge of managing large quantities of content, for both creators and consumers. The MPEG-7 standard is aimed at facilitating the management of multimedia content by providing a set of standardized tools for describing audio-visual information to support a wide range of applications. This paper describes the development of one specific application that allows consumers to access a large video database of football video footage using a mobile device. Development has followed a user-centred methodology to ensure that the application development is driven by user needs rather than by the MPEG-7 technology alone.

The process began with scenario-based design to create a broad use scenario from which initial user requirements and functional requirements were derived. Functional requirements were grouped to correspond with particular user goals and the system was modelled according to Beyer and Holtzblatt's 'user environment model' which creates a system view from the user's perspective. This model was then used as the basis for the user interface design for a mobile device. The design process is not yet complete so the paper presents the first phase of work to meet the needs of a specific group to search for sports footage using MPEG-7.

Keywords: MPEG-7, user-centred design, mobile video.

1 Introduction

In recent years large amounts of audio-visual material have become available and access to this material is becoming increasingly feasible for the general population. Numerous tools for content creation, digitization and distribution have led to this growth in the amount of audio-visual content and powerful multimedia compression standards have reduced the complexity of multimedia information. The exchange of general information also increased greatly during the 1990's due to the World Wide Web and the growth of broadband will play a further part in facilitating access for the general population to multimedia content. The development of devices that provide access for users across a range of networks is also broadening the type and amount of information content that will be available. This has brought about a critical need for tools and systems to index, search, filter and manage audio-visual content. The MPEG-7 standard for describing multimedia data is intended to promote maximum interoperability among such systems and to facilitate the creation of innovative applications in this area [1].

In the context of this need the BUSMAN[1] project (Bringing User Satisfaction to Media Access Networks) aims to design, implement, validate and trial an efficient and secure system for delivery and querying of video from large databases. This will facilitate access to large video databases for both professionals and members of the public. Although the project is considering delivery and querying across fast fixed networks and the Internet, as well as mobile environments based on GPRS and UMTS packet data communications, this paper will illustrate the development of a mobile client to provide access to video material for use by the general consumer.

The new technology that provides the key to organising and managing video on very large video databases is MPEG-7. The MPEG-7 standard, ISO 15938 [2,3,4] provides the tools to describe multimedia content. Descriptors (Ds), Description Schemes (DSs) and the Description Definition Language (DDL) enable audio-visual content to be described in a structured and detailed manner at different levels of granularity: region, image, video segment and collection. This provides support for content description, management, organization, navigation and user interaction for a wide range of applications including provision of the means for users to search and retrieve video data from very large databases.

The main research challenge in this work was to establish how to utilise the technology offered by MPEG-7 in direct response to the user needs associated with searching for and managing football video footage on a mobile device. In addition there was the challenge of converting the user's needs into a system design and a user interface design that would suit a mobile device.

There were two main areas of enquiry at the start of the project: the first associated with searching for video footage and the second related to the design challenge of designing the application for a mobile device. The research questions that were asked were:

- How will users want to search for football video material – in particular how will they initiate a search when they are mobile?
- How can the search requirements can be translated into a design for a mobile device.
- How will users want to manage the results of their searches and the collection of video footage that they will build up on a mobile device?

[1] email: alyson.evans@motorola.com

[1] The BUSMAN project is partly funded by the European Commission within the 5th Framework Programme (IST-2001-35152).

Answering these questions has been central in the work to define the system functionality and design of the user interface.

2 User- Centred Design

In its objectives the BUSMAN project emphasised the importance of a user-centred iterative design process based on ISO13407, the European standard outlining 'Human Centred Design Processes for Interactive Systems'[5]. One of the main principles of user-centred design advocated in ISO 13407 is the active involvement of users in the design process and a clear understanding of the users and task requirements.

This process acknowledges that straightforward engineering of the technology itself is not enough - the development of a new product must be set within the context of the activity that it will eventually support and related to the people who will use it. A usable system is one that is easy to learn (for novice and casual users), one that supports the user in the task or activities they are trying to achieve and does so with a level of performance that meets or exceeds a user's goals in terms of efficiency and flexibility [6].

The stages of the user-centred design process have been outlined by many authors, reflecting slight variations on the same theme, as well as in ISO 13407 itself. The key stages, outlined in Figure 1. illustrate the user-centred lifecycle adopted for BUSMAN.

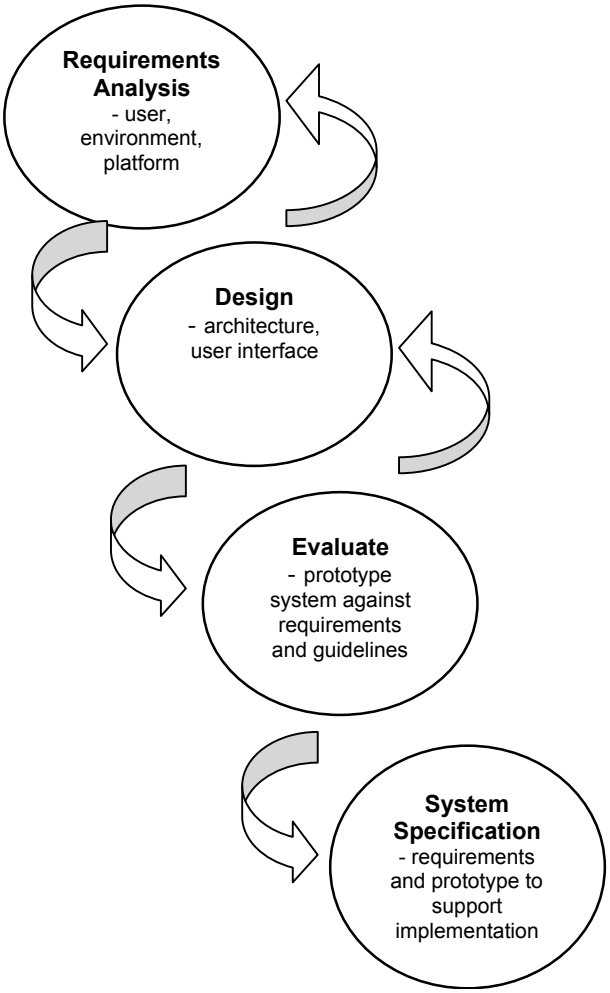


Figure 1: User-Centred Lifecycle Adopted for BUSMAN

This paper focuses on the work that has been done in the first two phases of the lifecycle - requirements analysis and how the requirements have been translated into an initial user interface design.

3 User Requirements

The first phase of the work aimed to identify an initial set of user requirements that could be used to shape the application of the MPEG-7 technology and the systems engineering that would come later in the project. The approach taken was to build up an understanding of how people currently make use of video and then to move from this starting point towards an understanding of how the new technology could be utilized to extend, make easier, make more exiting or even make entirely new ways for people to interact with video material. The aim of this phase of work was to understand the current goals and skills that potential users have in current circumstances and to combine this with what they could imagine doing in the future to make a creative leap towards new system possibilities.

This work was done by interviewing a total of 8 people (all football fans) about how they currently watch football and their use of football video footage. Interviews were a combination of 1:1 and groups interviews. The areas covered by the questions were:

- How do people currently watch football?
- How do people search for particular footage currently?
- What do people remember about a football match?
- What devices are used to watch football?
- How would people want to search for football video footage in a mobile situation if they had access to a very large database of football video footage?
- How would people decide what to watch?

It was particularly important to establish what people remember about video in order to understand what would come to mind when someone is mobile and has no reference material as a starting point for their search. Questions were also asked about mobility in general and the kind of situations in which people envisaged accessing a video database using a mobile device.

A wealth of information was gathered during these interviews. There was a significant consensus between those interviewed on what they remembered about football matches and how they would want to search for football videos from a large database. A surprising number of details were remembered about football matches, especially for the team or teams that each person supported. Football team names were remembered as were player names. People also had knowledge of which teams were playing in which leagues. More surprising was that people remembered scores for many matches, as well as controversial incidents or interesting player actions that occurred during a game. These findings provided input to the user requirements for the search criteria for the new application and helped define the annotation categories that were required within the MPEG-7 standard. The focus of users was on searching with semantic terms, so it was concluded that many of the lower level features of MPEG-7 e.g searching using colour, shape or texture, would

not apply to this specific application for the general consumer. This information was used to construct the metadata model required by the project – one that was closely related to the needs of the users in the context of searching for sports footage.

The interviews also yielded data that was used to construct a scenario of use for the new application. Scenario-based design is particularly useful when there is no pre-existing system that user's can relate to when they try to envisage how to use new technology. Creating potential use scenarios is part of a user-centred design process outlined in [7] that can be used to scope, bound and focus the requirements analysis process and to provide a connection between abstract ideas generated by potential users during interviews and a more tangible representation of a new system. The creation of a scenario provides a means of capturing, structuring and organising the ideas that people generate in a random fashion during an interview. Scenarios can also help to establish and maintain a connection between team members from different disciplines, in particular between those responsible for advocating the case of the user and those responsible for the engineering [8].Within the BUSMAN project the scenario provided a clear view of what the users thought they wanted from a mobile football video database application. The engineers in the project were able to debate the implications for the use of MPEG-7 and start to shape the use of that technology by constructing a metadata model (that outlined which parts of MPEG-7 were to be used) that was related directly to the user needs. An extract from the scenario 'Football Fans' that was generated from the initial user research is shown in Figure 2.

From this scenario and the user interviews it was possible to define an initial set of 50 user requirements. Each activity the users imagined carrying out with the new application would require specific capabilities in the new system. Hence, it was possible to identify a number of functional requirements needed to support the envisaged user activities.

4 Modelling User Activity and the New Application

The entire 'football fans' scenario contained seven individual use cases and through examination of each individual case it was possible to identify the details of exactly what the user would have to do to complete each activity, e.g. the first use case shown in Figure 2 is about finding a full length video of a football match. The user would need to carry out a number of activities to achieve this. They would need to switch on their mobile device, select the application and carry out a search of the database using a set of search criteria. Once the footage was identified the user would then need to pay for it and to watch it on their device. The user might also want to retain a link to a piece of footage for future use. This kind of analysis revealed the detailed actions required to achieve each use case and indicated further functional requirements.

To ensure that the user–centred focus was retained at this stage, before the system architecture was developed, a user–centred model of the new application functionality was developed from the use cases and the activities associated with them. The model was based on the idea of the 'user environment design' [9] and created the underlying application structure as envisaged from the user's perspective of the activities they would carry out with the application. The most logical grouping of functions, according to the different types of activities were identified and defined as activity 'focus areas'.

The 'user environment design' model has been compared with an architect's plan for a house – the architects plan shows key distinctions to support living, the 'user environment design' model shows key distinctions for supporting work practice or activity with a software system. The model is useful and powerful in the design process as it permits debate about the system structure before the design process becomes loaded with the details of the user interface design or software implementation. It is easily understood by users, human factors specialists and engineers alike. The model is used to show all the parts of a system (required for specific activities) that the user knows about and it also shows how the different parts relate to each other.

Football Fans

'Mike is traveling by train on the way to visit friends for the weekend when he decides he'd like to watch one of the more memorable games by the team he supports – Manchester United. He remembers a time when they beat Ipswich 9-0 and decides that he'd like to watch that game. He enters the team name, the opponent team's name and the match score into the BUSMAN system and it retrieves the video of that match. Mike pays for one viewing and settles down to watch the match.

Later that afternoon at his friend's house Mike watches a live football match. Manchester United beat Liverpool 3-2, with one particularly good goal by Roy Keane. Liverpool have one goal disallowed following an offside decision.

Later that evening, at a bar with a large group of friends, there is discussion about the game they watched earlier that day and they decide to watch the highlights. Using BUSMAN Mike enters the team names and chooses to see the goal highlights video. The group repeatedly watch the goal by Keane and discuss the skill it involved.

Conversation about the goal by Keane reminds Mike of a goal that Keane scored when he was playing for Nottingham Forest in the early '90s. He uses BUSMAN to search for goals scored by Keane around 8 years ago. BUSMAN returns a list of goals scored at that time. When Mike finds the one he was looking for he plays it and shows it to his friends so that they can appreciate its similarity to the one he scored earlier that day'.

Figure 2: Extract from the Use Scenario 'Football Fans'

For the mobile football video database four 'focus areas' were identified from the scenario and user requirements. One contained searching type activities like choosing search criteria, entering them into the application and submitting a search. Another focus area was concerned with viewing video and the play, stop, rewind and fast forward functions associated with that. The third focus area related to the management of a collection of footage that a user might build up over time. The final focus area related to payment for the footage.

The mobile application to search retrieve and view sports footage was modelled as shown in Figure 3 below, based on the 'football fans' scenario and the analysis of more detailed activities. The model provides a structure of user activity that formed the basis of user interface design and also influenced the development of the software architecture.

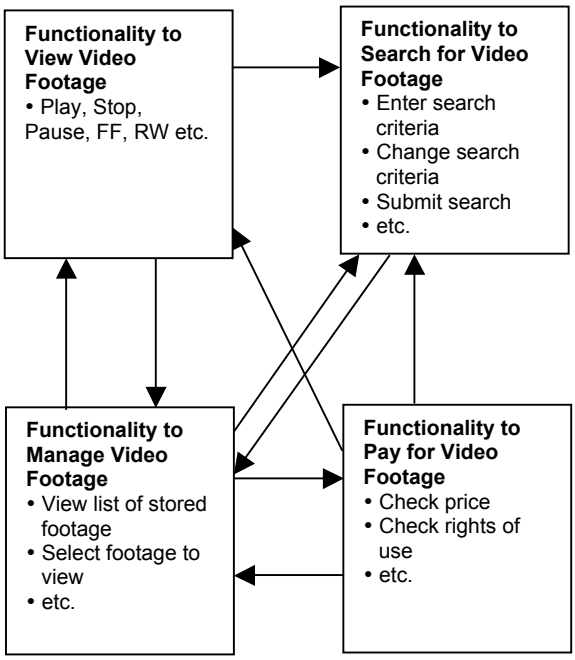


Figure 3: Conceptual Model for system to search, retrieve and view video footage

The BUSMAN conceptual model was used by engineers in the project to plan the structure of the system architecture. It provided the engineers with an early insight into what users wanted from the system and how the system functionality needed to connect. The conceptual model provided a vital connection between user needs and the system architecture at a stage in the lifecycle prior to implementation. This modelling allowed useful debate to occur before a commitment was finally made to the architecture to be implemented. The conceptual model also supports design by providing the underlying structure of an application before detailed ideas about input and outputs and information display are conceived.

5 Designing the User Interface from the Conceptual Model

Once the detailed functionality was meaningfully grouped into the focus areas defined by the conceptual model the user interface was designed with direct reference to it. Four main interface areas were defined – each corresponding to one of the focus areas. One area focused on searching activity – inputting search criteria, submitting the search as well as display of search results. Another area of the UI was devoted to playing the video – this required maximum display area for viewing plus the usual controls associated with playing and reviewing video. The third distinct area of the UI was for paying for video footage – this requires activities like inputting user identification information and bank account details – similar to the UI requirements for Internet shopping. The fourth area of the UI was devoted to management of the video footage. It is likely that a user will have access to a variety of footage at one time and they will need to be able to access it directly, without having to run a search each time. So an area was dedicated to contain the links to 'accessible footage', to 'favourite footage' and to footage identified as 'possibilities' to potentially be watched at a later date.

Figure 4 shows the screen of the user interface through which the user enters search criteria. Each search criterion was suggested by users in the interviews at the start of the project. Each criterion is a button which links to another page that presents the user with a range of possible criterion choices. The aim is to allow the user to enter a set of criteria for their search using the smallest number of key presses. The video footage that will be accessed using the mobile device will be annotated using the new MPEG-7 standard to correspond with this set of criteria

The links between the different areas of the UI shown in the conceptual model are implemented through the use of the buttons contained in the application navigation bar, which remains on display at the foot of almost every screen of the UI. From the search page the user can connect to the collection of videos that they have built up. No link is present here to 'play video' because there are no videos to choose.

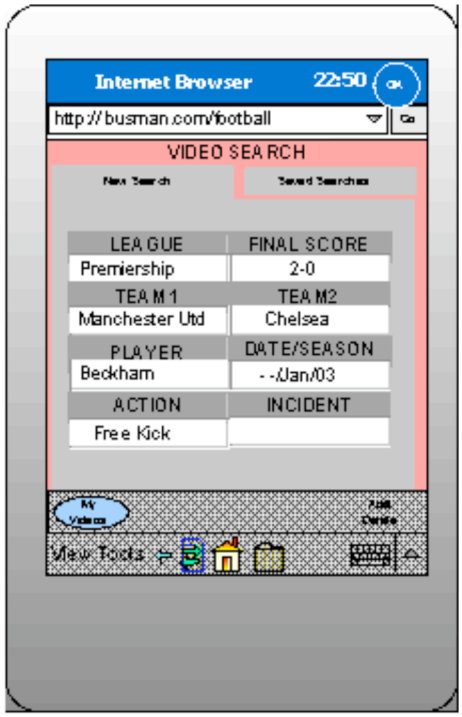


Figure 4: User Interface to enter search criteria

The UI area dedicated to managing video footage is shown in Figure 5. This is distinct in style from the UI used to enter

search criteria and provides the means for the user to manage links to different pieces of footage. Footage has been categorised into that which is already accessible as it has been paid for, other footage has been identified by the user as 'possibly' being of interest. A third category allows the user to keep a list of their favourite footage.

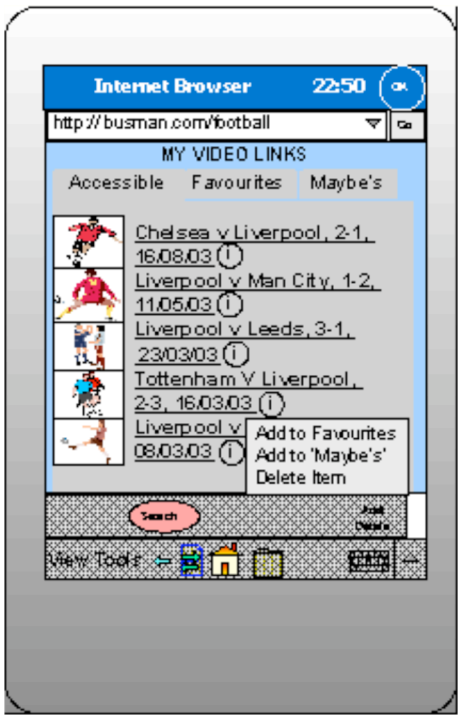


Figure 5: User Interface to Manage Video Footage

These pages are shown to illustrate how the conceptual model can provide a logical basis for structuring the user interface. Understanding the structure of the UI before individual pages of the UI are designed saves valuable time in what can be a somewhat confusing process if a strictly top down approach is taken.

The BUSMAN project has progressed as far as designing the first iteration of the user interface. The next stage in the user-centred process is to evaluate the prototype system in terms of its functionality, the structure of the functionality and the user interface itself. The first stage will be an 'expert evaluation' that will utilise guidelines on user interface design from ISO standards and other sources. The user interface will be re-designed following this evaluation and will then be subjected to at least 2 rounds of user testing with real users. The feedback gained from these evaluations will be used for further re-design of the user interface as well as development of the user requirements – these will be added to and refined and the evaluation phase progresses.

Hence, the entire user-centred design process illustrated in Figure 1 will be followed, and at the end of the project the aim is to have a comprehensive and relevant set of user requirements plus a prototype system that can be used to demonstrate the concept of searching for football videos in a large video database when the user is mobile.

6 Conclusion

The BUSMAN project set out to make best use of MPEG-7 technology on behalf of general consumers who, in the future, will be able to access video footage when they are on the move. By adopting a user-centred focus from the outset user needs were captured and used as a reference point as each step in the development lifecycle took place. By making use of two user-centred techniques – scenario development and the conceptual model the user needs were not simply noted at the start of the project but were integrated into the design of the system architecture and user interface. The user-centred techniques were new to many of the engineers on the project but their value has been appreciated as its output provided tangible reference points that were used to develop technology and system architectures. Although it is possible to argue that great leaps in the development of new technology could be constrained by paying too much heed to the conservative needs of its potential users, the methods outlined here illustrate how the user needs can be presented and understood by the whole project team in a way that permits informed debate between the technologists and those representing user needs in a way that allows intelligent decisions to be made about utilisation of new technology.

References

[1] Chang, Shih-Fu., Sikora, Thomas., and Puri, Atul. Overview of the MPEG-7 Standard. IEEE Transactions on Circuits and Systems for Video Technology, Vol 11, No 6, June 2001.

[2] Martinez, J.M (Ed). Overview of the MPEG-& Standard. ISO/IEC JTC1/ SC29/WG11 N4509, Dec 2001.

[3] ISO/IEC, 15938-3/FDIS, Information technology – Multimedia content description interface – Part 3 Visual, ISO/IEC, Oct. 2001.

[4] ISO/IEC, 15938-5/FDIS, Information technology – Multimedia content description interface – Part 5 Multimedia Description Schemes, ISO/IEC, Oct. 2001.

[5] ISO 13407. Human Centred Design Processes for Interactive Systems.

[6] Mayhew, D.J. The Usability Engineering Lifecycle. Academic Press, Morgan Kaufmann, 1999.

[7] Carroll, John.M., Making Use; Scenario Based Design of Human Computer Interactions. MIT Press 2000.

[8] McGraw, K. and Harbison, K. User Centred Requirements: The Scenario Based Engineering Process. Lawrence Erlbaum Associates Inc. 1997.

[9] Beyer, H. and Hottzblatt, K. Contextual Design: Defining Customer Centred Systems. Morgan Kaufmann. 1998.

Faces Everywhere:

Towards Ubiquitous Production and Delivery of Face Animation

Igor S. Pandzic[1], Jörgen Ahlberg[2], Mariusz Wzorek[2], Piotr Rudol[2] , Miran Mosmondor[1]

[1]Department of Telecommunications

Faculty of electrical engineering and computing

Zagreb University

Unska 3, HR-10000 Zagreb, Croatia

{Igor.Pandzic, Miran.Mosmondor}@fer.hr

[2]Visage Technologies AB

Tröskaregatan 90, SE-58334 Linköping, Sweden

www.visagetechnologies.com

{jorgen, mariusz, piotr}@visagetechnologies.com

Abstract

While face animation is still considered one of the toughest tasks in computer animation, its potential application range is rapidly moving from the classical field of film production into games, communications, news delivery and commerce. To support such novel applications, it is important to enable production and delivery of face animation on a wide range of platforms, from high-end animation systems to the web, game consoles and mobile phones. Our goal is to offer a framework of tools interconnected by standard formats and protocols and capable of supporting any imaginable application involving face animation with the desired level of animation quality, automatic production wherever it is possible, and delivery on a wide range of platforms. While this is clearly an ongoing task, we present the current state of development along with several case studies showing that a wide range of applications is already enabled.

Keywords: face animation, virtual characters, embodied conversational agents, visual text-to-speech, face tracking, lip sync, MPEG-4 FBA

1 Introduction

The human face is one of the most expressive channels of communication and appears in multimedia contents so universally that we take it for granted. Researchers have been fascinated with the possibility to recreate and animate human-like faces on computers since decades [1]. Early face animation research proposed various models for animating a 3D face model: procedural [8], pseudo-muscle [9] and muscle simulation [10][11] were the main categories. More recently, researchers worked on more realistic face models [12][13][14][24]. In parallel, work progressed on face animation production methods such as visual text-to-speech [6], automatic lip-sync [17], and face feature tracking in video [26][27][28].

Application areas for face animation are expanding from film production and advertising into such diverse areas as games, teleconferencing, messaging [25], news delivery [16], education and commerce [7]. In particular, research on Embodied Conversational Agents [15] is going towards the notion of human-like user interface that we can simply talk to – applications of such technology could be very wide in all kinds of automated services, support, consulting and more.

After three decades of research on face animation, most developed systems are still proprietary and do not talk to each other. It is rare, for example, that a lip sync system from one research group or company can directly drive a muscle-based animated face from another group. Yet this kind of compatibility, together with widespread support on various platforms, is essential for widespread applications. The same face animation content, regardless how it was produced, should be playable on platforms and systems as diverse as mainstream 3D animation tools, PCs, games consoles, set-top boxes and mobile phones (Figure 1).

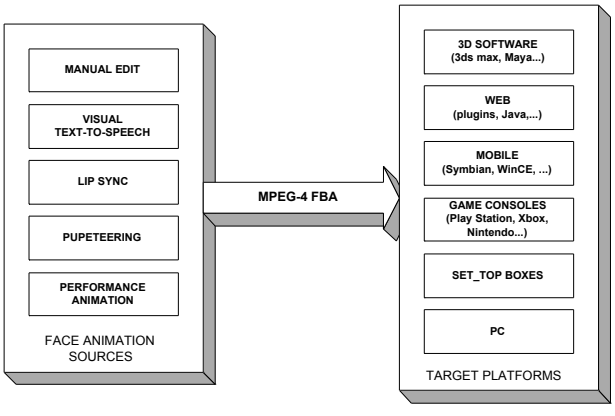


Figure 1: Portability of face animation

This kind of widespread portability is essentially made possible by the recent MPEG-4 Face and Body Animation (FBA) standard [3][4]. In our work, we take this idea one step forward and build a working set of tools that make this promise a reality: the visage framework. It is a set of software

components interfaced through standards such as MPEG-4 and VRML. Subsets of these components are assembled together, or with other 3rd party standard components, to deliver a wide range of applications based on face animation on various platforms. We believe that most currently considered applications can be successfully delivered in this way.

We present an overview of the visage framework in section 2, and describe the various components in sections 1 - 5. Section 6 showcases several case studies that demonstrate the versatility of the framework. The final section brings conclusions and future work ideas.

2 Overview of the visage framework

The visage framework consists of three major categories of modules: face model production, face animation production and multi-platform delivery. Figure 2 shows all modules. In a typical application only a selection of modules is used (see case studies, section 6).

Making a face model and preparing it for animation is typically time consuming. In the visage framework, static face models are imported in standard VRML format from mainstream modelling tools and prepared for animation using the semi-automatic Facial Motion Cloning method. This method essentially copies all necessary morph targets from an existing generic face model to the new face model. An interesting feature is the possibility to make available morph target sets with different animation styles for the generic model, and simply choose which animation style to clone onto the new model (e.g. standard, exaggerated, with wrinkles,…).

The visage framework contains tools for face animation production based on most currently known methods: video tracking, visual TTS, lip sync and manual editing. Each tool will be described in more detail in section 4. All tools produce standard MPEG-4 FBA bitstreams with face animation, making the tools completely interchangeable. Thanks to the standard format, the editing tool can be applied on the results of all other tools, and 3rd party standard tools can easily be incorporated. A particular advantage of the MPEG-4 FBA format is its efficiency – bit rates can be as low as 0.5 kbit/sec if only viseme-based speech animation is used, and typically do not exceed 5 kbit/sec for full animation.

The delivery is based on the very small and portable visage Face Animation Player core. This core exists in both Java and C++ versions, and can easily be ported on top of any software environment supporting 3D graphics, as illustrated in Figure 2.

By selecting appropriate tools, it is fairly straightforward to build applications involving face animation in various environments and platforms.

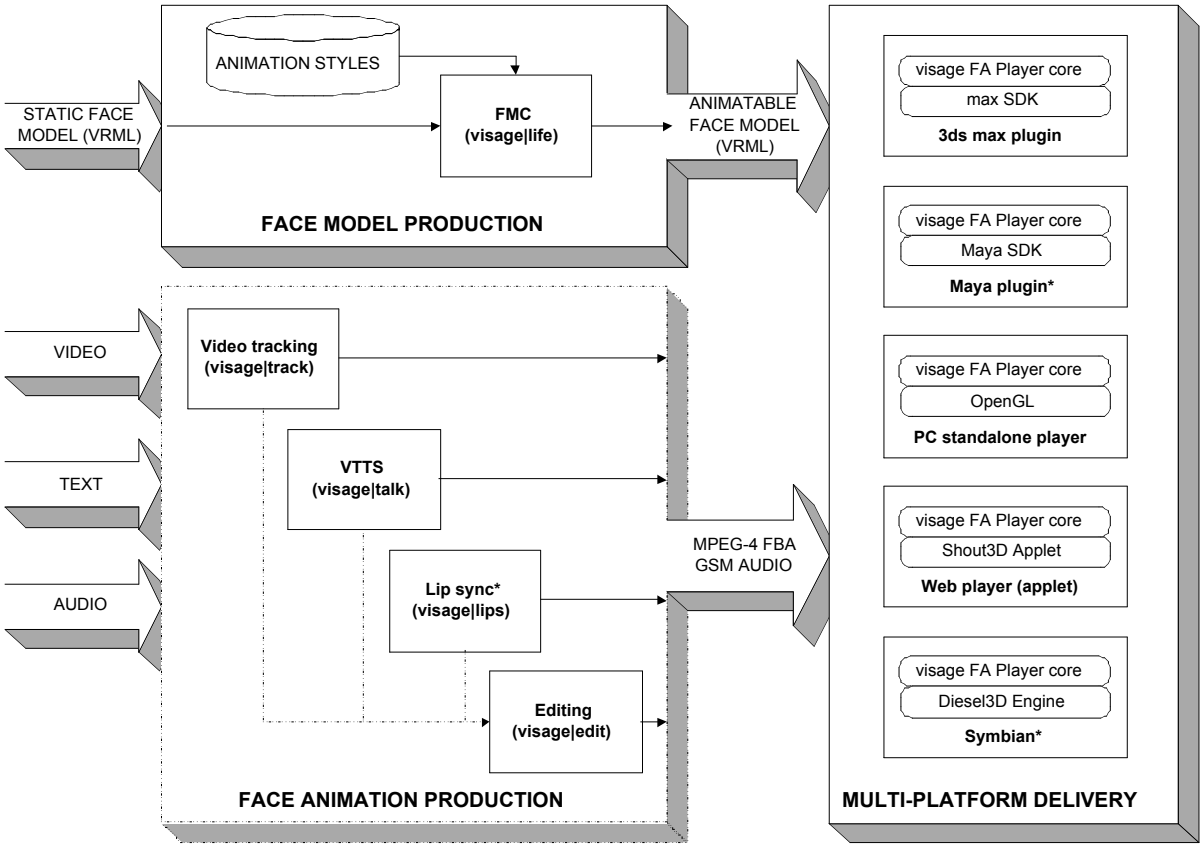


Figure 2: Overview of the visage framework (*currently under development)

3 Face model production

In this section we describe our approach to the production of face models that can be directly animated by all other tools in the described framework. We believe that the most important requirement for achieving high visual quality in an animated face is the openness of the system for visual artists. It should be convenient for them to design face models with the tools they are used to. The concept of morph targets as key building blocks of facial animation is already widely used in the animation community. However, morph targets are commonly used only for high level expressions (visemes, emotional expressions). In our approach we follow the MPEG-4 FAT concept and use morph targets not only for the high level expressions, but also for low-level MPEG-4 FAPs. Once their morph targets are defined, the face is capable of full animation by limitless combinations of low-level FAPs.

Obviously, creating morph targets not only for high level expressions, but also for low-level FAPs is a tedious task. We therefore propose a method to copy the complete range of morph targets, both low- and high-level, from one face to another. The source face with a complete set of morph targets is available, and different sets of morph targets defining various animation styles are being developed, so that a user can choose the animation style to be applied when copying the morph targets to a new face. The method we propose for copying the morph targets is called Facial Motion Cloning. Our method is similar in goal to the Expression Cloning [2]. However, our method additionally preserves the MPEG-4 compatibility of cloned facial motion and it treats transforms for eyes, teeth and tongue. It is also substantially different in implementation.

Facial Motion Cloning can be schematically represented by Figure 3. The inputs to the method are the source and target face. The source face is available in neutral position (*source face*) as well as in a position containing some motion we want to copy (*animated source face*). The target face exists only as neutral (*target face*). The goal is to obtain the target face with the motion copied from the source face – the *animated target face*.

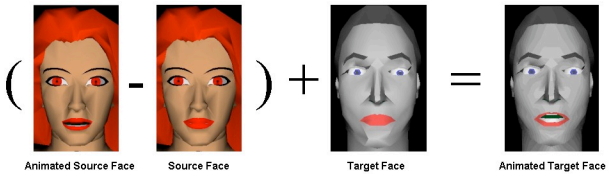


Figure 3: Overview of Facial Motion Cloning

To reach this goal we first obtain *facial motion* as the difference of 3D vertex positions between the animated source face and the neutral source face. The facial motion is then added to the vertex positions of the target face, resulting in the animated target face. In order for this to work, the facial motion must be normalized, which ensures that the scale of the motion is correct. In the *normalized facial space*, we compute facial motion by subtracting vertex positions of the animated and the neutral face. To map the facial motion correctly from one face to another, the faces need to be aligned with respect to the facial features. This is done in the *alignment space*. Once the faces have been aligned, we use interpolation to obtain facial motion vectors for vertices of the target face. The obtained facial motion vectors are applied by adding them to vertex positions, which is possible because we are working in the normalized facial space. Finally, the target face is de-normalized. The procedure is repeated for all morph targets we want to copy. The Facial Motion Cloning method is described with more detail in [5].

4 Face animation production

4.1 Video tracking

Tracking a face, and facial features like lip movements etc, in video is a simple task for the human visual system, but has shown to be a very complex problem for machine vision. There are numerous proposed methods, but quite few have so far reached the market. Our method is based on the Active Appearance Models [29], and offers 3D tracking of the face and important facial features (currently lip and eyebrow motion) in real-time or near real-time. The method is based on a statistical model of facial appearance, finding the most probable pose and deformation (i.e. facial feature motion) of the face in each frame. Technical details can be found in [26]. The current version of the tracking module typically needs to be trained for the person to be tracked, but this step is expected to be removed.

An illustration of the tracking is given in Figure 4. As can be seen, the simple face model used here is automatically adapted to the face in each frame. From the pose and deformation of the model, MPEG-4 FAPs can be computed.

The tracking module inputs a sequence of images and outputs animation parameters in an MPEG-4 FBA bitstream describing the head, lip and eyebrow motion. The animation parameters can then be applied to a model of the recorded face, potentially enabling very-low bitrate video telephony, or any other face model. Another usage is to record head motion to be used as "background motion", to which lip motion from the VTTS could be added, giving the final animation a realistic look.

Figure 4: Automatic face and facial feature tracking. The simple face model adapts to each frame in the sequence (every tenth frame shown), and MPEG-4 FAPs can then be extracted from the model.

4.2 Visual text-to-speech

The visual text-to-speech tool is based on the SAPI standard (SAPI-4 and SAPI-5 versions exist). The SAPI TTS generates events when phonemes are generated and provides timing of the phonemes. Tables are used to convert phonemes into MPEG-4 visemes, and these are encoded into an MPEG-4

FBA bitstream. In the current version co-articulation model is a simple one, using linear interpolation to blend visemes on the boundaries. Non-SAPI TTS systems can be integrated through a simple interface. Any TTS system that can generate phoneme/viseme timing can be connected, and the output is the standard MPEG-4 FBA bitstream.

4.3 Lip sync

A module for automatic lip sync is currently under development. The module inputs digitized speech and, like the VTTS module described above, outputs visemes in an MPEG-4 FBA bitstream. It is based on a method that extracts Mel Frequency Cepstral Coefficients from the audio, and then uses a set of neural networks to classify each audio frame as a certain phoneme. The module can operate in two modes: real-time and batch. In real-time mode, received audio can be played and animated in real-time with a delay of less than 80 ms. In batch mode, the delay is significantly higher, but offers a somewhat higher quality of the lip sync. Technical details will be published in the near future.

4.4 Animation editing

MPEG-4 Facial Animations can be easily manipulated using the visage|edit application. Using this program, animations can be merged or created from scratch. The user interface consists of four panels shown in Figure 5.

The first from the left is the Project Tree panel used for hierarchical selection. The first node of the tree is a virtual track. This track constitutes a reference for the project being edited. Additional nodes represent tracks included in the project. Next is the High-Level Editing panel which shows information in rows according to items visible in the Tree View. Each track in a project is represented by a bar, which can be moved (changes offset) and resized (changes scale in time) using mouse. Additional controls for choosing track manipulations are supplied. The Low-Level Editing panel consists of a plot area for displaying FAP values.

This plot can be scrolled using scroll bars or by "dragging" the plot area with Shift key pressed. There are two directions of zooming. Vertical zoom allows setting the range of FAP values to be shown. Horizontal zoom allows setting the range of frames to be displayed. The last panel is the Preview panel which shows a face model at all time for a user to see changes being done to the project animation.

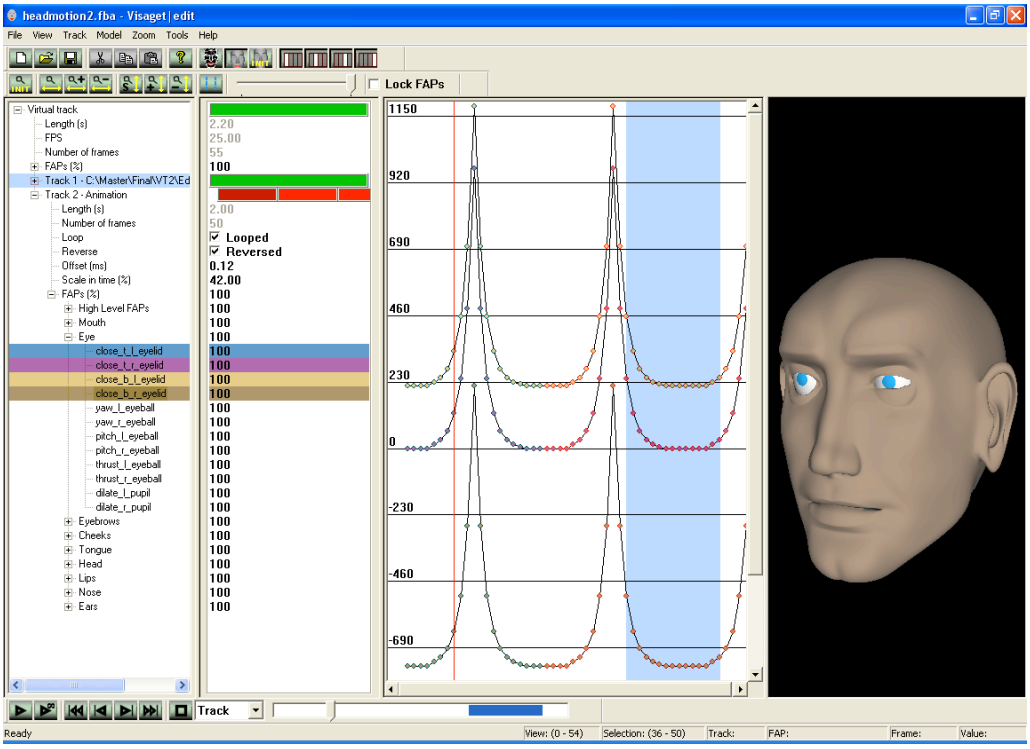


Figure 5: visage|edit application.

In order to enable users to manage projects consisting of several animations, the idea of a tree view was introduced. It makes it possible to display information about several tracks on different levels of detail at the same time. Additionally, the tree is a common reference for both high- and low-level panels – information shown in the two latter views corresponds to options chosen in the former.

The high-level mode allows editing tracks without going into details of FAP values. This mode can be used to manipulate already existing tracks. The obvious example of this mode is merging two animations in order to add eye blinks to a track obtained from the Visage|track application. Operations that can be performed in the high-level mode include adding offset, scaling, looping and reversing the track and additionally this mode allows applying multiplication factors on all FAPs, groups of FAPs and, on separate FAPs. The final value of the multiplication factor for a separate FAP is the product of the three multiplication factors and a corresponding factor from the "virtual track".

The low-level mode allows direct manipulation of FAP values. This mode is useful for fine-tuning the parameters of an animation and when "producing" new animation tracks, for example creating an eye blink track from scratch. Such an eye-blink can be looped to produce the eyelids' movement for the whole animation.

5 Multi-platform delivery

Multi-platform delivery, and the capability to implement support for virtually any platform in a very short time, is one of the strongest points of the visage framework. The strategy to achieve this is to use a bare-minimum face animation player core. This core can be easily ported to any platform that supports 3D graphics.

The player is essentially an MPEG-4 FBA decoder. When the MPEG-4 Face Animation Parameters (FAPs) are decoded, the player needs to apply them to a face model. Our choice for the facial animation method is interpolation from key positions, essentially the same as the morph target approach widely used in computer animation and the MPEG-4 FAT approach. Interpolation was probably the earliest approach to facial animation and it has been used extensively. We prefer it to procedural approaches and the more complex muscle-based models because it is very simple to implement, and therefore easy to port to various platforms; it is modest in CPU time consumption; and the usage of key positions (morph targets) is close to the methodology used by computer animators and should be easily adopted by this community.

The way the player works is the following. Each FAP (both low- and high-level) is defined as a key position of the face, or morph target. Each morph target is described by the relative position of each vertex with respect to its position in the neutral face, as well as the relative rotation and translation of each transform node in the scene graph of the face. The morph target is defined for a particular value of the FAP. The position of vertices and transforms for other values of the FAP are then interpolated from the neutral face and the morph target. This can easily be extended to include several morph targets for each FAP and use a piecewise linear interpolation function, like the FAT approach defines. However, current implementations show simple linear interpolation to be sufficient in all situations encountered so far. The vertex and transform movements of the low-level FAPs are added together to produce final facial animation frames. In case of high-level FAPs, the movements are blended by averaging, rather than added together.

Due to its simplicity and low requirements, the face animation player is easy to implement on a variety of platforms using various programming languages (Figure 2). For example, the Java applet implementation, based on the Shout3D rendering engine [18], shows performance of 15-40 fps with textured and non-textured face models of up to 3700 polygons on a PIII/600MHz, growing to 24-60 fps on PIII/1000, while the required bandwidth is approx 0.3 kbit/s for face animation 13 kbit/s for speech, 150K download for the applet and approx. 50K download for an average face model. This performance is satisfactory for today's mobile PC user connecting to the Internet with, for example, GPRS. More details on this implementation and performances can be found in [19]. Other implementations include a PC standalone version based on OpenGL, 3ds max and Maya plugins and an implementation on a Symbian platform for mobile devices (last two currently in development).

Implementation of the face animation player on Symbian platform for mobile devices is written as C++ application and based on DieselEngine [31]. Because of low CPU time consumption and low memory requirements, MPEG-4 FBA decoder can be used almost unaltered on mobile device. Most differences were concerning rendering 3D graphics on mobile device. For that purpose DieselEngine was used. It is collection of C++ libraries that helps building applications with 3D content on various devices. DieselEngine has a low level API (Application Program Interface) that is similar to Microsoft DirectX and high level modules had to be implemented. The most important is the VRML parser that is used to convert 3D animatable face model from VRML format to Diesel3D scene format (DSC). Other modules enable interaction with face model like navigation, picking and centering. We have tested this implementation on Sony Ericsson P800 mobile device with various static face models. Interactive frame rates were achieved with models containing up to 3700 polygons.

6 Case studies

The classical usage of the presented framework would be the film/video production. In this scenario, the face model is prepared using the FMC module, one or more animation production modules are used to prepare the face animation, and then the model and the animation are imported into a mainstream 3D animation package (e.g. 3ds max or Maya), incorporated into a scene and rendered. However, this is just one application scenario and there are many other potential applications that can be built based on the visage framework. In the next sections we will briefly describe several experimental applications that have been implemented.

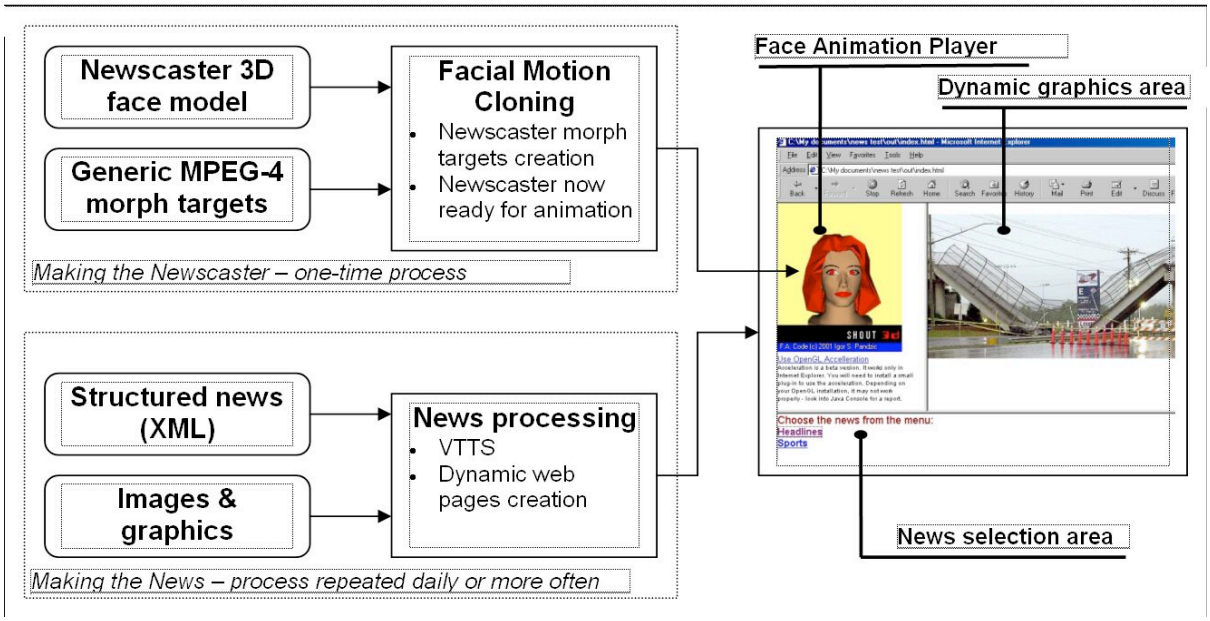


Figure 6: The virtual newscaster system architecture

6.1 Virtual newscaster

We have built a prototype of an interactive multimedia news system featuring a talking virtual character to present the news on the Web [16]. The virtual character is used as a newscaster, reading the news on the Web while at the same time presenting images and graphics. The users choose the news topics they want to hear. The content of the news is defined in an XML file, which is automatically processed to create the complete interactive web site featuring the virtual newscaster reading out the news. This allows for very frequent automatic updates of the news. The virtual character is animated on the client using a Java applet implementation of the visage face animation player, requiring no plug-ins. The bandwidth and CPU requirements are very low and this application is accessible to a majority of today's Internet users without any installation on the end-user computer. We believe that the presented news system combines qualities of other current news delivery systems (TV, radio, web sites) and therefore presents an attractive new alternative for delivering the news.

Figure 6 illustrates how components of the visage framework (Facial Motion Cloning, VTTS, Face Animation Player) are used together with application-specific components to deliver this application.

6.2 Talking email/SMS

Talking email combines a VTTS module on the server with the web version of the face animation player. It is a web application that allows the visitors of a web site to compose and send talking email straight from the web site, offering a great entertainment value. A talking email is a web page containing an interactive virtual person that talks, i.e. pronounces the email message. The sender inputs the message text and chooses the virtual person to deliver it. The speech and animation are generated on the server and the sender can immediately preview the talking email message, then simply input the email address and send it. The receiving party sees the talking virtual character in a web page delivered by email.

The SMS interface allows sending talking email messages from a mobile phone by SMS. Current development is going towards the delivery of the talking email directly on mobile phones, either through MMS or through full face animation player application running on the mobile phone.

6.3 Embodied Conversational Agents

Embodied Conversational Agents are virtual characters coupled with artificial intelligence (AI) techniques to deliver the impression of a live person that can lead a meaningful conversation. Such virtual characters are expected to represent the ultimate abstraction of a human-computer interface, the one where the computer looks, talks and acts like a human.

This would include audio/video analysis and synthesis techniques coupled with AI, dialogue management and a vast knowledge base in order to be able to respond quasi-intelligently to the user – by speech, gesture and even mood [22]. While this goal lies further on in the future, we present an architecture that reaches towards it, at the same time aiming for a possibility of practical applications in nearer future. Our architecture is aimed specifically at the Web.

Our system uses A.L.I.C.E. [20] as the AI server. It is based on Case-Based Reasoning or CBR, first used by ELIZA [21]. The AI server takes text as input, and outputs reasonable answers in form of text based on the A.L.I.C.E. knowledge base. The user interface is a web page with an integrated virtual character and text input field (Figure 7). When the user enters some text, it is sent to the server where it is first processed by the AI server in order to obtain an answer from the AI engine. The answer is then sent to the VTTS module which generates the speech and appropriate face animation; these are returned to the client and played.

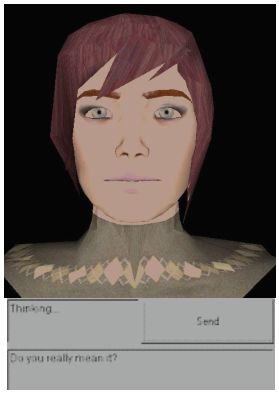


Figure 7: Embodied Conversational Agent

7 Conclusions and future work

We have introduced a framework for ubiquitous production and delivery of face animation and presented case studies showing how this framework can be configured into various applications. Work is ongoing on the development of new modules of the framework, specifically to support new platforms and improve the existing modules. In particular, the VTTS module is being extended with an automatic gesturing function that should produce natural-looking facial gestures based on lexical analysis of the text and a statistical model based on analysis of a training data set, similar to [23]. In parallel, new applications are being developed, in particular on mobile phones where we expect such innovative applications to have a great entertainment value.

8 References

[1] F.I. Parke, K. Waters: "Computer Facial animation", A.K.Peters Ltd, 1996., ISBN 1-56881-014-8

[2] Jun-yong Noh, Ulrich Neumann: "Expression Cloning", Proceedings of SIGGRAPH 2001, Los Angeles, USA.

[3] ISO/IEC 14496 - MPEG-4 International Standard, Moving Picture Experts Group, www.cselt.it/mpeg

[4] Igor S. Pandzic, Robert Forschheimer (editors): "MPEG-4 Facial Animation - The standard, implementations and applications", John Wiley & Sons, 2002, ISBN 0-470-84465-5.

[5] Igor S. Pandzic: "Facial Motion Cloning", accepted for publication in the Graphical Models journal.

[6] C. Pelachaud, "Visual Text-to-Speech", in "MPEG-4 Facial Animation - The standard, implementations and applications", I.S. Pandzic, R. Forchheimer (eds.), John Wiley & Sons, 2002.

[7] J. Ostermann, D. Millen, "Talking heads and synthetic speech: An architecture for supporting electronic commerce", Proc. ICME 2000

[8] F.I. Parke: "A Parametric Model for Human Faces", PhD Thesis, University of Utah, Salt Lake City, USA, 1974. UTEC-CSc-75-047.

[9] Kalra P., Mangili A., Magnenat-Thalmann N., Thalmann D.: "Simulation of Facial Muscle Actions based on Rational Free Form Deformation", Proceedings Eurographics 92, pp. 65-69.

[10] S.M. Platt, N.I. Badler: "Animating Facial Expressions", Computer Graphics, 1981, 15(3):245-252.

[11] K. Waters: "A muscle model for animating three-dimensional facial expressions", Computer Graphics (SIGGRAPH'87), 1987, 21(4):17-24.

[12] Y. Lee, K. Waters, D. Terzopoulos: "Realistic modeling for facial animation", in Computer Graphics (SIGGRAPH '95 Proceedings), 55-62

[13] B. Guenter, C. Grimm, D. Wood: "Making Faces", in Computer Graphics (SIGGRAPH '98 Proceedings), 55-66

[14] V. Blanz, T. Vetter: "A morphable model for the synthesis of 3D faces" , in Computer Graphics (SIGGRAPH '99 Proceedings), 75-84

[15] Embodied Conversational Agents, edited by Cassell J., Sullivan J., Prevost S., Churchill E., The MIT Press Cambridge, Massachusetts London, England, 2000.

[16] "An XML Based Interactive Multimedia News System", Igor S. Pandzic, 10th International Conference on Human - Computer Interaction HCI International 2003, Crete, Greece

[17] M. Brand, "Voice Puppetry", Proceedings of SIGGRAPH'99, 1999.

[18] Shout 3D, Eyematic Interfaces Incorporated, http://www.shout3d.com/

[19] Igor S. Pandzic: "Facial Animation Framework for the Web and Mobile Platforms", Proc. Web3D Symposium 2002, Tempe, AZ, USA, demonstration at www.tel.fer.hr/users/ipandzic/MpegWeb/index.html

[20] Artificial Linguistic Internet Computer Entity, http://www.alicebot.org

[21] Weizenbaum, J., "ELIZA - A computer program for the study of natural language communication between man and machine", Communications of the ACM 9(1): 36-45, 1966.

[22] The InterFace project, IST-1999-10036, www.ist-interface.org

[23] S. P. Lee, J. B. Badler, N. I. Badler, "Eyes Alive", Proceedings of the 29th annual conference on Computer graphics and interactive techniques 2002 , San Antonio, Texas, USA, ACM Press New York, NY, USA, Pages: 637 – 644

[24] Eric Cosatto, Hans Peter Graf: Photo-Realistic Talking-Heads from Image Samples. IEEE Transactions on Multimedia 2 (3): 152-163 (2000)

[25] J. Ostermann, "PlayMail: The Talking Email", in "MPEG-4 Facial Animation - The standard, implementations and applications", I.S. Pandzic, R. Forchheimer (eds.), John Wiley & Sons, 2002.

[26] J. Ahlberg and R. Forchheimer, "Face Tracking for Model-based Coding and Face Animation," Int Journal of Imaging Systems and Technology, 13(1):8-22, 2003.

[27] M. La Cascia, S. Sclaroff, and V. Athitsos, "Fast, reliable head tracking under varying illumination: An approach based on registration of texture-mapped 3D models," IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(4):322-336, 2000.

[28] C. S. Wiles, A. Maki, and N. Matsuda, "Hyperpatches for 3D Model Acquisition," IEEE Transactions on Pattern Analysis and Machine Intelligence, 23(12):1391-1403, 2001.

[29] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active appearance models," IEEE Transactions on Pattern Analysis and Machine Intelligence, 23(6):681-685, 2001.

[30] V. Mäkinen, Front-end feature extraction with mel-scaled cepstral coefficients, technical report, Laboratory of Computational Engineering, Helsinki University of Technology, September 2000.

[31] http://www.inmarsoftware.com/diesel.htm

Designing Gestures for Affective Input: An Analysis of Shape, Effort and Valence

Petra Fagerberg*, Anna Ståhl*, Kristina Höök
Stockholm University/KTH
DSV
Forum 100
164 40 Kista
* Authors in alphabetical order

Abstract

We discuss a user-centered approach to incorporating affective expressions in interactive applications, and argue for a design that addresses both body and mind. In particular, we have studied the problem of finding a set of affective gestures. Based on previous work in movement analysis and emotion theory [Davies, Laban and Lawrence, Russell], and a study of an actor expressing emotional states in body movements, we have identified three underlying dimensions of movements and emotions: *shape, effort* and *valence*. From these dimensions we have created a new affective interaction model, which we name *the affective gestural plane model*. We applied this model to the design of gestural affective input to a mobile service for affective messages.

Keywords: Affective interaction, gestures, user-centered design, mobile service

1 Introduction

By addressing human emotions explicitly in the design of interactive applications, the hope is to achieve both better and more pleasurable and expressive systems. The work presented in here is inspired by the field of *affective computing* [Paiva, Picard], even if our aim is to take a slightly different stance towards how to design for affect than normally taken in that field – a more user-centered approach.

Affective computing, as discussed in the literature, is computing that relates to, arises from, or deliberately influences emotions [Picard]. The most discussed and spread approach in the design of affective computing applications is to construct an individual cognitive model of affect from first principles and implement it in a system that attempts to recognize users' emotional states through measuring biosignals. Based on the recognized emotional state of the user, the aim is to achieve an as life-like or human-like interaction as possible, seamlessly adapting to the user's emotional state and influencing it through the use of various affective expressions [e g Ark et al., Fernandez et al.]. This model has its limitations [Höök], both in its basic need for simplification of human emotion in order to model it, and its difficult approach into how to infer the end-users emotional states through various readings of biosignals.

To get the users involved in a more active manner we would, instead, like to propose the user-centered approach to affective computing. Our aim is to have users consciously expressing their emotions rather than having their emotions interpreted or influenced by the system, while still maintaining the mystery and open interpretation of emotional interaction and expression. Inspired by the results of our previous work [Paiva et al.] we arrived at a set of four design principles, outlined in detail below: *embodiment* as a means to address physical and cognitive concepts in the interaction with the application [Dourish], *natural but designed expressions* as a means to communicate affect instead of aiming for complete naturalness, an *affective loop* to reach emotional involvement with both body and mind, and *ambiguity* of the designed expressions [Gaver et al.] to allow for open-ended interpretation by the end-users instead of simplistic, one-emotion one-expression pairs.

Our specific focus in this paper is to describe the process of finding affective gestures for interacting with a mobile service. Our idea is that gestures will address the body-part of emotions in people. When placed in an interaction that also speaks to our mind, the result may be an increased sense of actually communicating affect. Based on previous work in movement analysis [Davies, Laban and Lawrence], emotion theory building upon people's everyday understanding of emotion states [Russell], and a study of an actor expressing emotional states in body movements, we identified three underlying dimensions of movements and emotion: *shape, effort* and *valence*.

To exemplify our design principles and our ideas of affective gestures, we approached the design of an application for a mobile setting, an affective messaging service. An important part of telephone communication is its usage to maintain intimate and close relationships between people [Castelfranchi]. In mobile phones this is done both through phone conversations but also through text messaging (e g SMS[1] and MMS[2]) [e g Grinter and Eldridge]. In the messaging interaction, the affective bandwidth is narrow, and most of the richness of the emotional content is lost. This also has a negative impact on the communicative bandwidth.

The designed affective message application makes use of a combination of gestures and a pulse sensor as affective input, and uses emotional expressions in graphics (color, shape, animation) as output. An important goal is to mirror form and content of the gesture input in the emotional expressions added to the message.

Below we first describe our design principles in more detail, before we turn to the specific problem of designing the affective gestures. We describe our affective interaction model, which we name *the affective gestural plane model*. The mobile service for

[1] SMS: Short Message Service, used to send text messages between mobile phones.

[2] MMS: Multi-Media Messaging Service, used to send multi-media between mobile phones.

affective messaging, which we describe last, exemplifies how our design framework and the affective gestural plane model might be applied.

2 Designing for affect

While early theories on emotions regarded emotions as discrete states [Ortony et al., Roseman et al.], later work has seen emotions more as processes and appraisal functions that regulate behavior [Paiva], not on or off singular states. As discussed by Castelfranchi, [Castelfranchi], emotions are subjectively experienced states, and we all react differently depending on our background, our previous experience, our mental and physical state and other individual factors. Depending on the social setting we may also express our emotions differently. Expressing happiness during a football game will be quite different from expressing happiness at a business meeting. Thus, recognizing emotional states from biosignals or other physical or external signals is an extremely difficult task – especially in a mobile scenario with its ever-changing psychical and social contexts.

Therefore, emotions as part of human communication is better seen as a human, rich, enigmatic, complex, and ill-defined *experience*. This experience does not solely sit in the brain as part of a rational, cognitive reasoning process. Instead, body and mind are intimately connected [Davies, Dourish, Ekman, Laban and Lawrence, Picard], and *emotions* cannot be seen solely as a mental state but also a physical, bodily, state [Ekman, Picard]. Emotions can be generated through someone's imagination without physical interaction, but they can also be generated from body movements [Ekman]. Try moving as if you are extremely happy and you will probably also experience a warm feeling that slowly grows inside you. It is quite hard to feel sad while jumping up and down and smiling.

In order to design for subjective affective experiences with a user-centered perspective that addresses both body and mind, we extracted four, interrelated design principles that we adjusted to the particular motives and needs of our design situation.

2.1 Embodiment

Dourish [Dourish], defines embodiment as "the creation, manipulation, and sharing of meaning through engaged interaction with artifacts". By artifacts he does not only mean physical objects, but also social practice. Rather than embedding fixed notions of meaning within technologies, embodied interaction is based on the understanding that users create and communicate meaning through their interaction with the system and with each other through the system. The concept of embodiment allows Dourish to combine two trends from the human-computer interaction area; *tangible interaction* where interaction is distributed over the abstract digital world and objects in the physical world [Ishii and Ullmer], and *social computing* where social practice and the construction of meaning through social interaction is core in design [e g Bannon].

Designing for embodied affective interaction thus entails both looking for the physical artifact embodiment of abstract emotion concepts, as well as allowing for social practice and interpretation of meaning of the emotional expressions. The physical embodiment concurs nicely with the strong connection between body and emotion, as discussed above.

2.2 Natural but designed expressions

To get users physically involved, one approach is to build the interaction upon our previous physical and cognitive experiences of emotional processes. This approach can be applied to the design of the whole interaction, including both input and as well as output channels and the connection of the two in the application.

Human-computer interaction and human-computer-human interaction are not and should perhaps not be the same as human-human interaction. An application is a *designed artifact* and can therefore not build solely upon (whatever is meant by) "natural" emotional expressions. On the other hand, using mainly designed expressions bearing no relation whatsoever to the emotional experiences people have physically and cognitively in their everyday lives, would make it hard for the user to recognize and get affected by the expressions. Therefore we argue that emotional expressions should be aiming to be *natural but designed expressions*.

The specific focus of this paper is how to design for affective gestures. When studying the research done on gestures in computer interaction in general there are two main strands that exemplify the conflict: *designed gestures* [e g Long et al., Nishino et al.] and *natural gestures* [Cassell, Hummels and Stappers]. Designed gestures can, for example, be resembled to sign language. The gestures make up a language and depending upon the complexity of the language, it may take quite some effort to learn. Natural gestures, on the other hand, aim to be easier to learn as they build upon how people tend to express themselves in various situations. Body language, posture and more conscious gestures, however, vary between individuals, cultures and situation. Thus, designers of gesture interaction often aim for designed gestures based on natural behavior, looking for the underlying dimensions giving rise to the specific movements.

2.3 Affective loop

The aim of the affective loop idea, is to couple the affective channels of users closely to those of interactive applications, so that the user's emotions are influenced by those emotions expressed by or through the application, and vice versa. Through designing for physical expressions of the end-user (e g body posture, gestures, tangible input through toys, speech) that makes sense with regards to the design of the overall interaction or narrative or the system they interact with, we try to make users involved both physically and cognitively. By having users express their emotions in interacting with the system, they can be engaged in an affective loop, where their emotions are either affected or increased in intensity, either by the modality by which the emotions are submitted or as a response to output.

An example of a system that inspired and explored the affective loop idea is SenToy [Paiva et al.]. SenToy is a doll, which is used as an input device to a game. The end user interacts by acting out various emotions through movements with the doll. For example, to express anger, the user needs to shake the doll back and forth. The idea was that these body movements, together with the resulting activities appearing in the game progression would also influence users emotionally, both their body and mind.

The other part of the affective loop, the emotional output, concerns how the system in turn expresses its response to the user input. Some modalities, such as color and shape [Itten], movement, and music stand a better chance to address our physical experience. For example, according to Ryberg [Ryberg] humans have the same first instinctive reaction to colors. In movies music is used to put us in different emotional states [Bordwell and Thompson]. Bresin and colleagues [Bresin and Friberg] have produced a system, which given a piece of music can replay it to express different emotions.

2.4 Ambiguity

Most designers would probably see ambiguity as a dilemma for design. Gaver, however, looks upon it as "a resource for design that can be used to encourage close personal engagement" [Gaver et al.]. He argues that in an ambiguous situation people are forced to get involved and decide upon their own interpretation of what is happening. As affective interaction oftentimes is an invented, on-going process inside ourselves or between partners and close friends, taking on different shades and expressions in each relationship we have with others, ambiguity of the designed expressions will allow for interpretation that is personal to our needs. For example, if a system was to have buttons where each was labeled with a concrete emotion, users might feel extremely limited since they would not be able to convey the subtleties of their emotional communication to others.

Ambiguity may also follow from the ideas of embodiment, that sees meaning as arising from social practice and use of systems – not from what designers intended originally. An open-ended ambiguous design might allow for interpretation and for taking expressions into use based on individual and collective interpretations – both by sender and receiver of affective messages. Ambiguity in a system will perhaps also create a certain amount of mystery that will keep users interested. However, there needs to be a balance, since too much ambiguity might make it hard to understand the interaction and might make users frustrated [Höök et al.].

3 A model of affective gestures

While any service that attempts to instantiate the design ideas outlined above should be concerned with the whole interaction and not only one part of it, this paper will be focused mainly on the affective input side. As discussed above, we wanted to involve users physically with the application and our idea from the SenToy-work was that natural but designed gestures for affective expressions could be an interesting design alternative.

In order to find affective gestures that can express emotion, we turned to the work by Laban and his colleagues [Davies]. Laban was a famous dance choreographer, movement analyzer and inventor of a language for describing the *shape* and *effort*[3] of different movements. His work will not lend itself to turning emotional expressions into a table with one-to-one mappings of movements to emotions – but his theories of movement can be used to understand the underlying dimensions of affective body behaviors.

To map emotional body behavior to Laban's dimensions of movements, we invited Erik Mattsson[4], an actor, who works with counseling and education in human communication. We asked the actor to express nine different emotional processes in body language, while we videotaped him. In a questionnaire distributed to 80 SMS-users in Sweden we found the emotions they mostly wanted to communicate in mobile messages: *excitement, anger, surprise-afraid, sulkiness, surprise-interested, pride, satisfaction, sadness* and *being in love*.

Before we turn to the analysis of the movements, we need to introduce Laban's formalism for describing movements and theories about shape and effort, at least at a shallow level, in order to understand the analysis of the actor's expressions.

3.1 Shape and Effort according to Laban

Shape describes the changing forms that the body makes in space, while *effort* involve the "dynamic" qualities of the movement and the inner attitude towards use of energy [Zhao].

Motion factor	Dimensions	Examples
Space attention to the surroundings	**Indirect (flexible)**: spiraling, deviating, flexible, wandering, multiple focus	Waving away bugs, surveying a crowd of people, scanning a room for misplaced keys
	Direct: straight, undeviating, channeled, single focus	Threading a needle, pointing to a particular spot, describing the exact outline of an object
Weight attitude to the movement impact	**Light**: buoyant, weightless, easily overcoming gravity, marked by decreasing pressure	Dabbing paint on a canvas, pulling out a splinter, describing the movement of a feather
	Strong: powerful, forceful, vigorous, having an impact, increasing pressure into the movement	Punching, pushing a heavy object, wringing a towel, expressing a firmly held opinion
Time lack or sense of urgency	**Sustained**: leisurely, lingering, indulging in time	Stretching to yawn, striking a pet
	Sudden (quick): hurried, urgent, quick, fleeting	Swatting a fly, lunging to catch a ball, grabbing a child form the path of danger, making a snap move
Flow amount of control and bodily tension	**Free (fluent)**: uncontrolled, abandoned, unable to stop in the course of the movement	Waving wildly, shaking off water, flinging a rock into a pond
	Bound: controlled, restrained, rigid	Moving in slow motion, tai chi, fighting back tears, carrying a cup of hot tea

Table 1: The dimensions of effort according to Laban as described by Zhao [Zhao].

Shape can be described in terms of movement in three different planes: the *table plane* (horizontal), the *door plane* (vertical) and the *wheel plane*, which describes sagittal movements. Horizontal moments can be somewhere in-between spreading and enclosing, vertical movements are presented on a scale from rising to descending, and sagittal movements go between advancing and retiring (Figure 1).

Effort comprises four motions factors: space, weight, time and flow. Each motion factor is a continuum between two extremes (Table 1).

[3] Laban's theory oftentimes referred to as LMA (Laban's Movement Analysis) is composed of five major components: body, space, effort, shape and relationship. The focus in our analysis is on effort and shape as these best describe the emotion expression contained in gestures.

[4] http://www.ordrum.com/erik.html

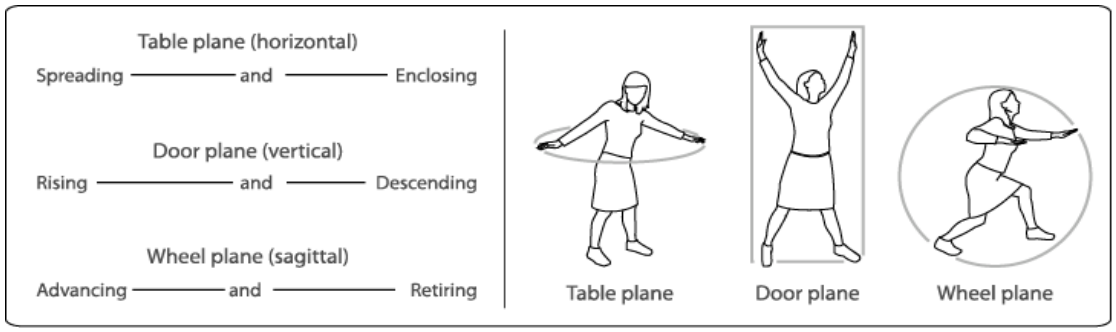


Figure 1: The three different planes of shape, adapted from Davies [Davies].

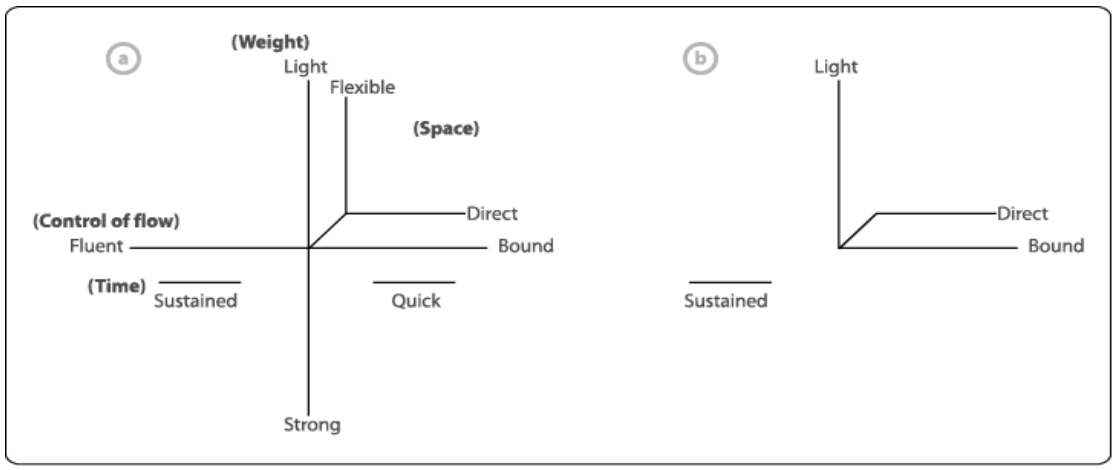


Figure 2: (a) Laban's effort graph, (b) an example effort graph of inserting a light bulb. [Laban and Lawrence]

In figure 2a we depict the graphs Laban uses to express effort. As an example, figure 2b presents an effort graph of the movement of inserting a light bulb where the movement is direct in space, light in weight, sustained in time and bound in control.

3.2 Analysis of emotional expressions in body movements

All of the emotions that the actor was asked to perform may of course give rise to a whole range of different body movements depending on the setting, the background and previous experience of the person, personality, culture and various other factors. This act is only one way that these emotions can be expressed.
Even though, the actor was asked to perform nine distinct emotions, his act was more like a process working on the concept of each given emotion, going from starting the expression to "feeling" it more and more, expressing it stronger, and then varying it using various alternative interpretations of when this emotion would arise. In figure 3, an example of the actor's expression of each emotion is depicted. The analysis, however, was performed on the whole sequence of expressions for each given emotion. Two independent persons (two of the authors) did the same analysis of the videotape, after which notes were compared and discussed.

3.2.1 Shape and effort

Using Laban's theories of shape the actor's interpretation can be described as follow:

- Excitement – extremely spreading, rising and advancing movements.
- Anger – somewhat spreading, rising and advancing movements.
- Surprise-afraid – enclosing, somewhat descending and retiring movements.
- Sulkiness – enclosing, somewhat rising and retiring movements.
- Surprise-interested – somewhat spreading, neutral in the vertical plane and advancing movements.
- Pride – somewhat spreading, rising and somewhat advancing movements.
- Satisfaction – neutral in all planes of movements.
- Sadness – enclosing, descending and retiring movements.
- Being in love – somewhat spreading, somewhat rising and somewhat advancing movements.

Figure 4 presents the corresponding effort graphs using Laban's notation.
From looking at our analysis of emotional body language the nine emotions, presented in figure 4, can be divided into three groups with different effort levels, starting with the one with highest effort:

1) Excitement, anger, surprised-afraid
2) Sulkiness, surprised-interested, pride, satisfaction
3) Sadness, being in love

This far we had worked with two variables, shape and effort, but the different emotions are still clustered, for example excitement and anger have nearly the same shape descriptions and exactly the same effort graphs (Figure 4). Therefore, we looked for a third variable, which we found in Russell's "circumplex model of affect" [Russell].

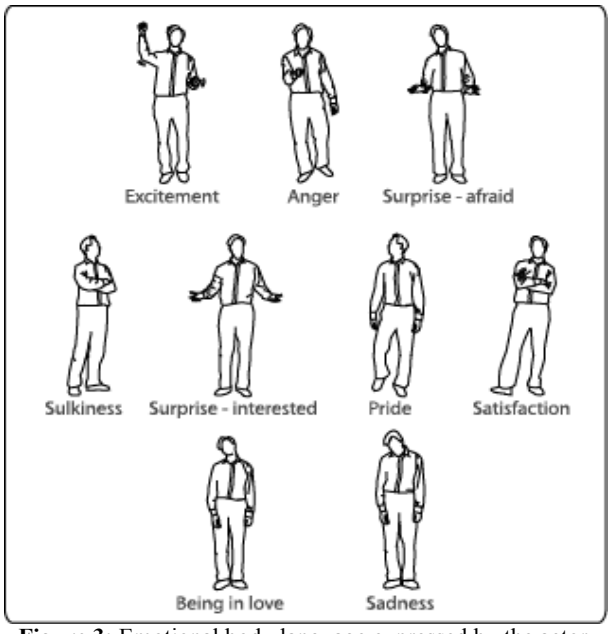


Figure 3: Emotional body language expressed by the actor.

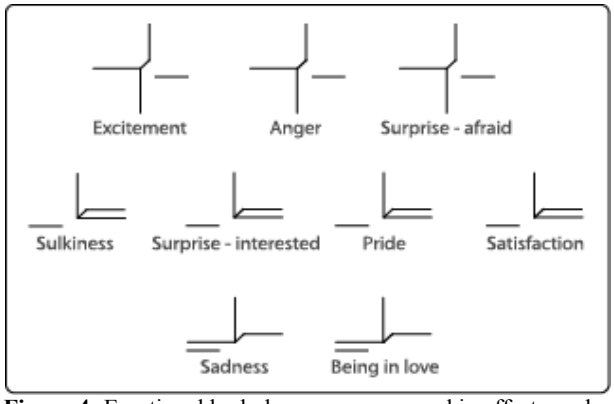


Figure 4: Emotional body language expressed in effort graphs.

3.2.2 Valence

In the "circumplex model of affect" psychologist Russell looks at emotions in terms of *pleasure* and *displeasure* (here named *valence*) and *arousal*. Since a high degree of effort brings a high degree of arousal and vice versa Russell's analysis of emotions concurs nicely with Laban's theories of movements. Thus, *valence* is our third variable. In a series of studies Russell established that people have the same mental map of how emotions are distributed in a system of coordinates where the y-axis is the degree of arousal and the x-axis is the valence (Figure 5). The subjects, for example, placed angry and delighted on the same arousal level but with different valence.

3.3 Designing emotional expressions with a basis in shape, effort and valence

To conclude the above analysis it is necessary to set up a combination of shape, effort and valence to create an affective interaction were it is possible to express all kinds of emotional states without resorting to a one-to-one mapping. It is not necessary, however, to incorporate all dimensions: shape, effort and valence, into a new modality. It can likewise be a combination of the modality and emotional expressions in the interface. We will show an example of the latter in the next section.

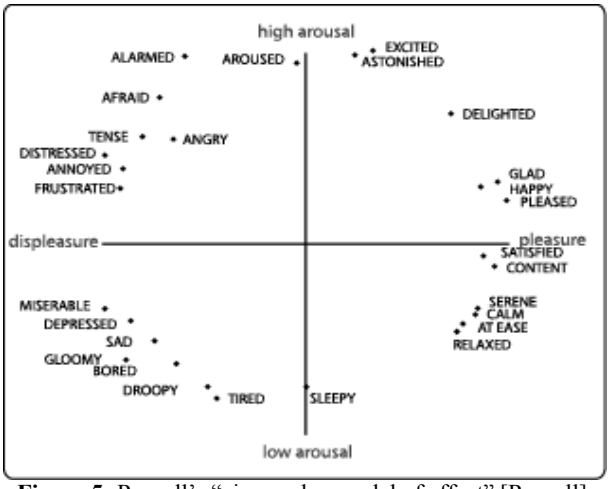


Figure 5: Russell's "circumplex model of affect" [Russell].

4 A mobile service for affective messaging

The goal of the affective message service is to provide users with a means to enhance their messages with emotional expressions. With today's technology, such as MMS, users can add photos, colors, sound or animations to messages, but it is quite time-consuming, difficult to create on the fly and to get the right expression of such a messages. Instead, our idea is to build an interactive service on top of the MMS-technology that expands on the expressive power while still allowing for ambiguity and open interpretation of the affective content.

In the questionnaire (mentioned briefly above) the answers indicated that most users feel limited or alien to expressions such as smilies as a means to express emotions in text messages. Not only is the emotional content restricted but also the emotional interaction with the other party. In a phone conversation, the voice itself can be a bearer of emotional content that complements what is being said. Thus, both parties in the conversation receive too little emotional feedback and are provided with too little emotional expressive power when composing or receiving text-messages. The users in our questionnaire expressed a need for a richer medium.

Below follows a description of the mobile service and thereafter we will explain how shape, effort, valence and the four design principles are incorporated.

Our design example is an emotional text messaging service built on top of a SonyEricsson P800 mobile terminal, where the user can write a text message and then adjust it to fit the emotional expression they want to achieve. The adjustments are mainly done through affective gestures, but with a little mystery added through obscuring the input through mixing it with measurements of the users' pulse. The affective gestures performed with the stylus used with the P800 terminal, together with the pulse will render an animated background with an emotional expression to the user's text message. Figure 6 shows a usage scenario.

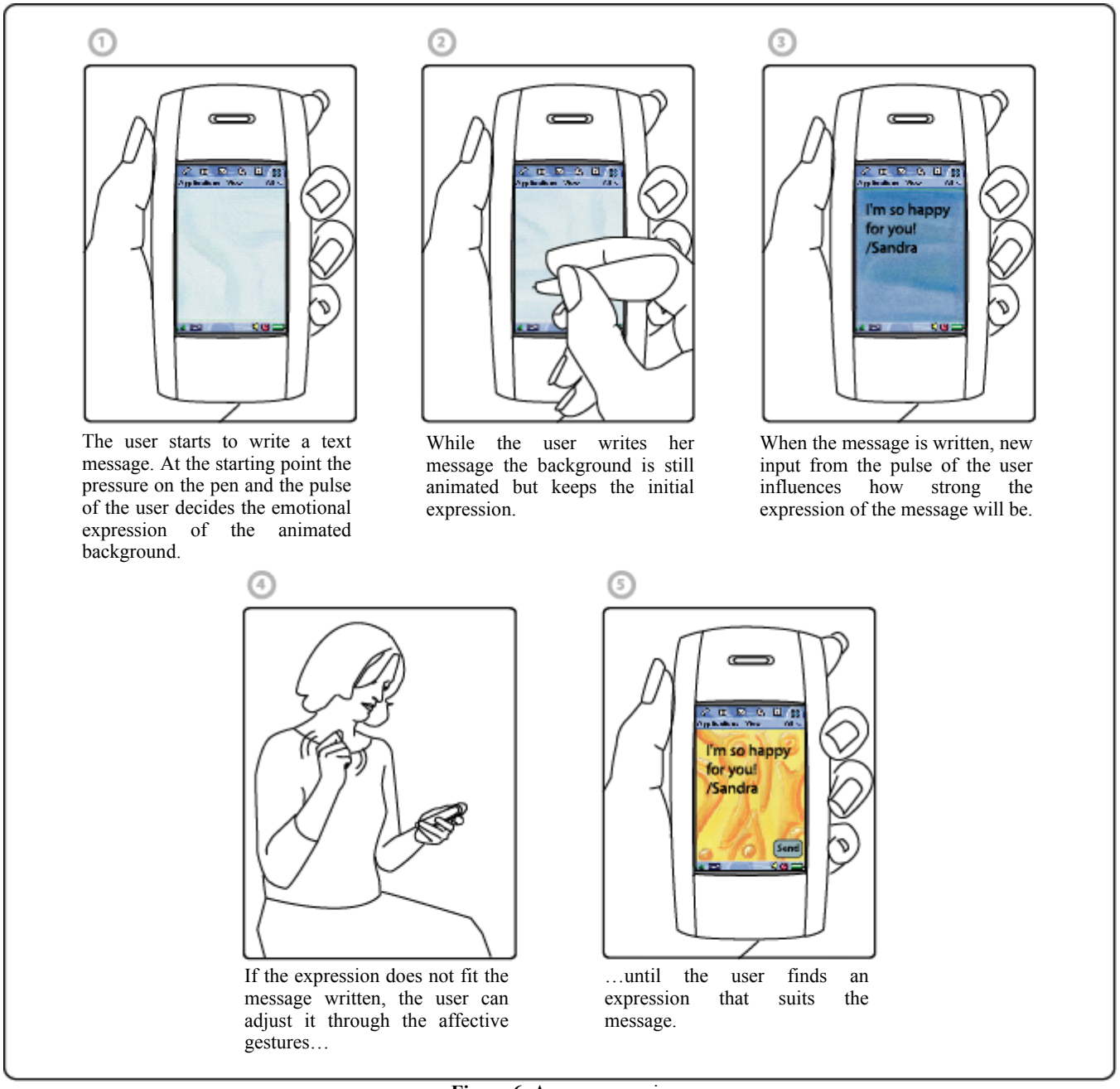


Figure 6. A usage scenario

4.1 Shape, effort and valence in the interaction

We use Russell's "circumplex model of affect" (Figure 5), as the basis for the interaction. The user will be moving around in the circular space of emotions expressing effort and valence of their emotional state through combinations of two basic movements that when combined can render an infinite amount of gestures. We call these combinations of the two movements the *circumplex affective gestures* (Figure 7):

- Moving along the valence scale towards displeasure is done through increasing the pressure on the stylus, decreasing the pressure on the pen results in higher pleasure on the valence scale.
- Shaking and making faster movements, with the hand holding the pen, requires more effort and therefore result in higher arousal, while more swinging, not so direct movements result in lower arousal.

The circumplex affective gestures are inspired by the shape, effort and valence analysis. Emotions with negative valence are associated with strain and tension, while positive emotions often involve less pressure and strain. Emotions with high effort are stronger in weight, more flexible in space and quicker in time, while emotions with less effort are less controlled, lighter and smaller in space. While the user is performing the circumplex affective gestures, the system is responding through showing the emotional expressions in color, shape and animations as indicated in figure 8. The emotional expression works like an animation in

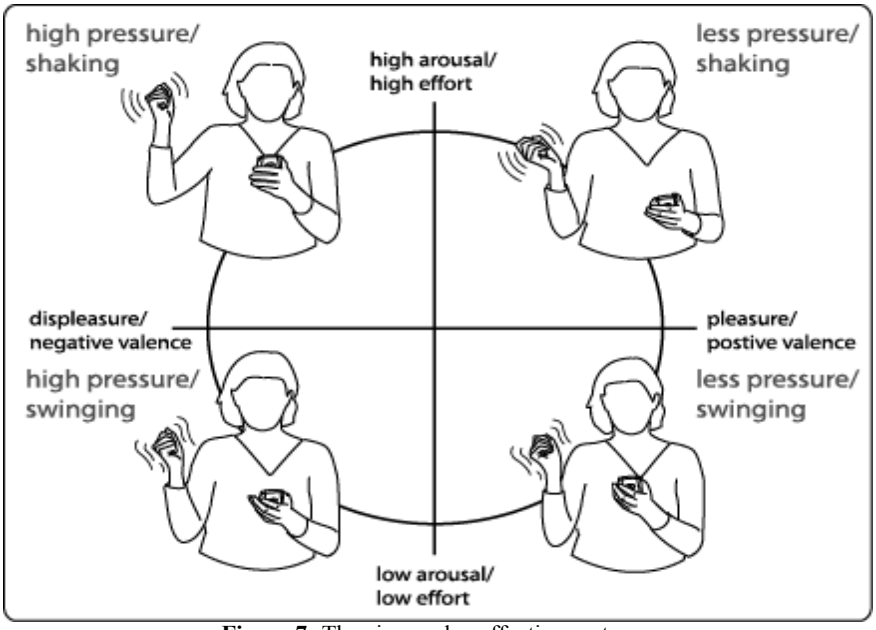


Figure 7: The circumplex affective gestures.

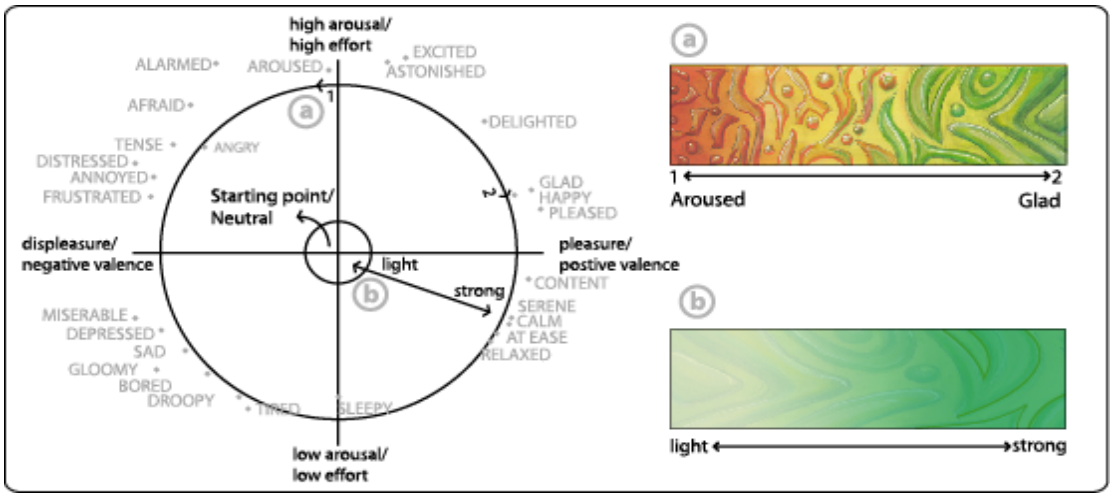


Figure 8: The affective gestural plane, a. showing how the output is expressed when interacting and b. showing how the pulse decides the width of this plane, presented in light to strong colors.

the background of the message, giving the writer immediate feedback on the appearance of the message (Figure 8a). The user activates this input by holding in a button on the pen. Once the user finds the expression she wants, the button on the pen is released and the expression is thereby chosen.

The animations allows the different emotions to float into each other similar to how Russell argues that emotions blend into one another and do not have any defined borders. Still, the characteristics of each emotion found in the analysis of body movements are clearly represented through the choice of colors, shapes and movements. Most of the emotions, or their position in Russell's circular model, can be expressed through colors. Red represents, according to Ryberg, the most powerful and strong emotions. Moving along a color scale ending with blue would be moving towards calm and peaceful emotions. The strength of an emotional state could then be expressed in terms of deepening the color. In this example we are not working with the actual text in the message, neither with sound, but it is something that can be added in future work. Much can be done with different typefaces, sizes and animations of text, [Forlizzi et al.], sound and music can also convey emotional content [Bresin och Friberg].

As an example, the characteristics of the emotion *excited* entail much energy, it is high in effort, and the movements are extremely spreading, rising and advancing. This can be used to create an animation and coloring as in figure 9 (where the animation cannot be shown in this paper).

The circumplex affective gestures would probably render a predictable and thereby less interesting interaction. We therefore decided to add the pulse sensor, which is integrated in the pen, measuring the user's pulse while writing.

The model combining pulse with the pressure on the pen, as shown in the usage scenario, decides where in the circular space of emotions the user initially starts:

- If your pulse is high and you are holding the pen firmly, you will start where there is high effort and negative valence
- With high pulse and a lighter grip around the pen you will end up where there is high effort and positive valence

- Low pulse and a firm grip will put you where there is low effort and negative valence
- Low pulse and a lighter grip will put you where there is low effort and positive valence

Figure 9: How "excited" is expressed in the message

The user always starts the message with a light emotional expression. When the user has finished writing her text, the pulse decides the width of the circular space of emotions, which is presented as the strength of the emotional expression – varying from light to strong (Figure 8b). This combination of circumplex affective gestures and the pulse sensor we named the *affective gestural plane model*. The intention is to achieve a kaleidoscopic effect, so that e g "sad" always has the characteristics of sadness but never takes on exactly the same expression. This will hopefully maintain the user's interest.
If the pulse signal were the only way for the user to provide input to the system, the user would not be in control of the interaction at all, which in turn would be both frustrating and probably render erratic interpretation of users' affective states most of the time. But since the circumplex affective gestures allows the user to move around the circle of the affective gestural plane, the user is still allowed most of the control.

4.1.1 The interaction device

Designing for emotional input requires a coherency between the actual product's physical design and the task performed. In this case, the stylus has to be designed in such a way that it appeals to our emotional sensing. You are probably more likely to hug and pat, for example, a teddy bear than a laptop.

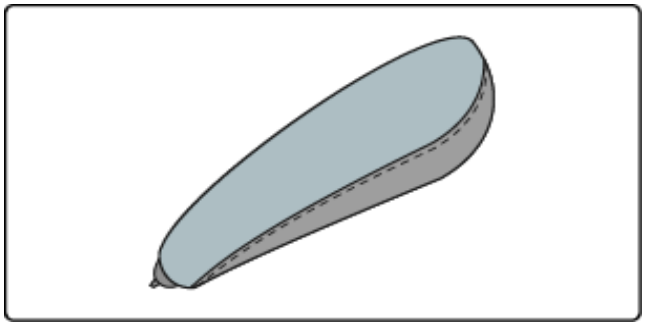


Figure 10: A design example of the interaction device.

On the other hand, it is also important that the interaction device does not take on any personality or emotional state in itself. It must not look like some character or carry a specific expression [Andersson et al.], but instead be bland enough to carry users' intentions. Making a pen that is quite characterless, but still emotionally appealing will provide a suitable artifact for affective computing but still keep the user focused on the interaction. Figure 10 shows a design example.

4.2 Incorporation of design principles

The design principles introduced above, all played an important role in the design of the affective message service. *Embodiment* is realized both in terms of the actual physical interaction with the extended stylus, as well as through how the user will experience the circumplex affective gestures as such. The two taken together, embody and aid users to externalize the internal emotional states they want to convey.
The principle of *Natural but designed expressions* is incorporated through the circumplex affective gestures and the interactive feedback that are designed to resemble the shape, effort and valence of natural emotional movements.
Since the design is trying to address both body and mind the emotional state of the user is reinforced not only through the gestures, but also through the response that the system generates, and therefore the interaction will involve the user in an *affective loop*. While not discussed in this paper, the interaction with the receiver of the affective message will also constitute another affective loop interaction.
Ambiguity is achieved in the affective gestural plane model as well as in the interactive feedback. The pulse sensor creates a small proportion of mystery in the interaction, thus keeps the user interested in exploring their emotional expressions further. By using circumplex affective gestures to navigate the affective gestural plane, we avoided that one gesture corresponds to one emotion, and instead created an interaction where users can create their own language and make their own interpretations of the interactive feedback.

5 Summary

We have shown how to go from a user-centered perspective, involving both body and mind, via theory of movements and emotional expressions, a study of an actor and his emotional expressions, to a specific design of a set of *circumplex affective gestures* for expressing emotion to a mobile messaging service.
We are aware of that this work is somewhat cultural dependent, however, we find this piece of work valid and interesting as input even if not entirely possible to generalize irrespective of culture and personality.
In particular, we have identified three underlying dimensions of bodily emotional expressions: *shape*, *effort* and *valence* that we have incorporated in the design of our mobile service both in the *affective gestural plane model* as well as in the interactive feedback. This frees us from design solutions that assume that users will be in discrete, well-defined emotional states, where one gesture (or input signal) corresponds to one emotion. Instead our specific design approach allows for an interpretative, interactive cycle with the emotional output that will place users, and their interpretation of emotional expressions and needs for how to express themselves, at core. This diverts from the existing trends in affective computing, where the focus is not on the emotional *experience* as such but on recognizing and adjusting to what the system believes that the user is feeling.

Acknowledgements

Foremost we like to thank Erik Mattsson for his cooperation in expressing the emotions. We would also like to thank Lars Wallin, Jakob Tholander, Mattias Esbjörnsson, Peter Lönnqvist, Jussi Karlgren, Annika Waern, Phillip Jeffrey and Martin Jonsson for discussions and comments on earlier drafts of this paper.
This work is partly funded by SSF through the "Mobile Services"-project.

References

Andersson, G., Höök, K., Mourao, D., Paiva, A. and Costa, M. (2002) Using a Wizard of Oz study to inform the design of SenToy, *Proceedings of the conference on Designing interactive systems*, ACM Press.

Ark, W., Dryer, D. and Lu, D. (1999) The emotion mouse, *Proceedings of HCI International 1999*. Munich, Germany.

Bannon, L. (1991) From humans factors to humans actors: The role of psychology and human-computer interaction studies in system design, *Design at Work: Cooperative Design of Computer Systems*, J. Greenbaum and M. Kyngs, eds, Hillsdale, N.J.:Erlbaum.

Bordwell, D. and Thompson, K. (2001) *Film Art, An Introduction*, Sixth Edition, McGraw-Hill, New York, USA.

Bresin, R. and Friberg, A. (2000), Emotional Colouring of Computer-controlled Music Performances, *Computer Music Journal*, (24)4, pages 44-63.

Cassell, J. (1998) A Framework for Gesture Generation and Interpretation, *Computer Vision in Human Machine Interaction*, R. Cipolla and A. Pentlan, eds. Cambridge University Press, New York, USA.

Castelfranchi, C. (2000) Affective Appraisal versus Cognitive Evaluation in Social Emotions and Interactions, *Affective Interactions, Towards a New Generation of Computer Interfaces,* Ana Paiva, ed, Springer-Verlag, Berlin.

Cummings, J. N., Kraut, R. and Kiesler, S. (2001) Do we visit, call, or email?: media matter in close relationships, *Proceedings of the Conference on Human Factors and Computing Systems (CHI '01)*, extended abstracts, Seattle, Washington, USA.

Davies, E. (2001) *Beyond Dance, Laban's Legacy of Movements Analysis*, Brechin Books ltd., London, UK.

Dourish, P. (2001) *Where the action is. The Foundations of embodied Interaction*, MIT Press, Cambridge, MA, USA.

Ekman, P. (1972) *Emotion in the Human Face*, Pergamon Press Inc., New York, USA.

Fernandez, R., Scheirer, J. and Picard, R. (1999) Expression glasses: a wearable device for facial expression recognition. *MIT Media Lab Tech. Rep. 484*.

Forlizzi, J., Lee, J. and Hudson, S. E. (2003) The Kinedit System: Affective Messages Using Dynamic Texts, *Proceedings of the conference on Human Factors in Computing Systems (CHI'03)*, Ft. Lauderdale, Florida, USA.

Gaver, W.W., Beaver J. and Benford, S. (2003) Ambiguity as a Resource for Design, *Proceedings of the conference on Human factors in computing systems (CHI'03)*, Ft. Lauderdale, Florida, USA.

Grinter, R, and Eldridge, M. (2003) Design for the socially mobile: Wan2tlk?: everyday text messaging, *Proceedings of the conference on Human factors in computing systems (CHI'03)*, Ft. Lauderdale, Florida, USA.

Hummels, C. and Stappers, P.J. (1998) Meaningful Gestures for Human Computer Interaction: Beyond Hand Postures, *Proceedings for the third IEEE international conference on Automatic Face & Gesture recognition (FG'98),* Nara, Japan.

Höök, K., Sengers, P. and Andersson, G. (2003) Sense and Sensibility: Evaluation and Interactive Art, *Proceedings of the conference on Human factors in computing system (CHI'03)*, Ft. Lauderdale, Florida, USA.

Höök, K (forthcoming) User-Centred Design and Evaluation of Affective Interfaces, *Evaluating ECAs*, Pelachaud, C. and Ruttkay, Z. Kluwer.

Ishii, H., and Ullmer, B. (1997) Tangible bits: Towards seamless interfaces between people, bits and atoms. *Proceedings of Conference on Human Factors in Computing Systems (CHI'97)*. ACM Press.

Itten, J (1971) *Kunst der Farbe*, Otto Maier Verlag, Ravensburg.

Laban, R. and Lawrence F.C. (1974) *Effort, Economy of Human Effort*, Second Edition, Macdonald & Evans Ltd., London, UK.

Long, A. C. Jr, Landay, J.A., Rowe L. R. and Michiels, J. (2000) Visual Similarity of Pen Gesture, *Proceedings of the conference on Human factors in computing system (CHI'2000)*, The Hague, Amsterdam.

Nishino', H., Utsumiya', K., Kuraoka, D. and Yoshioka, K. (1997) Interactive two-handed gesture interface in 3D virtual environments, *Proceedings of the ACM symposium on Virtual reality software and technology*, ACM Press, Lausanne, Switzerland.

Ortony A., Clore G.L. and Collins A. (1988) *The Cognitive Structure of Emotions*, Cambridge University Press, Cambridge, UK.

Paiva, A. (Ed.) (2000) Affective Interactions, *Towards a New Generation of Computer Interfaces*, Springer-Verlag, Berlin.

Paiva, A., Costa, M., Chaves, R., Piedade, M., Mourão, D., Sobral, D., Höök, K., Andersson, G., and Bullock, A. (2003) SenToy: an Affective Sympathetic Interface, *International Journal of Human Computer Studies*, Volume 59, Issues 1-2, July 2003, Pages 227-235, Elsevier.

Picard, R. (1997) *Affective Computing*, MIT Press, Cambridge, MA, USA.

Roseman I.J., Antoniou A.A. and Jose P.E. (1996) Appraisal Determinants of Emotions: Constructing a More Accurate and Comprehensive Theory, *Cognition & Emotion*, 10(3), Pages 241-277.

Russell, J.A. (1980) A Circumplex Model of Affect, *Journal of Personality and Social Psychology*, Vol. 39, No. 6, pages 1161-1178, American Psychological Association.

Ryberg, K. (1991) *Levande färger*, ICA Bokförlag, Västerås, Sweden.

Zhao, L. (2001) *Synthesis and Acquisition of Laban Movement Analysis Qualitative Parameters for Communicative Gestures*, PhD thesis, CIS, University of Pennsylvania.

Recognition Error Handling in Spoken Dialogue Systems

Genevieve Gorrell

Linköping University*

Abstract

Increasing use of mobile devices for information access brings with it a demand for robust speech recognition. An important way in which speech recognition system performance can be improved is through use of dialogue strategies to handle the situation in which the system fails to recognise the user's utterance. In the work described here, the case is made for combining multiple speech recognisers and appropriate dialogue strategies to handle poor recognition results. Two implemented systems are discussed as examples.

1 Introduction

Use of mobile and ubiquitous multimedia in the form of small informational devices, along with an increased interest in informational access during parallel activities brings speech recognition very much into focus. Firstly, mobile devices present challenges for data input, and speech is a very natural option for human beings. Secondly, combination of information access with other activities often requires the freeing of the hands, precluding many data entry options. At the same time, environments in which mobile devices might be used are challenging for speech recognition; noisy environments and distracted users both place demands on performance.

Consider an interactive informational dialogue system, such as a telephone system making train times or public information available, or selling cinema or airplane tickets. Speech recognition presents challenges for dialogue in that the possibility must always be accounted for that the system misrecognised what the user said. A spoken dialogue system is in this way quite different to a written one. Speech enabling an existing text-based system therefore involves more than just adding a speech-recogniser onto the front; a successful integration will involve some changes to the dialogue structure.

To introduce spoken dialogue into the mobile environment, a lot of work needs to be done on making such systems robust to error. Much work has been and is being done on error handling in spoken dialogue, both in the research sector and commercially, see for example [Err 2003]. Errors can be recovered from after the fact; a recognition result already accepted by the system can be corrected by the user after some implicit confirmation/feedback to the user about what the system believed was said. For example;

System: Destination?
User: Linköping
System: And how would you like to travel to Nyköping?
User: No, I said Linköping

More explicit confirmation can take the form of an entire dialogue turn devoted to telling the user what was understood and asking them if the understanding was correct; "Nyköping, is that correct?" The most appropriate strategy depends on the context - how critical is it that the system understood correctly - and also, how much doubt is there about the interpretation.

Examples of situations where a system mistake can be high-cost include home control systems, where we wouldn't want for example the freezer to be turned off by mistake, or ticket booking, where again we wouldn't want a bank account to be debited incorrectly. Low-cost situations might include question-answering systems, where it is as quick to answer the wrong question as it is to confirm the understanding of the question.

There are various ways in which the system can glean information about the reliability of a recognition result. For example, a low confidence score from the speech recogniser would be an obvious indicator that some confirmation or disambiguation would be advisable. Another source of information is the likeliness of that utterance given that context. An utterance might be irrelevant to the domain, semantically meaningless, or just simply beyond the scope of the system.

This work discusses the situation where either the user says something beyond the scope of the system, or the utterance is recognised incorrectly as something beyond the scope of the system. Two implemented approaches will be described. The first, Targeted Help [Gorrell et al. 2002] focuses on training the novice user. The second develops on the first, and is equally appropriate to the more experienced user, who perhaps is having difficulty getting recognised by the system due to the acoustic environment.

The next two sections will describe each approach in detail, before the two approaches are discussed in comparison to each other in the following section. The conclusion will focus on the context of the work and future directions.

2 Description of the First Approach

User: Hello?
System: Hello. How can I help?
User: Um, can I have some lights on?
System: Sorry, I didn't quite catch that. To turn a device on or off, you could try something like "turn on the kitchen light".
User: Turn on the kitchen light.
System: The kitchen light is now on.
User: What other sorts of things are there?

*email:gengo@ida.liu.se

System: I didn't quite catch that. To find out what devices there are, you could try something like "what devices are there", or "what devices are there in the lounge".
User: What devices are there in the lounge?
System: There are three things in the lounge; the computer, the vcr and the television.
User: Turn on the tv and the vcr and check the computer is off.
System: I didn't quite catch that. Long commands can be difficult to understand. Perhaps try giving separate commands for each device.
User: Okay, check the computer's off ...

When a dialogue system fails to obtain a recognition result with sufficient confidence, the usual way to handle it is to discard the result, perhaps ask the user to repeat themselves or change dialogue strategy. This is despite the fact that a poor recognition result can contain useful information.

It is also usual that some general help is available in how the user should address the system. This help is typically available whenever the user asks for it, for example, by saying "help", and is also perhaps returned automatically to the user after three recognition failures in a row. The help, being general to the system, can be quite unwieldy and tedious to listen to.

The Targeted Help approach features a number of more specific help messages, each helping the user to perform a particular task. When the user fails to get recognised, their utterance is used to choose the help message most likely to assist them in doing what they appear to be trying to do. This message is returned after each recognition; being specific and fairly brief, the main disincentive to giving help immediately on recognition failure is reduced. Subject trials show improved performance with the Targeted Help system compared with a standard strategy.

2.1 Description of the system

The base system is the On/Off House (OOH) system, a smart home system implemented using the Nuance Toolkit platform [Nuance 2002]. It offers English spoken language control, via telephone, of about 20 devices in a simulated home. Device types include both on/off and scalar. The dialogue manager is implemented in Visual C++ using the Nuance DialogueBuilder API. The mode of operation is primarily user-initiative, that is to say, the dialogue is led by the user. The grammar offers coverage of a fairly broad range of language, including commands ("Turn on the heater", "Turn off the light in the bathroom") and several types of questions ("Is the heater switched on?"; "What is there in the kitchen?"; "Where is the washing machine?"; "Could you tell me which lights are on?").

Targeted help has been added to the system such that whenever an utterance is not recognised above a certain confidence threshold using the grammar-based system some further processing is done. First, the utterance is passed to a domain-specific recogniser based on a statistical language model (SLM) for a second recognition. The result of this recognition contains information such as what words the recogniser recognised, what confidence scores it places on those words, what confidence score it places on the entire utterance, etc. This result is used to create a feature set. A decision tree classifier is then used to classify the feature set and return the class. This class maps to an error message, which is played to the user before returning to the main loop of the application. The error message played will typically be of the generic form

> "I didn't quite catch that. To *(carry out some action)*, you could try something like *(example of suitable command)*."

Section 2.3 contains examples of error messages.

The SLM, described in more detail in [Knight et al. 2001], was created using about 4000 transcriptions of utterances collected using the On Off House system, plus a further 200 utterances appropriate to the home control domain collected using only recognition feedback. The performance of this SLM is comparable to that of the grammar-based recogniser over a mixed corpus.

2.2 Classifier

The module responsible for selecting the help message has been implemented as a simple decision tree classifier built using the popular See5 system [RuleQuest 2002]. This classifier was trained on about 1000 utterances, including the 200 less constrained utterances used to create the SLM. The remainder of the classifier training corpus was also taken from the SLM training corpus. The training data was prepared by recognising each utterance in turn using the SLM and then processing the output of the recognition to produce the desired feature set. This means that both at training time and at run-time the classifier is using output of the same SLM.

The classifier feature set used consisted of the following: the individual words; their confidence scores; the utterance confidence score; the number of words in the utterance; the number of occurrences of each of the items "on/off", "the", "and" and "turn/switch" (four features); whether or not the utterance started with each of the items "what is there/what's in", "is there", "where/where's", "turn/switch", "the", "what is on/what is switched on/what's on", "which", "could/can/would", and "are/is" (nine features); whether or not the utterance contained an occurrence of each of the items "are there any/is there any", "what/what's", "is anything", "turn/switch", "please" and "everything/all" (six features); and whether the utterance ends with "are there".

2.3 Classes

The classifications used were hand-selected based on observation of the corpus. Below are the most common of the 12 classes with their associated error messages and percentage of the training corpus covered by each;

REFEXP_COMMAND(35%) - "I didn't quite catch that. To turn a device on or off, you could try something like 'turn on the kitchen light'."

LONG_COMMAND(13%) - "I didn't quite catch that. Long commands can be difficult to understand. Perhaps try giving separate commands for each device."

PRON_COMMAND(11%) - "I didn't quite catch that. To change the status of a device or group of devices you've just referred to, you could try for example 'turn it on' or 'turn them off'."

REFEXP_STATUS_QUERY(9%) - "I didn't quite catch that. To find out the status of a device, you could try something like 'is the light on' or 'is the kitchen light on'."

DEFAULT_ERROR(15%) - "Sorry, try again."

2.4 Decision Tree

The baseline error rate for the classification task is 65%, if the system classifies everything as a REFEXP_COMMAND (the most common class). Classification based only on the first word improves the error rate to 40%. The error rate for the final decision tree, measured using cross-validation on the training data, was 12.2%.

2.5 Evaluation

User trials were performed in order to determine the success of the approach. The targeted help system and a control system were made available over the telephone. Both systems made the user aware of the global "help" option in the introductory prompt. The global help message was the same in both systems and consisted of all the targeted help messages read out in sequence. Upon recognition failure, the "helpful" system issued a targeted help message as described above, whereas the control simply had one behaviour: first, to say "Sorry, try again" and then, on a consecutive failure, to issue a short version of the initial help message. This control strategy was chosen to reflect an approach commonly taken in commercial systems.

31 novice users were given a scenario which involved ringing a voice controlled house and leaving it in a secure state: fire risks were to be minimized, yet the house should appear occupied to deter burglars. After completing the task, users filled in a short questionnaire. Sixteen users performed the task with the targeted help system, and fifteen with the control. The following were measured; task completion, dialogue length, word error rates, in coverage rates, and users' knowledge of the house and the system's capabilities.

Analysis showed that users of the targeted help system had a significantly greater average task completion score, and their word error rates were very significantly lower than the control group's. The number of in-coverage utterances was also significantly higher in the targeted help system. This suggests users more quickly attune themselves given the intelligent help system. The word error rates for our statistical language model show a similar pattern over our corpus, being notably lower for the targeted help users. For a more detailed description of the results, see [Gorrell et al. 2002].

Users of the intelligent help system also used a greater variety of constructions in their interactions. For example, as well as "turn on the X" and "is the X on" they were more likely to try "what is on" or "what devices are there". Also, after hearing a system suggestion, they often used it immediately. Questionnaire results suggested that Targeted Help users were ending the calls with a greater awareness of the state in which they left the house. Examination of the conversations also suggest a more systematic approach to checking the house on the part of the Targeted Help users.

2.6 Discussion of the First Approach

The positive evaluation result suggests that there is scope for improving system performance with regards to recognition failure behaviour in spoken dialogue systems using the techniques outlined here. Based on this, the system described in the next section develops the theme.

3 Description of the Second Approach

User: Where does the swan go to in the winter? (recognised correctly)
System: Do you mean "Where does the swan migrate to"?
User: Yes
System: The swan migrates to ...
User: What about the seagull? (recognised as "water seagull")
System: Do you mean "how about the seagull"?
User: Yes
System: The seagull migrates to ...

In Targeted Help, twelve messages were designed to broadly cover all the things that the user might want to do with the system, and a classifier was used to attempt to deduce, from a statistical recognition result, which they were aiming at. The approach proved robust and was well-received by the test group. However, use of the decision tree classifier limited the number of help messages that could be offered; the more messages, the more mistakes the classifier will make. In fact, the interpretation grammar provides the ultimate definition of the range of utterances that the system can handle, from a natural language understanding perspective, and is much finer-grain than the approximately 12 categories that Targeted Help was limited to.

In this approach, a first step is taken toward using the interpretation grammar to assist the user in getting the desired response from the system. The dialogue is also streamlined, compared with Targeted Help; in the case where the system guesses correctly, the user is not required to repeat the suggestion.

The process, summarised, is as follows. Two grammars are used to recognise the user's utterance; a context-free grammar and a trigram language model. The trigram is used as a back-off, so a result accepted with a confidence greater than 45 on the CFG is not passed on to the trigram. The result is then parsed with a guide grammar. If the utterance does not parse, then a suggestion is formulated using the guide grammar and returned to the user. (If the result does parse, it is accepted by the system.)

3.1 Description of the System

A speech recognition component has been added to the BirdQuest system [**?**]. BirdQuest is a question-answering system that answers questions in Swedish about Nordic birds.

The recogniser makes use of Nuance 8, and consists of a hand-coded CFG grammar, plus a back-off statistical language model. The grammar-based recogniser has been in use since January 2003 and has been through some three development iterations. It achieves a word error rate of 56% on a transcribed corpus of 633 utterances collected using it. This is admittedly rather high but the corpus is small and contains a disproportionate amount of non-native Swedish speech. Also the domain is such that questions of very varied form appear. It includes many novice users and much of the data was collected in noisy conditions. The SLM was created from this corpus. The small size of the SLM training corpus leaves the reliability of the SLM open to question, and also precludes the separation of a test corpus for demonstration of its performance. However, anecdotally, performance is sufficient to demonstrate the technique that is the main point of this work. Combined, the recognisers perform surprisingly well, and it is hoped that both recognisers can be

improved upon in the future.

Speech recognition is implemented in such a way that a result from the grammar-based recogniser with a confidence score over 45 is accepted without question. An utterance rejected by the grammar-based recogniser is passed on to the SLM, and recognised with a confidence threshold of 30. If the utterance is not accepted by either recogniser, then the system simply rejects the utterance; "I'm sorry, try again". If however the utterance is accepted, then the following further processing is done.

A guide grammar has been created, closely based on the CFG used in the primary recogniser. Every result is first parsed using this grammar. If the utterance parses then the system accepts it. If, however, the result does not parse using the guide grammar, then the closest string in the grammar to the utterance is returned to the user as a suggestion; "Did you mean X?" The user can reply yes, in which case the suggestion is fed into the system. Or they can reply no, in which case the system apologises. Or they can ignore the suggestion entirely and simply say a new question (or repeat the old one).

3.2 The Grammar

The grammar is a hand-coded CFG in Nuance GSL format. It is ad hoc, not particularly linguistically-motivated, though an attempt has been made to enforce agreement constraints, and includes some hand-coded probabilities, which have been demonstrated to make a performance improvement of a couple of percent. It contains some 55 rules in total.

3.3 The Corpus

The corpus comprises three collections of utterances in which external users were allowed access to the system, plus a collection of material acquired during development and internal trials. It consists of a transcribed set of 633 wave files. The first of the collections performed with external users was obtained during a conference demonstration. At this stage, recognition was done using a smaller "first cut" grammar. There is noticeable background noise in this section of the corpus. Users were given minimal guidance. The second external data collection was performed during a university open day, and is similar in nature to the first. The third collection was acquired during a demonstration to a small group. At this stage, the grammar had been improved and the statistical recogniser had also been added to the system. The background noise is significantly reduced in this data collection. Additionally, the users were slightly familiar with the system, having had the opportunity to try it during the first external data collection. The majority of the utterances collected during development were spoken by the author, and are somewhat limited in coverage. Other speakers do also feature however.

Additionally, a quantity of written domain-appropriate material was available from a corpus collection performed earlier to inform the development of the original text-based system. This corpus has also been utilised in creating the recognition language models.

3.4 The SLM

Transcriptions of the entire spoken corpus, plus the written corpus, were compiled, using Nuance 8 tools, into a 3-gram statistical language model. Performance figures are not available at this stage because the corpus is too small to allow for a test section to be separated out. Anecdotally, however, the performance is comparable to or perhaps marginally better than that of the grammar, though the nature of the utterances on which good results are obtained differs.

3.5 Forming Suggestions - The Parser

The recognition result is processed in the following way. First an attempt is made to parse the result with the "guide grammar", a grammar similar to the recognition grammar but designed for this purpose. If this parse succeeds then the result is taken to be a good one. If the parse does not succeed then every path through the guide grammar is assessed for its closeness to the result. Closeness is gauged in terms of number of shared words. The best path, ie. the sentence allowed by the grammar that has most words in common with the recognition result, is returned to the user as a suggestion.

3.6 Discussion of the Second Approach

The second approach is intended as a development of the first in a number of ways; a finer grain of assistance is given, and the dialogue is streamlined and made appropriate for more experienced users. Whilst the work is currently unevaluated, initial impressions suggest it is well-received. The next section discusses both approaches in the context of comparable work and the thesis of this discourse.

4 Discussion of Both Approaches

We have described two systems that combine grammar-based and robust approaches to natural language understanding by using robust methods to assist the user in the case where their recognition result is poor. The first is applied to a command-and-control system, and the second, to a question-answering system. Evaluation of the first system has shown positive results in terms of user's increased ability to get recognised by the system and to accomplish a task. It is worth remembering that our sample was restricted entirely to people who had never used the system before. The second system develops on the first, allowing a finer grain of assistance to be given, as well as streamlining the dialogue and making in appropriate for more experienced users.

This work is intended to make the case for combining multiple speech recognisers and appropriate dialogue strategies to handle poor recognition results. Both the suggestions outlined here make use of recognition results that would normally be discarded; the first by using a second recogniser where the primary recogniser fails, and then using that result to select a help message for the user, and the second, by using a recognition result that the system is unable to handle to again provide some assistance, this time in the form of a suggestion. There is no reason why these two approaches, along with other similar ones, could not be combined in the same system. There are enormous possibilities for improving dialogue behaviour in the case of a poor recognition result.

Both the approaches outlined here use multiple recognisers; specifically, the differences in strengths between grammar-based and statistical language modelling are exploited. In the first, the statistical recogniser is specifically used to inform the selection of the help message returned to the user. In the second, the statistical recogniser is used as a back-off to the grammar-based recogniser, and therefore

forms part of the main recognition strategy rather that just a part of the recognition failure handling strategy, though in practice, it is a result rejected by the grammar and accepted by the statistical recogniser that is most often responded to with a suggestion; a result accepted by the recognition grammar often parses with the guide grammar and so is accepted without further processing. (One may wonder why the guide grammar and the recognition grammar differ at all; recall that the guide grammar reflects the system's abilities, whereas the aim in creating a speech recogniser is to recognise as much of what the user says as possible. Therefore the system may on the one hand not know what the user said, or on the other hand, know but be unable to assist. This distinction is evident in human-human communication and can very well be made in human-machine communication as well.) For a more in-depth discussion of the differing strengths of grammar-based and statistical language modelling for speech recognition, see [Knight et al. 2001]. For a further suggestion on how these differing strengths can be exploited, see [Gorrell 2003].

Future directions will include quantitative demonstration of the success of the second approach described here. Work remains to be done in developing the strategy used to select the suggestions. Furthermore, combination of these approaches and other similar ones in one multimedia system is an appealing next step.

References

2003. *Proceedings of Error-Handling in Spoken Dialogue Systems*. Eurospeech satellite event.

Gorrell, G., Lewin, I., and Rayner, M. 2002. Adding intelligent help to mixed initiative spoken dialogue systems. In *Proceedings of ICSLP*.

Gorrell, G. 2003. Using statistical language modelling to identify new vocabulary in a grammar-based speech recognition system. In *Proceedings of Eurospeech*.

Knight, S., Gorrell, G., Rayner, M., Milward, D., Koeling, R., and Lewin, I. 2001. Comparing grammar-based and robust approaches to speech understanding: a case study. In *Proceedings of Eurospeech 2001*, 1779–1782.

Nuance. 2002. http://www.nuance.com. as of 15 March 2002.

RuleQuest. 2002. http://www.rulequest.com. as of 15 Mar 2002.

Bubbles: Navigating Multimedia Content in Mobile Ad-hoc Networks

Erik Bach, Sigrid S. Bygdås, Mathilde Flydal-Blichfeldt*, André Mlonyeni, Øystein Myhre,
Silja I. Nyhus, Tore Urnes†, Åsmund Weltzien, Anne Zanussi‡
Telenor Research and Development,
Snarøyveien 30, 1331 Fornebu,
Norway

Abstract

We aim to support spontaneous and opportunistic human behavior by taking advantage of an emerging environment for mobile ubiquitous multimedia applications enabled by the fusion of ad-hoc networks, peer-to-peer computing, and media-rich mobile devices. Guided by an ethnographic study of spontaneous and opportunistic human behavior, a new concept, called the *Bubble* concept, is proposed that helps users navigate multimedia content made available in mobile ad-hoc networks. The concept is intended to guide the design of user interfaces that provide users with impulses that may trigger spontaneity and opportunism. We used the *Bubble* concept to design and implement a portable audio player application that provides music impulses to users. The application runs on WLAN-equipped iPAQs.

Keywords: spontaneous behavior, ad-hoc networks, multimedia content, user interface.

1 Introduction

Constant progress in hardware and software technologies brings us closer to the vision of ubiquitous computing where information technology becomes an integrated part of our daily lives. One recent technological development is the fusion of wireless ad-hoc networks, peer-to-peer computing, and multimedia content on network-enabled, mobile devices. Local, wireless ad-hoc networks form spontaneously as devices move within radio range of each other, briefly establishing connections before further movements bring devices beyond communication reach [Perkins]. Peer-to-peer computing enables decentralized applications to discover and to exchange resources that happen to be available on peer devices currently connected to the network [Oram]. We are now witnessing the rapid emergence of a variety of new network-enabled, mobile multimedia devices. A major trend is the addition of local networks, digital cameras, audio players, and gaming to mobile phones. Another trend is the addition of local networks and powerful computing hardware to both stationary and portable consumer electronic devices, e.g., cameras and audio/video players. The result of these technological developments is a new environment for multimedia applications, a highly dynamic, decentralized environment spanned by increasingly ubiquitous mobile devices within radio range that spontaneously discover each other and share multimedia content. A user carrying a device in this environment may experience connections and be exposed to subsequent interaction opportunities—typically of highly variable durations—in a completely unplanned and non-deterministic fashion. There may be no permanent (centralized) infrastructure to fall back on when connections are broken due to movements or other reasons.

We are interested in developing applications for this new dynamic environment that spontaneously makes multimedia content on nearby devices available to users. In order to do this we need to take actual human behavior and practice into account prior to developing concepts for new applications. We argue that the said technological environment for multimedia applications has its analogy in normal spontaneous and opportunistic human behavior and practice. Humans continually make spontaneous and opportunistic choices in response to impulses from their immediate environment. This is a type of practice that by nature is fundamentally unpredictable and ad-hoc. Knowledge about the processes of human spontaneity, choice making, and interaction is therefore fundamental and should inform both the design of applications and how information is presented to users.

We conducted an ethnographic study to guide the development of a concept, the *Bubble* concept, to support users in navigating multimedia resources in wireless ad-hoc networks. A key finding from the ethnographic study was that unplanned events and spontaneous actions lead to a constant redefinition of one's self-understanding and goals, in addition to the identity one seeks to convey to the surroundings. This, we argue, implies a need to convey to users the full nature and extent of the resources currently available in the environment, but to do so in a way that is un-confusing and easily navigable.

An information bubble metaphor is central to how we expose users to content sources briefly available in the environment. The metaphor derives from the notion that the extent of the short-range radio waves emitted from a mobile device may be pictured as a sphere. Advertised content stored on a device is available inside its information bubble; when a device moves, its information bubble moves with it. When two devices move within radio range of each other, their information bubbles merge and the combined advertised content of the two devices becomes available to both. Conversely, when two devices move away from each other, causing a connection break between the two, content belonging to either device will no longer be available to the other (their information bubbles become disjoint). The sudden appearance of information bubbles enables users spontaneously and opportunistically to access any content of interest before the inevitable, and equally sudden, disappearance of the same

---

† e-mail: tore.urnes@telenor.com
‡ Presently at Posten Norge
* Presently at the University of Oslo

information bubbles. Content access, in the context of our Bubble concept, translates technologically into the direct transfer of content between peers.

We believe that the *Bubble* concept is a promising approach for keeping users up to date about the overall nature and fluidity of multimedia content available in the nearby surroundings. Unfortunately, the limited rendering resources of a small-size mobile device greatly reduce the level of completeness and detail that may be visualized. Profiles are a common tool to help reduce the effects of information overload on users of mobile devices. However, we argue that the concept of profiles runs counter to our aim of supporting spontaneity and opportunism. Instead we propose to extend our *Bubble* concept with an alternative heuristic mechanism more tailored to generate impulses in users. This mechanism randomly and continuously selects atomic units of multimedia content currently available on other devices and briefly renders a description of the selected content.

Based on our proposed *Bubble* concept, we designed an audio player that allows users to navigate available audio resources in an ad-hoc networking environment. We present the user interface design and report on an expert evaluation of the interface. A technical prototype of the application has been implemented and tested with a small number of users.

This paper is organized as follows. We start by briefly describing related work. Then, as a motivation for the *Bubble* concept, we go into some detail about spontaneous and opportunistic behavior and report on the findings of an ethnographic study that we conducted. A more in-depth analysis of the relationship between the elements of the *Bubble* concept and their theoretical underpinnings is given next. Then, we give an account of the design and evaluation of the user interface of our audio player application. We also give a brief description of the technical prototype that we implemented. Finally, we offer our concluding remarks.

2 Related work

This section presents related work. We start by considering a prototype application that enhances the experience of chance encounters and whose design was guided by ethnographic fieldwork. Next we present two applications aiming to support opportunistic interaction. Then, we compare our work to work on social, location-based information spaces that are open to all mobile users. Finally we describe two recently proposed application concepts for joint music listening in mobile ad-hoc networks.

Hocman is a prototype mobile application for motorcyclists designed to enhance the social experience of encountering other motorcyclists on the road [Esbjörnsson]. Basically, Hocman prototypes detect each other when spontaneous connections are established in mobile ad-hoc networks and the establishment of a connection is signaled to the driver as an audible icon. During such brief connections on the road, information in the form of a simple web page is exchanged. The road encounter is enhanced in two ways: both by the early alert and by the knowledge that information about the other driver will be available for browsing. The researchers behind Hocman chose to conduct an ethnographic study of motorcyclists and how they encounter each other on the road. The results from the study informed the design of Hocman. Similarly, we also use ad-hoc networks, peer-to-peer exchanges, and mobile devices with multimedia content to support specific human behavior, i.e., spontaneity and opportunism. We also decided to conduct an ethnographic study to better understand the behaviors and practices in question. Our focus differs from that of Hocman in that we aim to generate impulses to be acted upon immediately rather than at a later time.

The Aquarium is an information retrieval metaphor and application for opportunistic exploration of on-line stores [Bryan]. The main aim of the Aquarium metaphor is to find a fun way of exposing millions of products in a store's inventory so as to agitate active or latent interests in the user or to create new interests. The proposed metaphor lets pictures of random inventory move across a screen like fish in an aquarium. The pictures are tied to product categories and simple operations allow users to ask for more or less of the category in question. A main challenge is to adjust the width and scope of categories to match the intensions of users. As a metaphor, an aquarium is an interesting alternative to information bubbles, though we deal with a far less centralized and structured content base. An interesting similarity to our work is the mechanism of picking random items from a vast inventory to generate impulses.

Proxy Lady is an application for mobile ad-hoc networks that aims to facilitate pre-meditated opportunistic communication between users [Dahlberg]. Basically, a user A decides that she wants to talk to user B and tells Proxy Lady to notify her should she happen to come within (radio) range of user B during the day. Our *Bubble* concept does not include the notion of subscribing to notifications. This is mainly because we aim to support opportunism in a more spontaneous (i.e., less pre-meditated) setting than Proxy Lady.

The decentralized, almost anarchistic nature of information spaces built with peer-to-peer computing technology on top of mobile ad-hoc networks invites users to advertise personal multimedia content to the world around them. This leads to a social and dynamic information system similar to that envisioned by researchers working on GeoNotes [Espinoza]. GeoNotes is concerned with digital, location-based spaces that are open to anyone to leave their "notes", e.g., messages or statements. The group behind GeoNotes predicts that this could lead to information overload and that users risk being bombarded with heaps of unwanted, irrelevant information. Filtering is therefore a key focus of GeoNotes. However, lack of experience from real usage makes it hard to judge the magnitude of the information overload problem in information systems focused on the immediate, local setting. As our focus is on generating impulses to support spontaneous and opportunistic behavior, we try to avoid explicit filtering. Nevertheless, our *Bubble* concept does provide some means of limited, implicit filtering, i.e. rendering all resources from a single device as one information bubble and picking a few random items from the entire, available content universe to hint at its flavor. Also, in a similar fashion to GeoNotes, the *Bubble* concept may allow each user some control over how their information bubble is rendered on the devices of other users (making a statement), in effect creating a kind of social filter.

[Axelsson] and [Bassoli] introduce the concept of mobile music listening as a shared, social activity. The idea is to equip portable music players with ad-hoc networks and to allow nearby users to connect to each other's players to hear what is currently being played. [Axelsson] has focus on music players in vehicles and aims at providing entertainment for occupants of cars. [Bassoli] envisions an urban setting where shared music experiences help form local communities. Music is also a central theme in the portable audio player we designed and implemented to test our

Bubble concept. Our player currently does not include the notion of joint listening since we focus primarily on supporting spontaneity and opportunism. We do, however, consider support for socializing an interesting feature and our *Bubble* concept provides some support for it through identity statements.

3 Spontaneous and Opportunistic Human Behavior

We are interested in supporting spontaneous and opportunistic human behavior. At an early stage, we saw a strong relationship between such behavior and characteristics of emerging ubiquitous computing environments based on mobile ad-hoc networks and peer-to-peer computing. We will now briefly describe some of those characteristics before taking a closer look at the nature of spontaneity and opportunism.

Ad-hoc Networks and Peer-to-Peer Computing

Mobile ad-hoc networks are spontaneous, self-configuring, wireless networks with no fixed infrastructure. Connectivity in an ad-hoc network is governed by distance and radio range. As devices move about they are able to detect and connect to other devices that are in sufficient proximity. When connected devices move outside of radio range, connections are broken. This means that in ad-hoc network settings, mobile devices come and go, and thus create a highly dynamic network structure. The local aspect of ad-hoc networks must also be emphasized, i.e., that typical radio ranges typically stay well within about 100 meters. We therefore say that ad-hoc networks are geographically situated locally. This attribute of ad-hoc networks relates them strongly to the situated-ness of human spontaneity, a point that will be further discussed below. The decentralized nature of ad-hoc networks fits well with the principles of peer-to-peer computing. Peer-to-peer computing refers to a class of applications that enables users to form logical networks on top of any infrastructure and to share and exchange digital content. Interactions between peers in a peer-to-peer network are by definition independent of central servers. Pure peer-to-peer networks also allow peers to join the peer-to-peer network and to discover peer resources without a server infrastructure. As such they are similar in spirit to ad-hoc networks and emphasize the autonomy of the individual user in being able to connect to and to exchange content directly with any other user in the network. Combined, ad-hoc networks and peer-to-peer computing lay a foundation for spontaneous digital information exchange and communication between nodes in a very rapidly changing environment.

The following scenario further illustrates the technical environment of this paper. Imagine people gathering at a bus stop with their mobile devices. Instantly, an ad-hoc network is forming, and as people leave with their buses and others join the crowd, the topology of the ad-hoc network changes. Also, multimedia content such as pictures, music, video, etc., may be available on devices for others to discover using peer-to-peer computing mechanisms. A highly dynamic and complex landscape of resources has been established, changing continually and unpredictably, perhaps even as a user is trying to access another user's content. From the individual user's point of view, resources appear and disappear, offering impulses to be acted on quickly or forgotten.

Exploring Spontaneity and Opportunism

We believe that the emerging environment for multimedia applications described above also has a strong foundation in social life, namely spontaneous and opportunistic behavior. Spontaneity refers to that human state of mind where choices and actions are made voluntarily based on momentary impulses; while by opportunistic behavior we mean taking advantage of opportunities as they arise, i.e., grabbing what is offered if it catches your interest. This type of behavior is basically unplanned, not necessarily (though often) guided by social demands like norms and trends, and responsive to the impulses we are continually exposed to, as we perceive our surroundings. This behavior is always situated, i.e., impulses are perceived here and now, and are reacted upon (or not) in a geographical, local setting. This is true simply because we as human beings are physical in essence, and thus always located. Our perception of the world therefore always includes a perception of our immediate context. An important question arises next: How do people make spontaneous choices, and what does this imply for our concept?

We performed an ethnographic study to examine this question more closely [Bach]. Through a qualitative microanalysis[1] we have investigated the relationship between what people say they are going to do and what they actually do, and how meaning is formed in situations where something unforeseen happens. The research was conducted through a participant observation of two core informants during several weeks. In addition to this, several interviews were conducted. Through participant observation we were able to identify the meaning related to actions and behavioral patterns, which the informants normally took for granted in daily life.

Our findings show that unforeseen events and actions during the day were redefined and became legitimized to fit the informant's understanding of his or her surrounding whole (context). In other words, the informants redefined their contexts and ultimate goals to accommodate surprising events or unplanned actions. The whole, the ultimate goal of the informant, was thus continually refitted to accommodate its unexpected parts, and these parts thus became meaningful to that person. There is, in other words, a strong relation between a person's understandings of context, i.e., its meaning, and the spontaneous, opportunistic choices that person makes.

An empirical example from our fieldwork describes this redefinition of ultimate goal: X is planning to watch a soccer match later the same day. His ultimate goal is to watch the game at night with his friends. During the day, several incidents occur and he watches the match with a completely different crowd. The meaning X ascribed to the football match was thus redefined because of the incidents during the day [Bach]. Following this argument, the meaning people assign to a predefined goal has to be seen in relation to unexpected events that occur: these events are continually re-contextualized. By neglecting to consider this hermeneutic circle, we run the risk of missing the meaningful aspects people give social contexts.

We now look at implications of the said findings in terms of our goal of supporting spontaneous and opportunistic behavior through the use of a technological environment based on a fusion

[1] A qualitative microanalysis is a type of methodology that covers a very large number of variables for a small number of informants. The 'micro' in microanalysis does not mean a small study, but rather a study of many minute details [Pelto].

of ad-hoc networks, peer-to-peer computing, and media-rich mobile devices. Firstly, we argue that the environment must be flexible and open enough to tolerate the continual redefinition of context intrinsic to spontaneous behavior. Also, this type of behavior implies a constant redefinition of one's self-understanding and the identity one seeks to convey to one's surroundings. The environment must therefore be as flexible and distributed as possible, and avoid pre-planning, filtering and central control. Secondly, we argue that the environment must be able to convey the nature and complexity of the context (the landscape of multimedia content) to the user, but at the same time making it intelligible and navigable. There is potentially a danger of information overload. However, filtering out information based on pre-conceived notions about relevance to the user in a particular context should be avoided. The rendering of the context should reflect its high level of dynamism. This is important to ensure a closeness between a user's perception of his or her surroundings and that user's understanding of the visual representation of these surroundings.

4 The *Bubble* Concept

In the last section we arrived at the conclusion that in order to provide application-support for spontaneous and opportunistic behavior, we need to solve the problem of simple and effortless navigation of multimedia resources available in ad-hoc, peer-to-peer environments. To solve this problem we have formulated the *Bubble* concept. Central to this concept is the vision of users carrying mobile, media-rich devices able to engage in spontaneous communication and content exchange. Devices and their multimedia content can be thought of as comprising information *bubbles*, i.e., as resources surrounded by a sensitive sphere, that, when overlapping with other information bubbles, would connect and facilitate communication and content sharing (see figure 1).

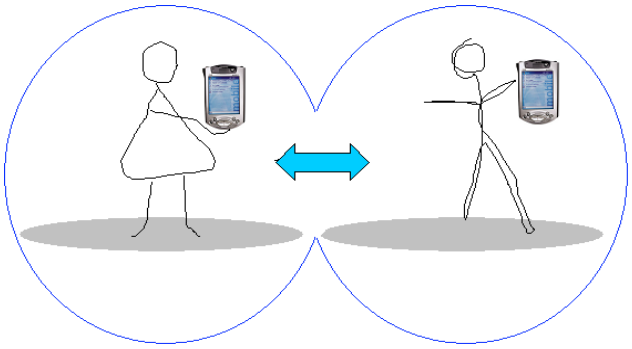


Figure 1: Overlapping information bubbles facilitate sharing of content.

The main mechanisms underlying the *Bubble* concept are those of discovery of other information bubbles and interaction in the form of content browsing and sharing. These mechanisms are always rooted in the local setting. The *Bubble* concept also includes the idea of letting users express their social identities and social group memberships. In practice this is done through the possibility of choosing or designing one's own iconic representation, and enabling users to make public statements such as grafitti, tags, or exclamations. In the Bubble system, this feature has some valuable navigational significance (see below).

A central question in systems that aid a user in navigating and understanding the immediate context is how to build systems that filter information often available to users and offer only that which the user wants. A common solution to this problem is to make use of user profiles that informs the system of the user's preferences. In relation to the behavioral context of spontaneity and opportunistic behavior, however, the use of user profiles is problematic: predicting or deciding how, when or where to be spontaneous in advance is impossible. In principle, therefore, user profiles work counter to the idea of spontaneous behavior. Furthermore, the constant redefining of identity that a user executes as he or she moves through various contexts makes it hard to predefine a given user profile; the profiles themselves changes as users move through different social contexts. This basic sociological insight has been illustrated by Erving Goffman [Goffman], and can be exemplified by the way most of us communicate radically different personalities to the world at work and at home. Moreover, the idea of an environment based on ad-hoc principles is incompatible with the idea of pre-defined user profiles. A system has the same problem as the individual user: it cannot predict when and where spontaneous behavior will occur. In addition, the continual redefinition of the context itself by the user further complicates such predictions.

The question of how to help users attain a minimum of context awareness still remains, though. For the *Bubble* concept this is an issue that we continue to work on. For the time being this problem rests heavily on users themselves. Firstly, as we have argued, there is a strong connection to actual place in the *Bubble* setting. The range of the radio technology used in the system and the demands of immediacy and spontaneity firmly roots the *Bubble* concept to a limited geographical area. This may suffice as an information filter for many applications, i.e., leaving it to users to decide which impulses are relevant. Secondly, the social identity features of the system work as an information filter and navigational aid. Social identity markers, like music taste or use of slang, are important ways of showing the rest of the world who we are and where we belong. This is a type of knowledge we use daily when we categorize, sort, and filter the impulses and information around us. As such, identity markers and statements can be used as a basis to choose the information one finds interesting in a particular situation. In other words, it is a simple filtering mechanism based on social navigation (see for example, [Forsberg], [Höök], or [Munro] on social navigation issues).

The process of perceiving and conceiving of the surroundings is a semiotic process. David Benyon has conceptually divided the world we live in into an information space and an activity space [Benyon]. These different but interconnected quantities represent an attempt to express this semiotic process. As a person perceives something in the activity space, this something is interpreted, contextualized, and finally understood, i.e., conceived and fitted into the information space. The information space is therefore based on the activity space, but also vice versa. As we conceive, we reconfigure our information spaces and thereby allowing ourselves to perceive our activity space from a new perspective. This semiotic process is at the hart of our proposed *Bubble* concept's context awareness. It is this process that the system is meant to support, namely to facilitate an understanding of one's context based on how this context changes as the available information changes. This process is at the heart of making spontaneous choices.

Limitations of the Bubble concept

We believe our *Bubble* concept is well suited as a foundation for user interfaces that expose users to impulses, hence supporting spontaneous and opportunistic behavior. We are, however, aware

that the *Bubble* concept may suffer from two potential limitations. Firstly, it is central to our vision of supporting spontaneous and opportunistic behavior that as little information as possible in the *Bubble's* vicinity be filtered away. This lack of information filtering may result in a situation of information overload, e.g., at big gatherings such as concerts. Secondly, the *Bubble* concept is targeted at a mobile setting while at the same time requiring the user's attention in order to communicate impulses. Currently, the *Bubble* concept provides little help for solving this user interface challenge of communicating impulses to mobile users without requiring too much attention. The severity of these potential limitations can only be determined through evaluations that we hope to conduct as part of future work.

5 An Audio Player Providing Musical Impulses

As a demonstration of our proposed *Bubble* concept, we now present the design and implementation of a prototype audio player application that provides users with musical impulses. The idea behind the application is that of a mobile audio player capable of discovering other player applications and sharing music playlists and other content. The goal is to provide users with music impulses by exposing available playlists and letting users navigate those that catch their interest.

User interface Design

The user interface should be dynamic and flexible without confusing the user. We have chosen a fairly large color display on a portable device as the medium of the user interface.

The audio player application's core feature is to exchange playlists and music, but content such as pictures and winks [Microsoft] are also supported. In this description we will focus on how users spontaneously connect to other bubbles and how content is exchanged.

Figure 2: The start-up screen

Figure 2 shows the main screen of the Bubble application. Note that all screen-shots are mock-ups; the complete user interface is still not implemented and tested on real users[2]. The screen is divided into three areas. The main area in the middle shows other bubbles within range. According to the physical locations and radio ranges of nearby users carrying wireless devices, bubbles may appear and disappear, thus the picture is reflecting the dynamic nature of the environment. The bubbles are placed randomly on the screen. The screen shows up to 7 bubbles at the same time, but the number of bubbles within range may be higher. The user may therefore use a zoom-tool to get a full overview of all bubbles.

The area on the top is a ticker, where the user is exposed to a random sampling from all the MP3-files that are advertised in playlists. The list moves from the right to the left. The user may select one of the icons and the MP3-file will immediately start streaming. The user can then decide to download the file or to go directly to the Bubble from where it is streaming and check out other content. This is a bottom-up approach meaning that the user is presented directly to the content. The ticker provides impulses that may trigger spontaneous or opportunistic behavior from users.

The area at the bottom is a collection of functions such as a shortcut to the user's own files, setting of configuration properties, searching in the currently available content, zooming, and help. The functions are context-aware and adapted to the task the user is performing.

The bubbles in the middle area are the main door step to content. To each icon, is augmented with information about how many files are available in each of three different categories: music, pictures and winks. The users themselves design the icons that represent the bubbles. This gives the users an opportunity to reflect their identity and give other users an idea of what kind of interest they have. This gives a user a first impression as to whether the content correspond to his or her interest. Furthermore, when a bubble is explored by clicking on it, a new screen shows more detailed information about the bubble and the content. Here the users can express themselves by writing a statement to show their identity. An example is shown in figure 3. From this screen the user can decide if he or she wants to explore and download content or go back to explore other bubbles. This method of navigating is based on identity, appearance, fashion, social statement and culture and is inspired by social navigation.

The functions area of the statement screen is different from that of the start-screen. A new function for getting in touch with the user in question by sending a wink and start chatting has been added. The zoom feature is not relevant in this context and has been removed from the function-list.

[2] Though, a subset of the interface has been implemented in a technical prototype.

Figure 3: Identity statement

If the user decides to explore the content of music files, the screen in figure 4 will appear. It is a simple list of MP3-files sorted alphabetically. Information about size and quality of the file is attached to each file. When a user clicks on the filename, the corresponding song (or perhaps a short thirty second, say, version) starts streaming across the network. This helps the user decide whether to download the song or not. To download one-by-one, the user selects a song and then clicks on a download button. It is also possible to download all music files by clicking on a single button.

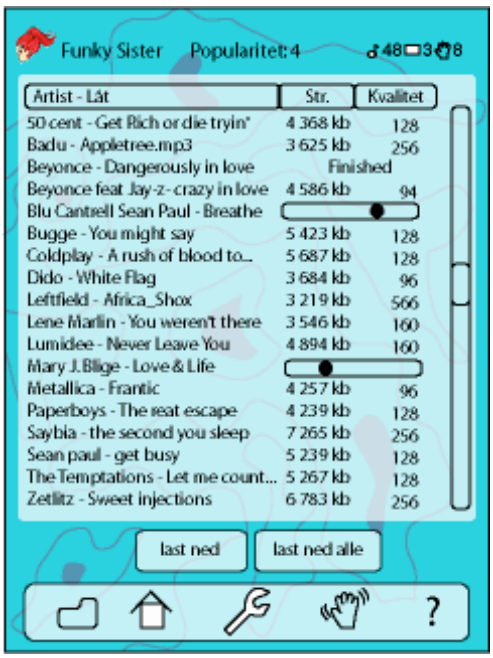


Figure 4: Exploring and downloading content

Expert Evaluation

The process of user interfaces design is iterative. In order to get feedback in the early iterations we conducted an expert evaluation. We ran three workshops where user interface experts and designers worked together in pairs. They were first presented with a scenario that explained the core functionality of the audio player application and then presented with mock-ups of screen-shots. Evaluations focused on metaphors, navigation and interaction design.

The feedback from the workshops was positive with regards to how we resolved the *Bubble* concept in the user interface. The evaluation resulted in several suggested changes related to interaction design and graphic design. The overall concept and metaphors received the approval of the experts.

System Description

To perform user evaluations, we implemented a technical prototype of the audio player. The prototype currently runs on WLAN-equipped iPAQ PDAs (see figure 5). Ideally, we would have preferred to run our prototype on mobile phones and portable MP3 players, but such devices currently lack support for wireless ad-hoc networking. Many advanced mobile phones currently support the Bluetooth local network interface, and also come with advanced software development platforms, but ad-hoc networking is not supported sufficiently (Bluetooth is a point-to-point protocol lacking support for IP broadcast-based discovery).

The prototype is implemented in C++ for the PocketPC 2002 operating system. For peer discovery and exchange of playlists, we use OpenTrek [OpenTrek], a middleware designed for mobile multiuser gaming in wireless ad-hoc networks. OpenTrek is based on a fully decentralized, peer-to-peer architecture and offers a notification-driven programming model. OpenTrek constantly monitors the ad-hoc network and issues a notification every time a device running our prototype enters or leaves the network. The OpenTrek network protocol, in addition to peer discovery, also offers limited support for the transmission of C++ objects, and we use this feature to exchange playlists and other metadata between peers.

Figure 5: The audio player prototype

The GapiDraw [GapiDraw] graphics toolkit is used to render the user interface. GapiDraw gives direct access to the device display and offers a range of powerful graphics commands, including limited animation support. We implement MP3 audio file streaming through a custom extension of the PocketPC version of the FMOD sound system [FMOD]. Access to MP3 files is through

HTTP servers running on each device. Figure 6 shows an overview of the implementation architecture.

6 Conclusion

In this paper we have reported on the development of the *Bubble* concept as it was guided by our own ethnographic findings on spontaneous human behavior. The development of this concept has been motivated by the emergence of a new environment for multimedia applications. This environment is highly dynamic, decentralized, and spanned by increasingly ubiquitous mobile devices within radio range that spontaneously discover each other and share multimedia content. It is also an environment that relates naturally to spontaneous and opportunistic human behavior.

We argue that in order to develop good multimedia applications for this dynamic environment, we need to base or concepts on studies of actual human behavior and practice. As a foundation for the development of the *Bubble* concept, we conducted an ethnographic study of spontaneous choice-making. Our assertion was that this type of behavior could be a good analogy to the said dynamic environment as it is basically unpredictable and ad-hoc. One of our principal findings in this study was that unplanned events and spontaneous actions lead to a constant redefinition of the way we understand our contexts, our goals, and ourselves. This insight, we have argued, implies a need to convey to users the full nature and extent of the resources available in their environment. However, this must also be done in a way that makes this complex environment less confusing and more easily navigable.

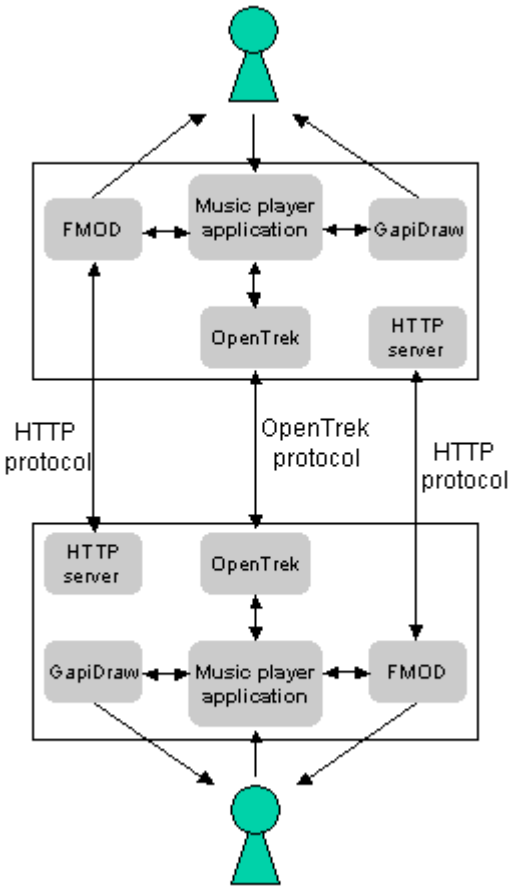


Figure 6: Overview of implementation architecture

We have developed the *Bubble* concept to meet the special requirements of spontaneous behavior in highly dynamic settings. In this concept, an information bubble metaphor is central to how we expose users to sources of content that are only briefly available. As information bubbles join and disjoin, the content available on the corresponding mobile devices becomes available and disappears, respectively, to others. The sudden appearance of information bubbles enables users to spontaneously and opportunistically access digital content from their immediate surroundings before the content just as suddenly disappears.

Our belief is that the *Bubble* concept is a promising approach for keeping users up to date about multimedia content in such a dynamic environment. To test this assertion we have made a technical prototype of an audio player that allows users to navigate available, local audio resources according to the principles of the *Bubble* concept. The user interface of the audio player emphasizes strongly the dynamics of the user environment and has been optimized for immediacy and ease of use. Our hope is to be able to deploy a full prototype of the audio player on mobile phones and portable MP3 players with support for ad-hoc networking when these become available. This would allow us to conduct field trials of the *Bubble* concept. At this stage, evaluation amounts to expert evaluations of the user interface design.

Acknowledgements

We are grateful to Sven Inge Bråten for contributing valuable ideas and inputs to the work presented in this paper during his tenure as a Master's student. We would also like to thank user interface evaluators Kari Hamnes, John Rugelbak, and Heidi Rognskog from Telenor R&D for their help with the development of the user interface. Thanks also to Nina Khalayli for helping us with the graphical design of the interface.

References

AXELSSON, F. AND ÖSTERGREN M. SoundPryer: Joint Music Listening on the Road. In *Adjunct Proceedings of Ubicomp 2002*, Göteborg, Sweeden, 2002.

BACH, E., BLICHFELDT, M. F. AND WELTZIEN, Å. *Spontane valg og meningsdannelse*. (In Norwegian), Telenor R&D N39, 2003.

BASSOLI, A., CULLINAN C., MOORE, J. AND AGAMANOLIS, S. TunA: A Mobile Music Experience to Foster Local Interactions. Interactive Poster at *Ubicomp 2003.* Seattle, Washington, USA. 2003.

BENYON, D. A Functional Model of Interacting Systems; A Semiotic approach. In Connolly, J. H. and Edmonds, E.A . (eds.) *CSCW and AI,* Lawrence Erlbaum. 1993.

BENYON, D. Beyond Navigation as Metaphor. In PROCEEDINGS *of 2nd EuroDL conference*, Crete, 1998.

BRYAN, D. AND GERSHMAN, A. The Aquarium: A Novel User Interface Metaphor for Large, Online Stores. In *Proc. 2nd Intl. Workshop on Web-Based Information Visualization* (WebVis 2000). 2000.

DAHLBERG, P., LJUNGBERG, F., AND SANNEBLAD, J. Proxy Lady – Mobile Support for Opportunistic Communication. *Scandinavian Journal of Information Systems*, Vol. 14, pp. 3 – 17. 2002.

ESBJÖRNSSON, M., JUHLIN, O. AND ÖSTERGREN, M. The Hocman Prototype: Fast Motor Bikers and Ad Hoc Networks. In *Proc. of MUM 2002*, Oulo, Finland. 2002.

ESPINOZA, F., PERSSON, P., SANDIN, A., NYSTRÖM, H., CACCIATORE, E., AND BYLUND, M. GeoNotes: Social and Navigational Aspects of Location-Based Information Systems. In *Proc. Ubicomp 2001*, Atlanta, Ga., USA. 2001.

FMOD: http://fmod.org/

FORSBERG, M., HÖÖK, K. AND SVENSSON, M. Design Principles for Social Navigation Tools. In *Proc. UI4All*, Stockholm, Sweden. 1998.

GapiDraw: http://www.gapidraw.com/

GOFFMAN, E. *The presentation of self in everyday life*. Penguin Press, London. 1971.
HÖÖK, K. AND DAHLBÄCK, N. *Designing Navigational Aids for* Individuals. Position paper presented at the *CHI'97 Workshop on Navigation in Electronic Worlds*, Atlanta, GA, 1997.
Microsoft Three Degrees: http://www.threedegrees.com/
MUNRO, A.J., HÖÖK, K. AND BENYON, D.R. *Social* Navigation *of Information Space*. Springer, London. 1999.
OpenTrek: http://www.opentrek.com/
ORAM, A., editor. *Peer-to-Peer: Harnessing the Power of* Disruptive *Technologies*. O'Reilly Press. 2001.
PELTO, P. AND PELTO, G. *Anthropological research: The* structure *of inquiry*. Cambridge University Press, Cambridge. 1978.
PERKINS, C. *Ad Hoc Networking*. Addison Wesley Professional. 2000.

A Service Oriented SIP Infrastructure for Adaptive and Context-Aware Wireless Services

Wei Li
Department of Computer and Systems Sciences
Royal Institute of Technology
liwei@dsv.su.se

Abstract

Due to the variety of widespread network communication technologies, a user has more possibilities to access various services. However, having alternative networks and services does not bring ease to the user immediately, but often results in increased burdens in terms of repetitive configuration and selection work, although the user is only interested in the actual use of the appropriate services. Our project tries to attack this problem, aiming to facilitate the user's (in particular the mobile user's) ability to make use of different services in an affordable way, based on adaptive services which exploit their awareness of the user's context. In this paper, we propose a service-oriented context infrastructure, to simplify the exchange of context among services, thus facilitating users and applications access of services in an efficient way. The Session Initiation Protocol (SIP) and its sibling protocols were adopted for transferring context information (encoded in XML) within our infrastructure.

Keywords: context-aware, service-oriented infrastructure, SIP

1 Introduction

The computer has to this date been with us for more than a half century. During this period, it has experienced a rapid development in many aspects; from increasingly powerful microprocessors to decreasing energy consumption, physical size and price. Affected by the Internet spreading almost all over the world, the computer today is becoming a crucial tool for accessing and exchanging information across the limit of physical boundaries. The promising goal "access information everywhere anytime" has further paved the way for the inhabitation of computers in our daily life by riding the tide of wireless network technologies (such as Wireless LAN, Bluetooth and GPRS), and has also shaped the computer itself into mobile and wearable fashions like laptops, Tablet PCs, personal digital assistants (PDA) and smart phones. As computers become ubiquitous, people find themselves immersed in a world full of computing power.

However, the availability of rich computing resources does not immediately mean usefulness. For example, a nearby resource in a Bluetooth network cannot be accessed directly through a Wireless LAN or GPRS network without a third party standing in between and acting as a proxy or gateway, since inherently they are not compatible networks. On the other hand, a user may in many cases have several personal devices (e.g., a laptop, a PDA and a smart phone) and there may be additional resources like a desktop computer, a projector and a printer available in the surrounding environment, such as a meeting room. It becomes a critical issue how to "tame" so many different devices to work together fluently without burdens of configuration overwhelming the user, especially when the user is new to the environment and has less knowledge about computers and other devices.

A service-oriented perspective can simplify the problem by presenting different resources (including devices) in the form of services. The user will only see a printing service and a projecting service available in the case above, thus hiding other complexities such as devices and networks. This service-oriented perspective also introduces an emerging computing paradigm - Service Oriented Computing, evolved from mobile and distributed computing, and the object-oriented and component-based computing. In this paradigm, the whole system is constructed by services (with levels) as the building blocks, which are autonomous, platform-independent computational elements and also have capabilities to communicate with each other across networks. Service-oriented computing brings more concrete hints on how to reuse previous work than component-oriented computing since a service normally embodies more complete application logic and often appears as a runtime entity. Along with the service-oriented computing paradigm, many efforts such as Web Services [W3C 2003], GRID computing [Ian 1998], as well as peer-to-peer network technologies work together to envision a future computing environment where computing resources in terms of services will be available to be used as conveniently as today's electricity.

Nevertheless, the term "service" has been used too extensively, and now suffers from a less concrete meaning. Today, we call many things services: network service, printing service, … to discovery service. As a consequence, it is difficult for the user to grasp the relationships between them. For instance, a printing service will not be available without a network service and the ability to discover the printer. Without a good arrangement and guidance for this multitude of services, users can easily get overwhelmed by the great number of services even though they are not interested in the distinctions and implications among them, but rather the immediate use of the interesting services.

Our work in Adaptive and Context Aware Services (ACAS) project tries to attack these problems, aiming to facilitate the user's (in particular mobile user's) ability to make use of different

services in an affordable way based on adaptive services – which exploit their awareness of the user's context. In this paper, we will first give an introduction to context-aware computing and why it is meaningful for mobile users, together with some problems of using context information. Then in the next section, we present our context stack (a layered structure) to help the understanding of context information acquisition. In section 4, we will talk about the mechanism of aggregating and distributing the acquired context information. We will also elaborate how we adopt the Session Initiation Protocol (SIP) and its sibling protocols to build our context delivery infrastructure conforming to a service-oriented architecture. Then in section 5, we will present our prototype application scenario, which exploits the benefits from our context delivery infrastructure. Finally, we will refer to some related work, and discuss some relevant issues and future work.

2 Context-Awareness Computing

Context is a longstanding concept in the human-computer interaction area where it is thought implicit (e.g., a quiet corridor), as opposed to a user's explicit interaction with a computer, which conventionally consists of actions like moving a mouse or pushing a button. The idea behind context-aware computing is to support computer human-literate by feeding in various background information (so-called context) so that the computer can adapt to the user and her situation accordingly. For instance, a mobile phone with location awareness would switch to vibrating mode automatically when its carrier enters a concert. The main purpose of using context is to facilitate user's interactions with computers by replacing some explicit interactions with automatic system interactions, thus reducing attention distraction of the user. This kind of adaptability has great significance for mobile users who are usually willing to pay less attention to the management of a changing environment.

Context-aware computing has been a vital research area, although there are no agreements on the definition of context yet. Many researchers have proposed different interpretations, among which we prefer the most comprehensive one – "context is everything but the explicit input and output" to a system [Henry et al. 2002]. Fortunately, the notion of context has been commonly understood as the physical and social situation where the interaction takes place. It is often associated with primary questions like "When", "Where", "Who" and "What", or in other words: time, location, entity and activity. This also appears to include the most useful context information. In addition to these, we think computing status is also one type of important context, such as the running program and network status (there are actually measured by local processes).

Although there have been many context-aware systems and applications tested since last decade, most of them are still prototypes and only available in research labs or academic world. The common problems with context-aware system lie in the complexity of capturing, representing and processing the contextual data [Pascoe et al. 1997], because the adaptability of a context-aware system will only be appreciated if the context can be interpreted correctly. However, computer technology to date can only try to approach user's intention by "guessing" based on predefined rules applied to different context data and historical behaviors. Another critical issue, which we believe has affected the spreading of context aware systems, is the difficulty of distributing acquired context information broadly among massive context-aware applications across different networks. In a later section, we first present our context stack to help to solve the problem of context acquisition and processing, and after that, we elaborate our context distribution architecture based on SIP Presence framework [IETF 2003].

3 Context Information Acquisition

Sensor technologies have been widely used to capture context information in the context-aware computing arena. The most well-known ones include the early IR-based Active Badge [Want et al. 1992], RF tags in DUMMBO [Salber et al. 1999], the later GPS receiver in Cyberguide [Long et al. 1996], digital camera in StartleCam [Healey et al. 1998], and today's compound sensors: iButton [Dallas Semiconductor] can store any information (with a limited capacity) as context information, such as Identity; and the Mica Motes [Crossbow 2003] (a more advanced sensor board developed by UC Berkeley) can accommodate many different sensors on one board including thermometer, accelerometer, magnetic and photo sensors. The Mote also has the capability to communicate with other sensor boards to create a dynamic wireless sensor network.

The wealth of sensor technologies brings more possibilities for context-aware systems. On the other hand, it makes context information acquisition more complicated, especially when the same context information (in different forms) can be retrieved from different sensor data-sources. Just as most communication means based on radio signals transmitting (like GSM, Wireless LAN, Bluetooth), can be used for positioning and location information, the same sensor data can also be used to give different context information. For example, a GSM phone can simultaneously give presence, location and identity context, a camera can detect motion to tell the presence of people in the room, and can be used to identify users (with facial recognition technology) as well. In such cases, it becomes more puzzling for the system to retrieve the useful and correct context information because adding sensors also adds noise and introduces the possibility of apparent contradictions.

3.1 Logic and physical sensors

There have been many different categories for context information in the field to simplify the understanding and the use of context from different perspectives, such as active vs. passive context [G. Chen et al. 2000] according to whether an application will react to the context changes or not. One of the most accepted divisions of context is to distinguish context as sensed context (retrieved directly from sensors) or derived context which cannot be acquired directly from sensors, but is obtained through inference (also called context fusion, e.g., an activity context may be inferred from a fusion of context information based on people, time and location). The divisions are sometimes also called physical and logical sensors respectively. As one of the results of our survey in the context-aware computing area, we strongly feel the absence of a standard view to support discussion and analysis of context information thoroughly. This view should be more comprehensive than the two-category division into physical and logic sensors.

3.2 The Context Stack

We propose a layered Context Stack like figure 1, along with a service-oriented perspective to promote the understanding of context information acquisition. This view was inspired by the Open System Interconnect (OSI) seven-layer network model and the work of the Location Stack [Jeffrey et al. 2002] as well. We

also borrow some terms from the Location Stack, but recast with our definition to support broader context information.

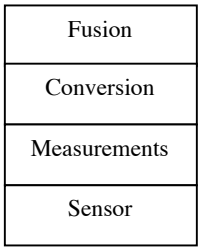


Figure 1. The Context Stack

- **Sensor layer:** Contains the physical hardware (sensor) and its corresponding driver (in some cases some embedded algorithms/firmwares as well). The sensor layer offers the upper layer access to "raw" sensor data, such as the NMEA sentences from a GPS receiver.

- **Measurement Layer:** Contains algorithms to interpret raw data into "canonical" data types, which is more meaningful and practical to general applications. Taking the GPS receiver for example, the Latitude and Longitude should be deduced in this layer by parsing the raw NMEA sentences.

  As many researchers also pointed out that *history behaviors* should also be included as context in some cases, we think it is appropriate to have this kind of context in this layer, such as a data set to plot a curve of temperature changes in a specific time-span.

- **Conversion Layer:** The measurement layer normally offers sensor data in limited (canonical) forms. We think it is very important to have an abstract layer taking responsibility of converting the context data into different formats/units. For example, a software service can convert temperature from Kelvin to Celsius or Fahrenheit. The conversion layer can have a chain of services to support conversion between many different units. How to determine the optimal path from several possible service-conversion chains becomes an interesting issue per se.

- **Fusion Layer:** Contains a set of very abstract context information, such as location, time and activity. Each of them may be inferred from a group of sensor data but represents the same context information to different extents (maybe in different confidence, accuracy or granularity). For instance, the location fusion in figure 2 can be inferred from different data-sources with different characteristics, where GPS will only be available outdoors with an accuracy of 10-20 meters, and GSM gives less accuracy, roughly around 100 meters. The inference engine (also called context refiner) can also be backed with self-refinement [Anind et al. 2000] or even self-learning capability with the support of machine learning technologies.

Both the Conversion layer and Fusion layer can be further used repetitively to deduce higher-level context information. Like in figure 3, a context-aware application is interested in activity and people, wherein activity is a context information derived from a fusion based on time, people and location, wherein the location context information is derived from a fusion as well. We can impose a service-oriented view on this context structure by treating each supplier of the context information as a single context service, so every context service will only need to know other context services in which it is interested, without bothering to grasp the entire structure of services. Undoubtedly, some intermediary conversion services have to be available when one service cannot understand other required services. Considering when discovery is enabled among these context services, the context structure could be constructed on demand dynamically triggered by the user's request.

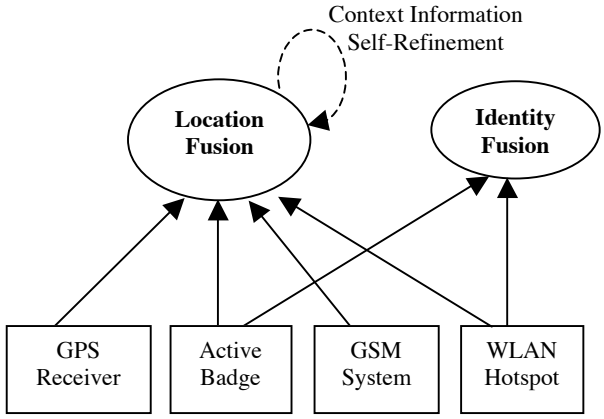


Figure 2. Location Context Fusion

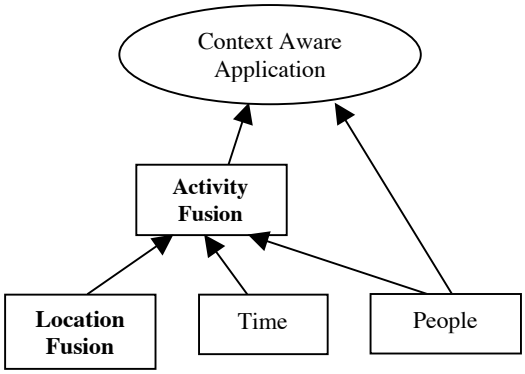


Figure 3. Context Service Tree

3.3 Computational Context

We believe computing status is also an important type of context (we call it *computational context*). It includes the running program and network status etc. Computational context helps to improve the deduction of the user's status and activities with more accuracy. For example, there may be a presentation taking place because user's computer is running Microsoft PowerPoint in a meeting room. And as an adaptation, all the mobile phones in the room turn to mute/vibrate mode automatically. Another preferable automation can be that the system assists users to fluently switch to a better interaction mode by transferring the communication session from PDA to desktop PC after arriving in the office. The examples require the system to have the capability of detecting the running programs status, e.g., to learn and remember the browsing web page in a running web browser application. However, computational context has not been fully used mainly because of the difficulty of acquiring the information of running processes, which normally can only be acquired from application program interfaces (APIs) coupled closely with the operating system.

To solve the problem of retrieving computational context information, we suggest a database solution. The running processes use database to store and retrieve computing status. Such a solution requires the processes to have extra knowledge about the database interface and data schema for accessing computational context. However, comparing with the complexity of massive low-level system APIs, it gives a uniform structure to simplify the computational context exchange. Practically, each device can have a computational context agent monitoring running processes, and report process changes to the context database. Then on the other side, another context agent observing the context database for changes will present itself as a context service to fit into our context stack (on the fusion layer). Connectivity is a good example of computational context which is actually measured by local process. It is very significant for many applications to determine the optimal communication means.

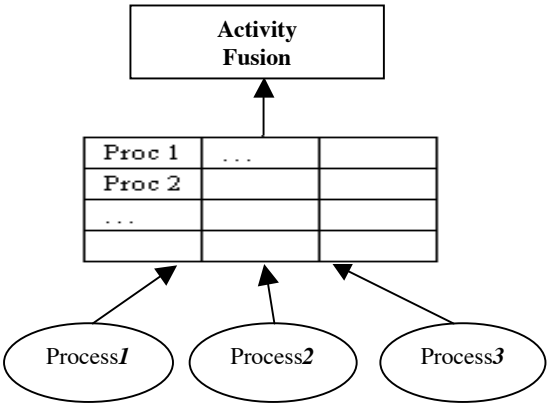


Figure 4. Computational Context Based on Database

3.4 Context Representation (Meta-data)

It is also important to show the semantics of the context acquired by using context meta-data, which is useful to identify facts such as how and from where the context is acquired. For better interoperability, XML [W3C 2000] was chosen as the encoding format due to the fact that XML has become a de facto data standard. So far, we represent context information in an atomic fashion like the example in Table 1, which can be read as an acquisition of a set of context, consisting of the temperature from wireless sensor board MICA2 and the location from an RF tag. Such context information may come from the output of *context refiners* producing refined context information through the processing of acquired context (e.g., MICA Mote does not report temperature in any standard unit directly) taking place locally or remotely. However, we decided to make our context representation neutral to the means they are acquired. We did define a *source* element which currently has only one attribute *uri* exposing the sensor entity (or device) to give the clue how this context was acquired. And we left room in the content of source element for further extension (so far it is simply the entity ID). To identify our context information and be able to co-exist with other workers, we also defined an XML namespace. We have successfully applied this context schema with SIP based Presence framework in our context information distribution infrastructure.

Table 1. XML-encoded Context Example

```
<acas:node xmlns:acas="urn:acas:">
  <acas:contextelement>
    <acas:value datatype="integer"
      unit="temperature/kelvin">300</acas:value>
    <acas:time>Fri Sep 26 16:42:25 CEST 2003</acas:time>
    <acas:source
      uri="http://MICA2_001.fuse.k2lab.dsv.su.se">
      MICA2_001 </acas:source>
  </acas:contextelement>
  <acas:contextelement>
    <acas:value datatype="string" unit="location/floor">
      Forum Floor 7 </acas:value>
    <acas:time>2003-9-17 12:00 am</acas:time>
    <acas:source
      uri="http://RFID001.fuse.k2lab.dsv.su.se">
      RFID001</acas:source>
  </acas:contextelement>
</acas:node>
```

4 Context Delivery

Most of the accessible context-aware systems are only available in research labs and the academic world. They are constrained within a specific space such as a room and a building, or more precisely by the coverage of specific networks which prevents context information to be shared among different context-aware applications.

Today, with the emerging network technologies of GPRS and 3G, context-aware systems can be deployed in a much wider area. However, considering their high cost and limited bandwidth compared to a conventional local area network (LAN), as well as the popularity of Wireless LAN and the other short-range network technologies such as Bluetooth, the optimal hybrid use of such heterogeneous networks for context delivery becomes an issue of significant value.

The optimal use of networks requires a smooth-switch capability, e.g., context information is delivered through an available Wireless LAN, and then switched to GPRS when moving out of the coverage of the Wireless LAN hotspot. However, the frequent network switch (handover) cannot be smoothly handled by the conventional network technologies like TCP/IP, Help exists from some technologies targeting this purpose, such as MobileIP [IETF 2002] which pushes the responsibility to the network infrastructure. Here, we propose another solution based on the Session Initiation Protocol (SIP) [Rosenberg et al. 2002], an application-layer control (signaling) protocol for creating, modifying, and terminating sessions. This application layer protocol creates an overlay network for communication without modification of existing network infrastructures.

4.1 SIP Protocols

The Session Initiation Protocol (SIP) is a general-purpose communication protocol supporting interactive session establishment across the Internet. It defines a complete process mechanism for establishing a distant communication session, which is independent of the underlying transport protocol and without dependency on the type of session to be established. It is a text-encoded protocol using Uniform Resource Identifiers (URI) [Berners-Lee et al. 1998] as the addressing mechanism, which resembles the normal e-mail address like “sip:user@domain”. SIP works on a client-server transaction model akin to HTTP. A SIP client generates a SIP request to the network, and waits for the response from other SIP servers, in order to establish a communication session. Every SIP message includes enough routing and session status information, so that each single

message can be delivered to its destination accordingly. In addition, SIP supports personal mobility by discovering users and locating devices, as well as the negotiation among session participants with different capabilities to determine an agreed communication session.

SIP was originally designed for IP telephony. However, due to its neutral session establishment characteristics, many other features were added with a series of complementary specifications to improve the functionality for other usages, such as supporting asynchronous information transfer.

The SIP-Specific Event Notification specification [Roach et al. 2002] addresses the issue of asynchronous notification of events through a SUBSCRIBE/NOTIFY mechanism. An application (called a Subscriber in the specification) subscribes to certain events, and will then receive notifications when the corresponding changes occur. This mechanism is especially suitable for distributed context information delivery. And the further issues pertinent to subscription duration, event packages, re-subscribe and un-subscribe etc have already been addressed in detail in the specification.

Meanwhile, SIP for Instant Messaging and Presence Leveraging Extensions (SIMPLE [IETF 2003] - another SIP-related IETF working group) focuses on introducing the Instant Messaging and Presence (IMP) service extensions to SIP network. In SIMPLE Presence Event draft [Rosenberg, J. 2003], two new notions are introduced: the *Presence Agent* (PA), a representative of a SIP Presence entity, and the *Watcher*, a consumer of a PA for status information. A new method *Publish* is proposed in the draft [Campbell et al. 2003] for sending information to any SIP Presence entity without establishing a session beforehand. Then there is also a mechanism of *Presence Aggregation* proposed in a later draft [Roach et al. 2003] to enable the aggregation of a group of presence information into a single presence entity for subscription. This mechanism can easily be adapted to present a context entity (such as a user) with a collection of different context information related.

Due to the fact that SIP and its sibling protocols have many advanced features satisfying the criteria of delivering context information in a distributed heterogeneous network environment, we chose to build our context delivery infrastructure on SIP, it is also because we believe SIP will be commonly accepted as a standard communication protocol in the near future with the 3G technology coming into force.

4.2 Context Information Distribution Infrastructure

Despite the advantages of SIP for context information transfer, the adoption of SIP is not easy due to the complexity of its transaction centric process mechanism for communication, which does not match with the principle of traditional network application programming, thus demanding developers a long study curve to get familiar with the details of the SIP protocols. We decided to bridge this gap by taking a service and component-oriented design principle to define a set of software components taking charge of SIP-based communication such as SUBSCRIBE/NOTIFY mechanisms, hence hiding the complexity of SIP network communication from context service development. With these components, context services can be easily constructed and connected to build a context network infrastructure in a *plug-and-play* style.

As SIP Presence framework is very appropriate to be extended to delivery context information, and two basic SIP entities in the framework are also explored: the Presence Agent and the Watcher, which can be implemented as reusable building blocks to construct context-aware systems. Our design components are shown in figure 5.

- **SIP Stack Wrap:** Takes the responsibility of setting up communication parameters such as network protocol (TCP or UDP) and port number, which any SIP entity has to have. The network changes such as switching IP addresses when roaming between networks will be handled by this layer without the awareness of upper layers.

- **Presence Agent (PA):** Takes charge of handling the subscription of presence and other context information, and also offers a simple interface for context services (such as fusion) to report and update detected context changes. Notifications are organized (coded in XML format) and sent to the registered Watchers automatically. The PA also handles the issue of subscriptions expiring in an autonomic way.

- **Watcher:** The Watcher is a context consumer who first registers to the PA with an indication of interesting context, and then waits for the notifications when the corresponding context updates are noticed by PA. Besides the basic subscribe, Watcher component also gives developers with interfaces for un-subscribe and re-subscribe etc.

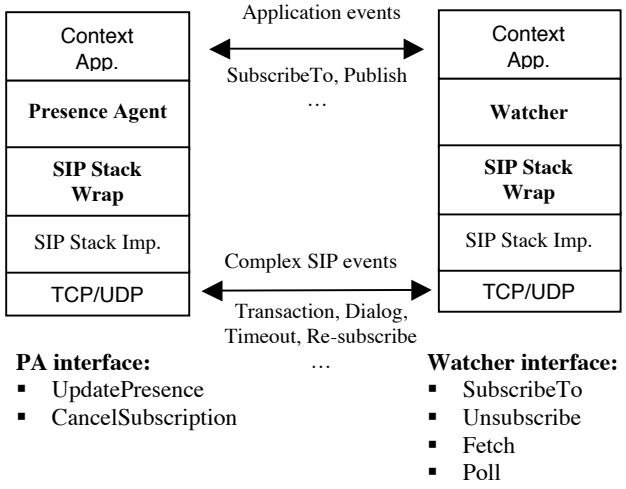


Figure 5. Presence Agent and Watcher

Our SIP components are built on top of NIST-SIP 1.2 [NIST 2003], an open source SIP stack implementation in Java, compliant to the JAIN-SIP-1.1 API standard [JCP 2002]. With the encapsulation of software components, developers can easily build context-aware systems upon SIP network to enable wide-area delivery of context information. In the following section, we will elaborate on an application scenario with which we built our context-aware system prototype based on these software components.

5 Prototype

We defined a general application scenario where a project group needs to keep live contact. We assume that project members are always online, so that they can be aware of each other's status, as

well as the progress of the project including documents and meeting schedules. To achieve this, we constructed an application infrastructure using our SIP components to support mobile ICQ-like applications. Additionally, such infrastructure provides rich context information and consequent implications rather than merely user presence information. This characteristic offers abundant possibilities for further context-aware applications development.

In figure 6, each group member subscribes to a Group Facilitator (GF) for receiving the status information of other members and the project as well. The GF also subscribes to every member's Presence Agent for acquiring information from every group member, and then aggregates the individual status into a group view shared by the whole group. The Presence Agent can be located in a stationary office desktop or in a mobile device such as Sony Ericsson P800 mobile phone.

We are testing to equip our mobile users with P800 mobile phone or iPAQ, Mica Mote and GPS receiver to acquire and report different context information including location, speed, noise-level, temperature and the people in the vicinity, as well as the available services discovered, which all together give rich implication for user and system to determine the optimal communication and interaction means. We have also connected this context infrastructure to our internal SIP telephony network with the first context-aware service – a context-aware call controller, which determines whether the incoming calls will get through, be blocked or redirected to be voice mail, according to user's current state (inferred from context) and the preference.

The infrastructure is built on top of our reusable SIP components. The GF is actually implemented as a combination of a PA and several Watchers, since GF on the one hand receives context information from the PAs (each represents a group member), while on the other hand presents an aggregated group view to subscribers (group members). The client application is based on a Watcher component with a graphical user interface (GUI) showing the status of the project resources and group members. This infrastructure will be connected to our interactive Lab – iLounge, (an interactive room with various sensors) to evaluate the potential influence on mobile users with access to rich context information from the environment. The SIP telephony network is constructed on an open source SIP server implementation – Vocal v1.5.0 [Vovida 2003], acting as a SIP proxy for delivering various SIP messages, such as Voice over IP (VoIP) call request. Upon which we developed a call-control service (also based on Watcher component) that makes call routing decision based on user's status inferred from the presence and context information retrieved through GF, and dynamically generates the corresponding Call Processing Language (CPL) [Lennox 2003] script to control the Vocal server. We have tested caller applications with Microsoft Windows Messenger (v5.0) and some other free SIP softphones successfully, the multimedia calls, including instant message text, audio and video, got across fluently. Since our context distribution is also based on SIP, we can tie our infrastructures seamlessly with any other SIP-based networks. This brings great advantages on system scalability and extensibility.

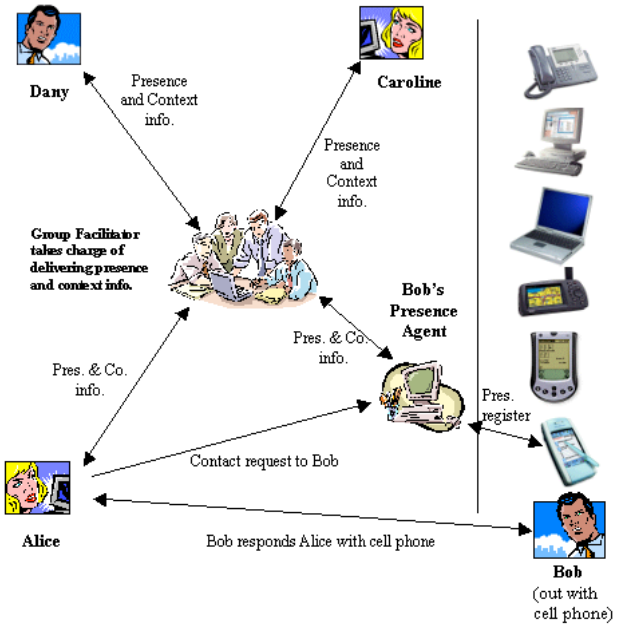


Figure 2. Group Facilitator

6 Related Work

There have been many studies on context acquisition and distribution. Here, we list some most relevant work. Schilit's infrastructure for ubiquitous computing in 1995 is probably the first context framework, which concentrates mostly on location with an active badge infrastructure [William 1995]. Then the Context Toolkit [Anind 2000] in GeorgiaTech defines a distributed architecture supporting context fusion and delivery with three notions of widget, server and interpreter based on an object-oriented architecture. Technology for Enabling Awareness (TEA) project [A. Schmidt 1999] also models the context data in a layered structure from time-stamped raw sensor data layer, Cue layer for data interpretation, to the "Context" layer. The Location Stack [Jeffrey et al. 2002] gives seven layers structure similar to OSI network model for acquisition of mainly location context.

On the other hand, the earlier work of Roychowdhury and Moyer [Arjun et al. 2001] proposed the use of presence information for remote awareness and instant messaging over SIP for remote control. Shim et al presented Spontaneous Enterprise Communications (SEC) [Hyong et al. 2001], a system designed for ad-hoc conferencing within an organization. SEC also relies on SIP for the infrastructure, using presence and availability management, conference control and text messaging.

7 Conclusions and Future Work

In this paper, we have proposed a layered context stack to help to understand the collection, interpretation, aggregation and analysis (fusion) of sensor data to deduce useful high level context information, and to facilitate the communication between researchers working on context. We suggested a service-oriented view on context information processing by presenting each process as a service, thus making these processes reusable and the context information (in different forms/levels) sharable. In addition, with this service-oriented view, the technologies applicable for services, such as service description and service discovery, can be applied to context processing as well, and the context processing may also benefit from other services which are already available in the network.

We have also explored the use of SIP and its sibling protocols for context distribution over networks, due to its advantages on flexible establishment and modification of general communication sessions. We think this advanced feature will simplify the communication between applications (and services), especially in a frequent changing environment, such as delivering the context of a mobile user.

However, there are many issues not addressed in this paper. For example, how to support dynamically adding new sensors into the infrastructure in a flexible style like plug-and-play? How to model and describe context information in a complex ubiquitous computing environment, then how to add context discovery support upon the model? ... So far, we have chosen a very simple prototype scenario, and our current model is fairly preliminary as well. It only consists of two entity types – the user and the group. The relationships between them are also rather static – the GF has the priori knowledge about the address of each group member, and so does every member about the address of GF.

We will look into such issues in detail in our future work with more complex context-aware applications, complying with the service and component-oriented architecture. One practical way is to compose more context-aware software components and small-granularity services by wrapping the existing applications we have in everyday use with context-awareness to promote the development of adaptive services. Furthermore we will also refer to the development in other research areas such as ontology and semantic web for relevant support.

Acknowledgements. This research is led and organized by Wireless@KTH, and partially funded by the Swedish Foundation for Strategic Research within the "Internet & Mobility" program.

References

W3C, Web Services Description Language (WSDL 1.2), http://www.w3.org/TR/wsdl12/

IAN FOSTER, 1998, *The Grid: Blueprint for a New Computing Infrastructure*

HENRY LIEBERMAN,TED SELKER, 2002. Out of Context: Computer Systems That Adapt To, and Learn From, Context, MIT Media Laboratory http://lieber.www.media.mit.edu/people/lieber/Teaching/Context/Out-of-Context-Paper/Out-of-Context.html as of August 7, 2002

PASCOE, J., 1997. The Stick-e Note Architecture: Extending the Interface Beyond the User, International Conference on Intelligent User Interfaces, Orlando, Florida, USA. ACM.

WANT, R., HOPPER, A., FALCAO, V., GIBBONS, J. 1992. The Active Badge Location System, *ACM Transactions on Information Systems 10*(1) pp. 91-102.

SALBER, D., DEY A.K., ABOWD, G.D. 1999. The Context Toolkit: Aiding the Development of Context-Enabled Applications, CHI'99.

LONG, S., ET AL. 1996. Rapid Prototyping of Mobile Context-aware Applications: The Cyberguide Case Study. *2nd ACM International Conference on Mobile Computing and Networking (MobiCom'96)* November 10-12, 1996.

HEALEY, J.; PICARD, R.W. 1998. Startlecam: A Cybernetic Wearable Camera. *2nd. International Symposium on Wearable Computers*, Pittsburgh, Pennsylvania, 19-20 October, 1998, pp.42-49.

Dallas Semiconductor. iButton Home Page. http://www.ibutton.com/.

Crossbow, http://www.xbow.com/Products/Wireless_Sensor_Networks.htm as of June 2003

G. CHEN AND D. KOTZ. 2000. A survey of context-aware mobile computing research. Technical Report TR2000-381, Dartmouth College, Computer Science, Hanover, NH, Nov 2000.

JEFFREY HIGHTOWER, BARRY BRUMITT, AND GAETANO BORRIELLO. 2002. The Location Stack: A Layered Model for Location in Ubiquitous Computing," in *Proceedings of the 4th IEEE Workshop on Mobile Computing Systems & Applications (WMCSA 2002)*, (Callicoon, NY), pp. 22-28, June 2002.

ANIND K. DEY, JEN MANKOFF AND GREGORY D. ABOWD. 2000. Distributed Mediation of Imperfectly Sensed Context in Aware Environments GVU Technical Report GIT-GVU-00-14. September 2000.

W3C. 2000. Extensible Markup Language (XML) 1.0 (Second Edition), http://www.w3.org/TR/2000/REC-xml-20001006, 6 October 2000.

KAREN HENRICKSEN, JADWIGA INDULSKA, ANDRY RAKOTONIRAINY. 2002. Modeling Context Information in Pervasive Computing Systems. 167-180, in *proceedings of Pervasive 2002*: Zurich, Switzerland.

MARTIN JONSSON. 2003. *Supporting Context Awareness in Ubiquitous Service Environments*, Licentiate Thesis, Royal Institute of Technology/Stockholm University.

IETF, 2002. IP Mobility Support for IPv4, RFC 3344, August 2002, http://www.ietf.org/rfc/rfc3344.txt

ROSENBERG, J., SCHULZRINNE, CAMARILLO, JOHNSTON, PETERSON, SPARKS, HANDLEY AND SCHOOLER. 2002. SIP: Session Initiation Protocol, RFC 3261, June 2002.

BERNERS-LEE, T., FIELDING, R. AND L. MASINTER, 1998. Uniform Resource Identifiers (URI): Generic Syntax, RFC 2396, August 1998.

ROACH, A., 2002. Session Initiation Protocol (SIP)-Specific Event Notification, RFC 3265, June 2002.

IETF, 2003. SIP for Instant Messaging and Presence Leveraging Extensions (SIMPLE) http://www.ietf.cnri.reston.va.us/html.charters/simple-charter.html as of June 2003.

ROSENBERG, J. 2003. A Presence Event Package for the Session Initiation Protocol (SIP)", IETF draft-ietf-simple-presence-10 (work in progress), Jan. 2003.

CAMPBELL, B., OLSON, S., PETERSON, J., ROSENBERG, J. AND B. STUCKER. 2003. SIP Presence Publication Mechanism Requirements, IETF draft-ietf-simple-publish-reqs-00 (work in progress), February 2003.

ROACH, A., ROSENBERG, J. ET AL, 2003. A Session Initiation Protocol (SIP) Event Notification Extension for Resource Lists, IETF draft-ietf-simple-event-list-04, (work in progress), June 2003.

NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY (NIST), 2003. http://snad.ncsl.nist.gov/proj/iptel/ as of June 2003.

JAVA COMMUNITY PROCESS (JCP), JAIN SIP API Specification, Aug. 2002.

VOVIDA. 2003 http://www.vovida.org/vocal as of Aug. 2003

WILLIAM NOAH SCHILIT.1995. *A system architecturefor context-aware mobile computing*. PhD thesis,Columbia University, May 1995.

ANIND K. DEY. 2000. *Providing Architectural Support for Building Context-Aware Applications*, PhD thesis, College of Computing, Georgia Institute of Technology, December 2000.

A. SCHMIDT, K.A. AIDOO, A. TAKALUOMA, U. TUOMELA, K. VA N LAERHOVEN, AND W. VAN DE VELDE. 1999. Advanced Interaction in Context. In H. Gellersen (Ed.) *Handheld and Ubiquitous Computing*, *Lecture Notes in Computer Science* No. 1707, ISBN 3-540-66550-1, Springer-Verlag Heidelberg: 1999, p. 89-101.

ARJUN ROYCHOWDHURY AND STAN MOYER. 2001. Instant messaging and presence for sip enabled networked appliances. http://www.iptel.org/2001/pg/final program/22.pdf, April 2001.

HYONG SOP SHIM, CHIT CHUNG, MICHAEL LONG, GARDNER PATTON, AND SIDDHARTA DALAL. 2001. An example of using presence and availability in an enterprise for spontaneous,multiparty, multimedia communications. http://www.iptel.org/2001/pg/final program/13.pdf, April 2001.

LENNOX, WU, AND SCHULZRINNE. 2003. CPL: A Language for User Control of Internet Telephony Services", IETF draft-ietf-iptel-cpl-08.txt (work in progress), Aug. 2003

Dynamic Distributed Multimedia: Seamless Sharing and Reconfiguration of Multimedia Flow Graphs

Marco Lohse Michael Repplinger Philipp Slusallek

Computer Graphics Lab, Saarland University, Germany

Abstract

Mobile devices with multimedia and networking capabilities are quickly becoming ubiquitous through the availability of small notebooks, PDA, and in particular, mobile phones. However, most multimedia systems are still centralized, concentrating all multimedia computing on a single device. At best they deploy a strict client/server architecture mostly for streaming multimedia from some server while processing it locally.

Ubiquitous computing environments on the other hand require systems that allow multimedia content and processing to be flexibly distributed across different devices that are currently available in a network while supporting dynamic migration between devices as users move or requirements change.

This paper concentrates on the two most important operations for dynamic distributed multimedia: combining two flow graphs so that common processing can be shared between them as much as possible, and – based on this service – reconfiguring a running multimedia flow graph across a network. We use a number of different application scenarios to motivate the basic middleware requirements including device control and distributed synchronization. Finally, we present two key applications and report on results of an implementation of this approach.

1 Introduction

Today, there is a strong demand for mobile and ubiquitous computing in the area of multimedia entertainment. Mobile devices like Personal Digital Assistants (PDAs), portable web pads, or even cellular phones already provide decent multimedia and from reasonable to very good networking capabilities even in mobile scenarios.

While we are increasingly surrounded by many of these devices, multimedia processing in most systems is still concentrated in a single device. For example, we cannot seamlessly play the music stored on an MP3-player, a phone, or a PDA on our car stereo when getting into the car or switch to the living room speakers when coming home. Nor can we use the mobile devices to control the volume or any other aspect of multimedia processing.

This is even more surprising given that multimedia systems generally implement a very flexible processing approach, where many small processing elements or nodes are interconnected into a multimedia flow graph such as in Microsoft's DirectShow [Microsoft 2003] or in the Java Media Framework [Sun 2003]. Each of these nodes provides fine grained access to specific multimedia devices (e.g. TV receiver, DVD drives, or audio output) or processing capabilities (e.g. MPEG compression or demultiplexing). In such a system, distributed multimedia is conceptually simple to implement by routing the data to and from processing or device nodes located on remote machines. Several such systems have been built in the research community as will be discussed below.

While the details of such distributed multimedia systems are non-trivial, the main challenge is the dynamic (re-)configuration of active flow graphs. In this paper we describe the functionality of sharing parts of active flow graphs from within different applications running on different computers (Section 5). We then extend this service to realize the dynamic reconfiguration of flow graphs. The flow graph of running applications can be reconfigured and migrated during runtime to remote systems (Section 6). During this reconfiguration, media playback stays continuous and fully synchronized.

To motivate the requirements of such services, we discuss some application scenarios and related work in Section 2. Section 3 describes the underlying middleware while Section 4 gives an overview of the synchronization architecture, which forms the basis for synchronized playback across distributed devices. Finally, we conclude this paper and describe future work in Section 7.

2 Application Scenarios and Related Work

To motivate the requirements for the presented work we introduce two application scenarios. The first scenario is a media playback application started on some computer, e.g. watching a movie from a DVD drive. Another user could then join watching the movie but might want to listen to a different language track using his PDA to receive and play the audio via earphones.

To realize this scenario, the PDA needs access to parts of the DVD application so it can connect to the running flow graph. Furthermore, in a scenario where several users share one display but listen to different audio tracks, synchronization across all connected applications must be provided to keep lip-sync. In Section 5, we introduce the concept of a *session* as an abstraction for a flow graph of already connected and active devices and refer to this scenario as *session sharing*.

Previous work, such as [Kahmann and Wolf 2002], uses a proxy architecture together with standard streaming servers and clients for collaborative media streaming. In contrast to our approach, no support is provided for synchronized playback or sharing of flow graphs. In [Roman et al. 2002], an application framework for active environments is proposed that can map running applications between different active environments depending on the users position. Furthermore, another user can connect to these running applications and receive the same data. Our approach is similar to this one but is more flexible because instead of mapping whole applications we map only the required parts of a flow graph.

Based on this service, several other applications, that require reconfiguration can be realized, such as seamless and synchronized

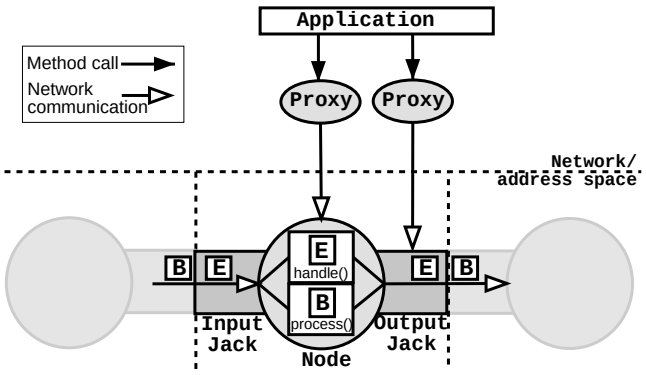


Figure 1: A node is connected to other nodes via its input and output jacks. Messages ('E' event, 'B' Buffer') are sent instream. Proxies allow distributed access to nodes.

handoff. Let us consider a playback application running on a mobile device, e.g. a PDA, playing audio data through its internal speakers or earphones. When the user moves around and enters an environment with richer I/O capabilities (e.g. a hi-fi system in a living room, a car, or an office) the output could be handed off to these more capable devices. If the user later leaves this environment the output should be handed back to the mobile device.

To realize this scenario, the application must be able to dynamically adapt an active multimedia flow graph to changing environments. In addition, the handoff should be seamless and continuous. This means that no data is lost or duplicated and media playback is fully synchronized at all times, which is especially important for the playback of music. Contrary to a common *handoff* scenario, which describes the change of an existing network connection, this scenario requires to reconfigure and migrate parts of an active flow graph. Therefore we use the term *handover* to describe it.

The main disadvantage of existing approaches is the lack of synchronized playback during handover. For example, the approach described in [Lin et al. 2002], is based on moving application sessions across different devices. They use existing media players as clients and therefore cannot handover parts of a flow graph during synchronized playback. In [Wedlund and Schulzrinne 1999] the usage of standard protocols like SIP is examined but without the aspect of synchronized playback. Another approach, similar to ours, is described in [Carlson and Schrader 2002]. They are using two simultaneous processing chains to enable a zero-loss behavior. However, they cannot provide synchronized playback during handover, either. Our approach presented in Section 6, allows the reconfiguration of an active flow graph without loss of any data. In addition, the playback remains synchronized during handover.

3 Middleware Requirements

For the following discussions we assume a multimedia middleware that is based on a distributed flow graph approach where any node can be distributed across the network while transparent access is provided by some means, e.g. using a proxy architecture (see Figure 1). Furthermore, this middleware must provide a registry service for node discovery and a distributed synchronization infrastructure (discussed in the next section).

These requirements are fulfilled by the *Network-Integrated Multimedia Middleware* (NMM) that is especially designed to access, control, and integrate distributed multimedia devices. A detailed description of the middleware can be found in [Lohse et al. 2002].

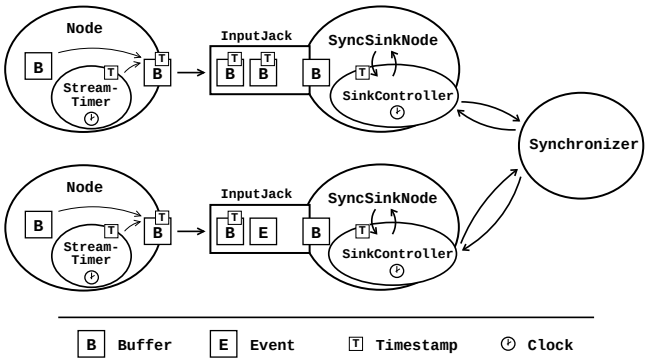


Figure 2: Time stamped buffers are handled by controllers. A synchronizer coordinates playback for different nodes.

NMM represents hardware devices or software components by *nodes* that offer input and output ports, referred to as *jacks*. Jacks specify the multimedia *formats* supported for multimedia data and are used to connect nodes into a flow graph. Since a node can have several input and output jacks they are labeled with a *jack tag*. Jacks receive or send messages, as shown in Figure 1. Furthermore output jacks can be dynamically duplicated if they are requested multiple times. They then form a *jack group*, that forwards outgoing messages to all its output jacks. The concept of a jack group is an essential requirement for the middleware services described in Section 5 and 6. The messaging system provides two different types of messages. *Buffers* are used to transport multimedia data and *events* are used for arbitrary control information.

A registry services must be provided for discovery, reservation, and instantiation of possibly remote devices. We assume that this registry service stores a complete *node description* of all available nodes, which includes the specific type of a node (e.g. "TVCardNode") and the supported formats (e.g. "video/raw"). The registry processes application queries specified as *graph description*. A graph description includes a set of node description, connected by *edges*.

After successfully processing the request by matching available nodes to the query, the queried nodes are instantiated and returned by the registry. The registry service must also be able to setup and create distributed flow graphs by delegating instantiations to remote registries.

4 Synchronization Architecture

To realize synchronized playback of nodes distributed across the network, NMM provides a generic distributed synchronization architecture.

Since the synchronized play out of different streams (e.g. lip-synchronized audio and video) is important for every single buffer (e.g. some audio samples or a single video frame), it is especially important to minimize communication needed for synchronization in distributed environments. Therefore, NMM strictly distinguishes between *intra-stream* and *inter-stream* synchronization [Gordon Blair and Jean-Bernard Stefani 1998]. Intra-stream synchronization refers to the temporal relations between several presentation units of the same media stream (e.g. subsequent frames of a video stream), whereas inter-stream synchronization maintains the temporal relations of different streams (e.g. for lip-synchronizing of audio and video).

Figure 2 provides an overview of synchronization during playback in the NMM architecture. Synchronization at source nodes (e.g. for capturing of live data) is performed similarly. The follow-

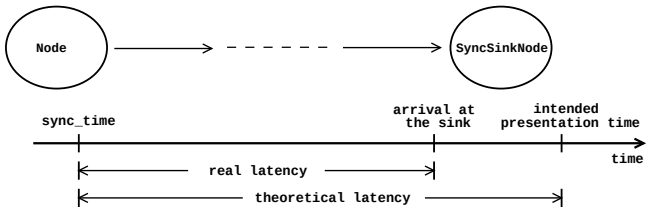


Figure 3: The real latency is computed for each stream; the desired or theoretical latency of all streams has to be identical for synchronized playback.

ing sections will then show how this architecture is used to realize different synchronization protocols, e.g. for shared session or seamless reconfiguration of flow graphs.

The basis for performing synchronization is a common source for timing information. We are using a static `Clock` object within each address space. This clock represents the system clock that is globally synchronized by the Network Time Protocol (NTP) [NTP: The Network Time Protocol 2003] and can therefore be assumed to represent the same time basis throughout the network. With our current setup, we found the time offset between different systems to be in the range of 1 to 5 ms, which is sufficient for our purposes.

Each buffer also holds a `Timestamp` that represents the time when this buffer should be presented referring to some arbitrary time base. The time base and the value of the timestamp is specific for the flow graph instantiated by an application, e.g. a timestamp might be taken from the multimedia data stream (e.g. the MPEG timestamps) or might be generated by a node. `StreamTimer` objects help setting timestamps for these different cases.

All sink nodes that allow synchronized processing of buffers are subclasses of `SyncSinkNode`. Such a node delegates intra-stream synchronization to a `SinkController` that decides if and when to present the buffer by inspecting the timestamp. As controller objects resides within the corresponding node (i.e. within the same address space), no network traffic is involved in this step.

If multiple data streams are to be presented in sync, the controller objects involved are connected to a `Synchronizer` that realizes inter-stream synchronization by implementing a specific synchronization protocol. Notice, that the different controller objects might be running on different hosts or in different address spaces as the synchronizer. In order to minimize network traffic, the controller objects locally implement the intra-stream decisions of the `Synchronizer` that are communicated only when needed.

How can a controller decide when exactly to present a buffer in comparison to another stream of buffers? The main idea is that the different controllers running within different sink nodes should present their buffers as if they had the same *latency* (called *desired* or *theoretical latency*, see Figure 3) with respect to processing in the flow graph.

During runtime, a controller computes the latency for each incoming buffer – the *real latency*. If this latency exceeds a predefined value depending on the desired latency, the buffer is too old and declared as invalid. If the real latency is smaller than the desired latency, the presentation of the buffer will be delayed.

The goal of the inter-stream synchronization is that the latencies for the different streams are equal. To achieve this, every controller sends its computed latency for the first buffers to arrive to the synchronizer. These values are taken as a first estimate of the overall latency: the synchronizer computes a new desired latency as the maximum and sets this value for all connected controllers. During runtime, all controllers also compute the average value of the latencies of incoming buffers, where the number of buffers to consider may be varied by the synchronizer or the application, a typical value would be for every 25 video buffers or 50 audio buffers. Only this

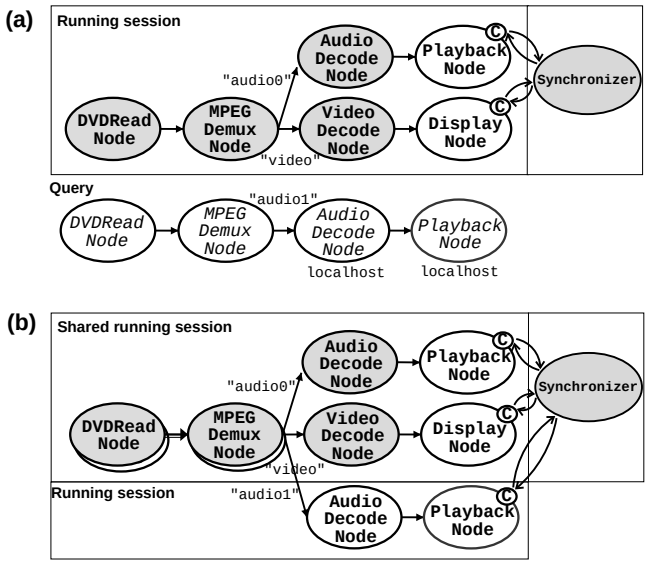


Figure 4: Session sharing. The query for a different audio track of an already running session (a) is mapped by the session sharing algorithm to use a shared sub-graph (b). The controllers (marked as 'C') of all sink nodes are connected to the synchronizer.

value is sent to the synchronizer. The synchronizer then again computes a theoretical latency for all connected controllers and updates its controllers. Notice, that with this architecture, network traffic is reduced to a few messages per second with only a few bytes per message.

We use a more advanced protocol that further reduces network traffic: since interruptions or dropped buffers of audio streams are much more disturbing than for video streams, the controller of the first audio stream is chosen as master; all other controllers act as slaves that have to adjust their playback speed to the master. To realize this, the synchronizer propagates the latency of the master as theoretical latency to all slaves; only if the controller of a slave detects that its real latency has increased by a multiple of the allowed values (e.g. due to long-term changed networking conditions), it will report this to the synchronizer and the synchronizer will use this value as new desired latency – and therefore also interrupt the master stream once.

Within this architecture, we have implemented a synchronizer that can handle several audio and video sinks and allows to dynamically add and remove streams.

Notice, that this synchronization protocol also compensates for the drift between the internal clock of the sound board (that controls audio playback) and the system clock (that controls for example video playback). We have found this drift to be in the range of one second for 30 minutes of video. When using more than one audio sink, the controllers of the slave sink nodes have to adjust their playback speed, e.g. by dropping or doubling samples.

5 Automatic Session Sharing

As described in Section 2, there are a number of scenarios that need the possibility to share parts of a flow graph. Therefore, if the specific nodes for a flow graph are requested, a sharing policy can be set: nodes can be marked as *shared* for explicitly shared access, *exclusive* for explicitly exclusive access, and *exclusive then shared* to share an exclusively requested node. A combination of policies such as *exclusive or shared* is also possible. In this case a shared reservation is chosen if an exclusive request failed.

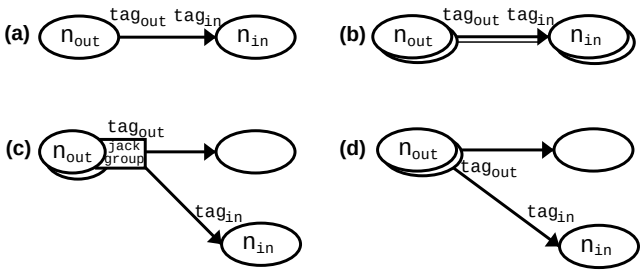


Figure 5: An edge of a query in (a), a complete overlap (b), a partial overlap using a duplicated output within a jack group in (c), a partial overlap with a different output connected in (d).

A flow graph of reserved and connected nodes is stored as a *session* within the registry service. Shared nodes of a session can then be reused within newly created flow graphs. To realize the synchronized playback for shared sessions, a session also contains a synchronizer object that can handle several audio and video tracks and that allows to dynamically add new tracks as described above.

As an example, a running session with shared nodes is shown in Figure 4(a). In this example, the application chose to share all nodes (shaded dark) required for watching a DVD, except the sink nodes for rendering audio and video. Another application that wants to access and playback a different audio stream of the DVD, being used in the running session, will use a query such as the one in Figure 4(a). Here, the mode "exclusive or shared" is set for all node descriptions, except the nodes for decoding and playback. These nodes should run on the local host and use the "audio1" output of the demultiplexer (instead of "audio0").

If a query (described as a graph description, see Section 3) is processed by the registry service, a session sharing algorithm searches the running sessions for overlapping sub-graphs that can be reused. Intuitively, an overlapping between two flow graphs only makes sense, if they share some common source of data, like the same DVD drive. Otherwise, the two different flow graphs would try to share internal nodes for different data, which does not make sense.

The algorithm tries to "grow" overlapping sub-graphs starting from such a source node: for each outgoing edge e^q from node n_{out} to n_{in} of the query graph (see Figure 5(a)) and an edge e^r of the already running graph, different types of overlaps can exist:

- A *complete overlap* exists if all constraints of e^q are fulfilled by e^r. These constraints include the node types, the sharing policies and the hosts the nodes n_{out} and n_{in} should run on. Additionally, the input and output jack tags have to match (see Section 3). In such a case, both nodes of e^r and the connecting edge can be used by e^q without the creation of any additional node.

- A *partial overlap* of e^q and e^r exists in two cases: first, if only the *outgoing* parts of the above mentioned constraints are fulfilled. An example would be an edge where n_{in} does not allow sharing. In such cases, only the node n_{out} can be reused within the query and an additional node for n_{in} has to be created. Then, the already connected output of n_{out} is copied and inserted into the jack group (see Section 3). The data flow will be forwarded to both outputs (see Figure 5(c)). Although nodes always use jack groups, we only depict them if two or more jacks are used.

  Secondly, two edges partly overlap, if the *outgoing* parts match, but e^q corresponds to a different previously unconnected output tag_{out} of node n_{out}. In this case, again, n_{in} has to be created and will get connected to the other output (see Figure 5(d)).

For a complete overlap, the algorithm recursively continues the search starting from the node n_{in}. For a partial overlap, the sub-graph connected by e^r is removed (since it can not be reached anymore) and the search is continued. Within all recursions, the search ends if no more nodes can be overlapped. Although in general, an exhaustive search is performed, the strict criteria for the different types of overlappings and the pruning of the search tree greatly reduce the number of iterations needed for typical flow graphs. More details on the algorithm can be found in [Lohse et al. 2003].

If different overlaps for a query were found, a value function is used to decide which overlap should be taken. We currently use a very simple function that prefers the overlap with the most shared nodes and edges, but more sophisticated approaches can be used as well.

5.1 Results

Figure 4(b) shows the result of the graph sharing algorithm for our example: the `DVDReadNode` and the `MPEGDemuxNode` and their connected edge are now shared for the second session (*complete overlap*), whereas an additional edge was created to connect the second audio output of the `MPEGDemuxNode` to the newly instantiated nodes `AudioDecodeNode` and `PlaybackNode` that are running on the local host (*partial overlap* to previously unconnected edge). With this setup, a different audio stream will be rendered on the device that runs the second session. On a commodity Linux-PC, the runtime of the sharing algorithm for this setup is about 59 milliseconds.

The synchronizer described in Section 4 is used to provide synchronized playback for this distributed flow graph: the audio sink of the running session is chosen as master; the video sink and the second audio sink act as slaves. Although the first few buffers for the second audio sink usually arrive too late and will be discarded, the new sub-graph catches up after a few dropped buffers.

6 Seamless and Synchronized Reconfiguration

In order to provide the highest quality output to the user or to distribute multimedia data processing, parts of an active flow graph can be seamlessly migrated to other devices during runtime. This service builds upon the session approach as described below.

Figure 6 shows one possible application scenario for such a reconfiguration: the user wants to playback media files stored on a mobile device. If no other system is nearby, the presentation of the media data is performed on the mobile system. In our case, this would be the playback of decoded MP3 files through internal speakers. The flow graph for this example consists of three nodes, all running locally in the beginning, but all accessed via proxies and interfaces as described in Section 3. The `ReadfileNode` reads the encoded file from the internal memory or a memory card, the `AudioDecodeNode` decodes the data (e.g. MPEG-audio or Ogg/Vorbis), and the `PlaybackNode` performs the audio output.

If a stationary system with richer I/O capabilities – e.g. high-quality stereo audio output – is nearby, the playback of audio data should be handed over from the mobile device to the stationary system. If the stationary system also provides the possibility to perform the decoding of the audio data, the corresponding node should also be migrated. As mentioned above, the selection of the nearby system is currently done manually. If the user moves on, the session might get handed over back from the stationary system to the mobile device or from one stationary system to another.

As there are different application scenarios where such a dynamic adaptation of an active flow graph is needed, the basic idea is always the same: We configure the new parts of the current flow

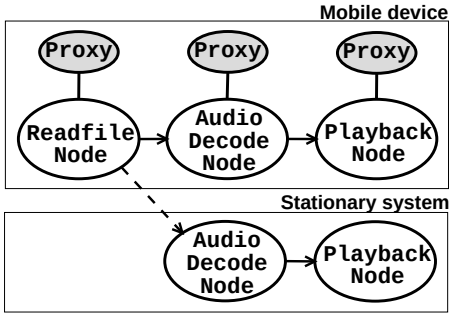
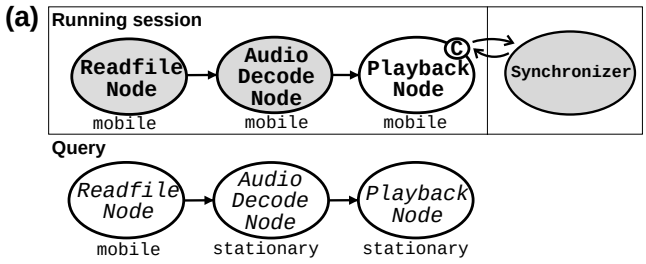


Figure 6: Application scenario showing handover from mobile to stationary system.



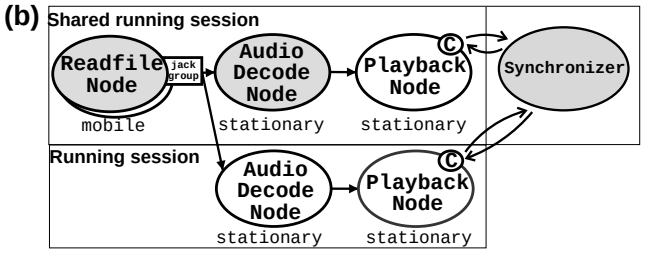


Figure 7: Automatic setup of the slave graph: the query specifies how the running session should be reconfigured (a). The slave graph as found by the session sharing service in (b): additional nodes were created on the stationary system; both controllers (marked as 'C') are connected to the synchronizer.

graph while keeping the original media processing running continuous until both are synchronous. This synchronized handover might also include the migration of nodes to remote hosts.

There are two main steps that have to be performed for a seamless handover:

- The first step is to create reconfigured second instances of the specific parts of the current flow graph (possibly on another host). The sub-graph of the current flow graph is called the *master graph*; the newly instantiated sub-graph *slave graph*.
- The second step is then to switch the data and control connections from the currently used instances to these newly created instances while keeping media processing continuous and synchronized.

Due to the fact that at some time during this procedure, the old data connection has to be torn down while the new data connection will be established, there is no guarantee that the playback will stay continuous and synchronized. Therefore, the main idea to provide this feature is to setup the slave graph as soon as possible and additionally start streaming data through the slave graph while still

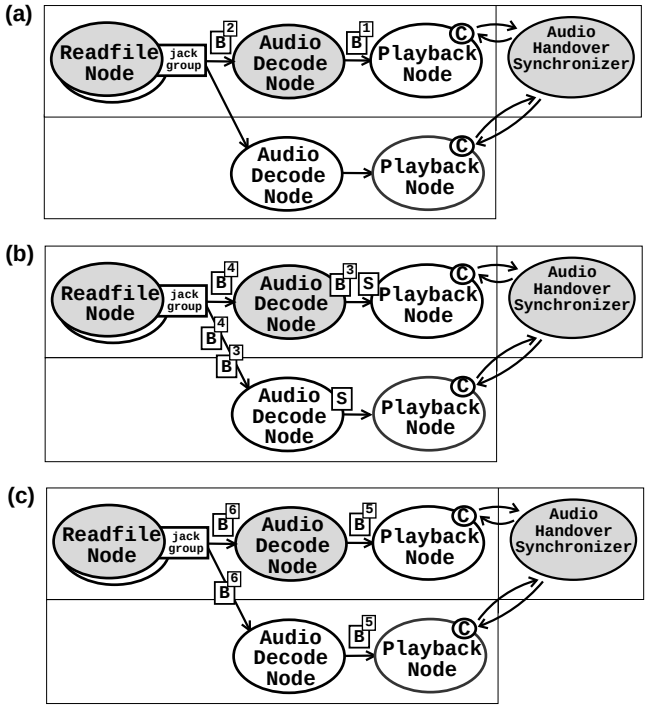


Figure 8: Synchronized continuous handover. While data buffers (marked as 'B', timestamp as number) are flowing in the master graph, remote nodes are setup (a), instream events are sent ('S' syncReset) and the slave graph is filled (b), data is flowing in both flow graphs simultaneously (c).

streaming data through the master graph. Presentation of multimedia data (e.g. playback of audio) will be done within the master graph only until the slave graph can present data synchronized with the master graph. Then, the presentation within the master graph is stopped and synchronously started within the slave graph.

The session sharing service described in Section 5 already provides all facilities needed for the first step, namely the automatic creation of the slave graph. All an application has to do, is to create a query that is first of all a copy of the currently running flow graph. Then, in this copy, all parameters can be reconfigured as wanted: in our example, the node for decoding audio and the sink node are configured to no longer run on the mobile system but on the stationary system (see Figure 7(a)). Also, additional nodes can be inserted, as desired.

The query is then forced to be mapped to the running session by additionally specifying its ID. The session sharing algorithm automatically maps the query to the running session; additional nodes are created and connected: again, for our example, the `ReadfileNode` is found to be shared (as it runs on the specified host "mobile"), and the `AudioDecodeNode` and the `PlaybackNode` are instantiated on the stationary system as specified in the query (see Figure 7(b)). The output jack of the `ReadfileNode` is duplicated within its jack group and connected to the remote AudioDecodeNode (*partial overlap* with copied output, see Section 5).

To realize the second step – namely the seamless and synchronized switching from the master to the slave graph – a special synchronizer is used that is – for our example – called `AudioHandoverSynchronizer`. This synchronizer gets connected to the controller objects of the sink nodes involved in the handover process. Furthermore, the functionality of the jack group is extended as described below.

In Figure 8(a), timestamped data buffers are only flowing in the

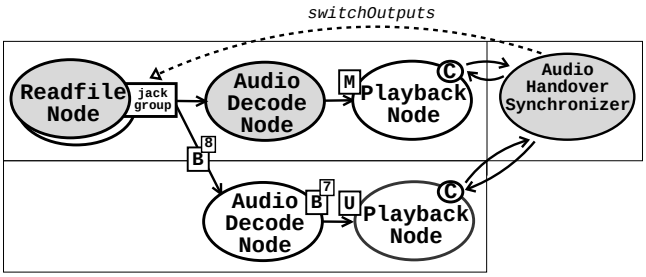


Figure 9: Complete handover using instream events for switching playback from the master to the slave graph ('M' mute, 'U' unmute).

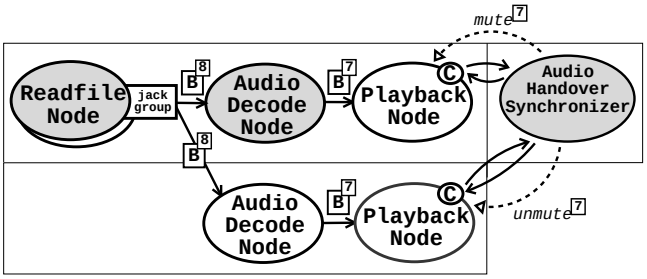


Figure 10: Complete handover using method calls for switching playback from the master to the slave graph (mute and unmute with additional timestamp).

master graph and playback is therefore only performed on the mobile system; the sink node is connected to the synchronizer. The slave graph is then set up on the stationary system as described above.

Once the connection is setup, the jack group sends an instream event called "syncReset" to both of its connected outputs *before* sending any multimedia data to its second output (see Figure 8(b)). This instream event will flow downstream and reach both sink nodes where it will force to reset the controller objects and send a new measured latency to the synchronizer object. From this moment, the same multimedia data will arrive at both sink nodes although with different latencies.

The synchronizer will then send the controller of the slave graph the latency as measured in the master graph. This is due to the fact, that the playback within the master graph should not been interrupted during handover and therefore the slave graph has to "catch up" with the master graph.

Depending on the used networking technology and protocol and due to the fact that the pipeline of the slave graph has to be filled first (including the delay of the algorithm for decoding multimedia data), the first data buffers usually arrive too late at the sink in the slave graph. Therefore, the controller of the slave graph will discard these buffers.

The next step in the handover procedure can be performed as soon as data in the slave graph arrives "in time" at its sink node. Since the slave graph will discard data that arrives too late, this will always be the case, if the time it takes for one buffer to stream through the complete slave graph is smaller than the time the same buffer will be presented (in our case: the raw audio data will be played back in the master graph). If this is *not* the case, the delay of the slave graph is *always* too large to ensure timely playback, which – in our case – would mean that the processing power provided for the slave graph is not sufficient for timely decoding and playback of the MP3 data. Of course, such a precondition should be fulfilled.

As soon as the first data buffer arrives in time, it could be played back synchronous with the sink node in the master graph (see Figure 8(c)). Depending on the behavior desired by the application there are two possibilities: The audio streams in both graphs – the master and slave graph – can be played back *simultaneously* from now on (*case 1*). This is basically the same behavior as used for session sharing in general (see Section 5).

The second possibility is to perform a *complete* handover (*case 2*): the audio playback would then stop within the master graph and start within the slave graph at the same time. To realize this strategy, the synchronizer will notify all jack groups that were created during the setup of the slave graph. Such a jack group will then insert an instream event "unmute" into the outgoing data stream to the slave graph, and an instream event "mute" into the outgoing data stream to the master graph (compare Figure 9). It will then stop sending data to its first output, namely the output that is connected to the master graph. Upon receiving the (synchronous) "unmute"-event, the sink node in the slave graph will seamlessly start synchronized playback of the audio data. The sink node in the master graph will stop playback when receiving the "mute"-event.

Another realization of this last step lets the synchronizer decide when directly to "unmute" playback in the slave graph respectively to "mute" playback in the source graph. This is done by estimating the maximum delay it takes to send a message to the two controllers. The "mute" and "unmute" method calls then additionally hold a timestamp that marks the execution time of the commands (see Figure 10). We found both approaches to be similarly efficient.

6.1 Results

We have measured the times for the described reconfiguration with two different setups: one uses two commodity Linux-PCs connected over 100 MBit Ethernet as master and slave; the other uses a Linux-PDA (namely the iPAQ H3870) with 11 MBit WLAN as master. For two PCs, it takes 0.562 for connecting the slave graph and another 0.461 until the playback is synchronized; together 0.983 seconds (see *case 1* above). A complete handover (see *case 2* above) takes additional 0.302 seconds. Using the PDA as master, these values raise to around 4 seconds (case 1) respectively 4.5 seconds (case 2) due to the lower computational power of the mobile device. Still, we think that these delays are tolerable since the media playback stays continuous and synchronized all the time.

7 Conclusions and Future Work

In this paper we presented an approach for sharing parts of an active flow graph within different applications. Using a generic synchronization architecture, this allows for joint and synchronized access of multimedia data on different devices. Based on these services is a mechanism for dynamically reconfiguring active flow graphs. During such an adaptation process, media playback stays continuous and synchronized. We demonstrated our implementation with different application scenarios and pointed out the most important results.

Future work will concentrate on the integration of Quality of Service measurements to guide the reconfiguration process. We will also study different policies for sharing devices in multi-user scenarios.

Acknowledgements This research has been supported by Motorola, Germany, and the Ministry of Saarland, Germany.

References

CARLSON, D., AND SCHRADER, A. 2002. Seamless Media Adaptation with Simultaneous Processing Chains. In *ACM International Conference on Multimedia*.

GORDON BLAIR AND JEAN-BERNARD STEFANI. 1998. *Open Distributed Processing and Multimedia*. Addison-Wesley.

KAHMANN, V., AND WOLF, L. 2002. A proxy architecture for collaborative media streaming. *Multimedia Systems 8*, 5.

LIN, J., GLAZER, G., GUY, R., AND BAGRODIA, R. 2002. Fast Asynchronous Streaming Handoff. In *Protocols and Systems for Interactive Distributed Multimedia Systems (IDMS/PROMS)*.

LOHSE, M., REPPLINGER, M., AND SLUSALLEK, P. 2002. An Open Middleware Architecture for Network-Integrated Multimedia. In *Protocols and Systems for Interactive Distributed Multimedia Systems (IDMS/PROMS)*.

LOHSE, M., REPPLINGER, M., AND SLUSALLEK, P. 2003. Session Sharing as Middleware Service for Distributed Multimedia Applications. In *Multimedia Interactive Protocols and Systems (MIPS)*.

MICROSOFT, 2003. DirectShow. http://msdn.microsoft.com/.

NTP: THE NETWORK TIME PROTOCOL, 2003. http://www.ntp.org/.

ROMAN, M., HO, H., AND CAMPBELL, R. H. 2002. Application mobility in active spaces. In *1st International Conference on Mobile and Ubiquitous Multimedia*.

SUN, 2003. Java Media Framework. http://java.sun.com/products/java-media/jmf/.

WEDLUND, E., AND SCHULZRINNE, H. 1999. Mobility Support Using SIP. In *International Workshop on Wireless Mobile Multimedia*.