Proceedings of the Workshop on Environments for Personalized Information Access

Working Conference on Advanced Visual Interfaces

AVI 2004

Gallipoli, Italy
May 25, 2004

Editors: Liliana Ardissono and Giovanni Semeraro

Environments for Personalized Information Access

Proceedings of the AVI 2004 Workshop on Environments for Personalized Information Access

Gallipoli, Italy, May 25, 2004

Liliana Ardissono and Giovanni Semeraro (Eds.)

Dipartimento di Informatica, Università di Torino

Dipartimento di Informatica, Università di Bari

Volume editors

Liliana Ardissono
Dipartimento di Informatica
Università di Torino
Italy

and

Giovanni Semeraro
Dipartimento di Informatica
Università di Bari
Italy

Sponsoring Institution

Università degli Studi di Bari

This volume has been published with a contribution of Università degli Studi di Bari.

Table of Contents

Preface

Nowadays, most large-scale applications are designed for a large variety of users. An emerging research issue is to enable these systems to satisfy heterogeneous needs. The traditional “one-size-fits-all” approach is outdated: the challenge, now, is to enhance the system's ability to adapt its own behaviour to individual users needs. Some research areas focus on automatically personalizing applications based on observations of user behavior, by employing techniques from machine learning and the broader Artificial Intelligence community. For example, the systems that observe a user's actions may be able to identify the tasks (s)he is performing and, through this understanding, they may adapt the user interfaces that will facilitate performance of tedious tasks or provide assistance with complex procedures. Moreover, the personalization of applications moves along another dimension: with the advent of ubiquitous services, the adaptation to the characteristics of the transmission channel (broadband, quality of service, etc.), to the user device (memory size, etc.) and to other factors (e.g., emergency, stress, etc.) become extremely important properties of the User Interfaces. The deep heterogeneity of users and devices makes the design and development of usable Advanced Visual Interfaces a challenging topic for the research in personalization, that can contribute to improve them by enhancing their capabilities to adapt to specific contextual conditions. This workshop is intended to provide a forum in which researchers from diverse fields such as machine learning, knowledge engineering, psychology, cognitive sciences, adaptive user interfaces, user modelling and virtual reality can examine the personalization aspects of the advanced user interfaces.

We hope that this workshop will generate interesting discussions and will give you the inspiration for new research directions.

We would like to thank all the members of the workshop Program Committee, who supported us in the selection of the papers to be presented at the workshop and who provided insightful comments to help the authors improve their contributions.

Liliana Ardissono and Giovanni Semeraro
Gallipoli, May 25, 2004

Program Committee

Fabio Abbattista	(University of Bari, Italy)
Mathias Bauer	(DFKI, Germany)
Peter Brusilovsky	(University of Pittsburgh, USA)
Sandra Carberry	(University of Delaware, USA)
Luca Chittaro	(University of Udine, Italy)
Berardina De Carolis	(University of Bari, Italy)
Anna Goy	(University of Torino, Italy)
Matthias Hemmje	(FhG-IPSI, Germany)
Anthony Jameson	(DFKI, Germany)
Mark T. Maybury	(MITRE Corporation, USA)
Alessandro Micarelli	(University of Roma Tre, Italy)
Claudia Niederée	(FhG-IPSI, Germany)
Michael J. Pazzani	(National Science Foundation, USA)
Oliviero Stock	(ITC-irst, Italy)
Carlo Tasso	(University of Udine, Italy)

Local Organization

Marco Degemmis	(University of Bari, Italy)
Pasquale Lops	(University of Bari, Italy)

Determinants of Social Presence in Videoconferencing

Yevgenia Bondareva and Don Bouwhuis

Technische Universiteit Eindhoven, Den Dolech 2, P.O.Box 513
5600 MB Eindhoven, Netherlands
y.bondareva@tue.nl; d.g.bouwhuis@tue.nl

Abstract. A critical issue concerning communication between remote persons by means of videoconferencing is the ability of the system to provide a high degree of social presence. In this paper we describe the results of a comparative evaluation of two videoconferencing systems. One was a traditional one with a camera mounted above the display. Main disadvantage of such systems is that they do not convey direct eye contact. The key idea of the alternative Direct Eye Contact design is a combination of back-projection and mirroring techniques. This system gives participants on both sides the experience of having direct eye contact. It also provides a wide visual field with a life-size participant's image and the illusion of a shared working table. The Direct Eye Contact system provided a significantly higher degree of social presence than the traditional one, as based on subjective, and especially behavioral measures, like body movements, facial expressions, eye fixation and gestures.

1 Introduction

Nowadays, it is relatively easy to organize communication between remote persons by means of videoconferences. Nevertheless, the amount of actual physical travel for business purpose continues to increase each year. In order to have face-to-face contact people prefer traveling in spite of time and travel expenses. It appears that face-to-face communication has some advantages that videoconferencing does not provide. The feeling of being together during immersive communication might be a critical issue also for families where members are living far away from each other.

1.1 Analysis of the traditional videoconferencing systems

What is a videoconferencing system (VCS) nowadays? Market overviews show that the basic element of the system (called here: Traditional VCS) is usually a TV set or a projection screen with a camera mounted above, and sometimes below or beside the display. So, users are limited to a single camera position with respect to their location. Typically, a single video stream is provided. It is incapable of providing both the full field of view of the normal human vision and the resolution of the human visual fovea at the same time. Only one audio channel is provided with a limited dynamic range and no directional information [6].

Most current commercial videoconferencing systems do not render the users life size. However, experiments have shown that presenting the users life size increases social presence for people interacting with them [9]. The term “social presence” is used for describing the sense of “being together” for the participants of videoconference located in different places.

In face-to-face meetings we recognize intentions of the participants that are not expressed by verbal language. For example, when people lack confidence, they cast our eyes down. We gaze at others to impel them to speak. [8]. Nonverbal communication is often unconscious. A large size of the screen image helps to see the body language of the fellow speaker in details. This is highly important for communication because only by combining verbal and non-verbal cues, the true intentions of the interlocutor can be correctly interpreted.

Consequently, eye contact is likely to be of considerable importance for social presence. Mutual gaze serves several communicative functions, such as regulating the flow of conversation, communicating emotions, and communicating interpersonal relationship. [7] However, since the cameras are usually mounted above the screen in conventional teleconferencing systems, direct eye contact cannot be achieved. This happens because, when a participant is fixating someone’s image on the display, he is not looking into the camera, mounted above the display. This causes a visual parallax between the location of the camera lens and the on-screen representation of the participants. As a result people see their interlocutor looking not at their eyes but somewhere down, on the chin or chest. This might be annoying and makes people feel uncomfortable. (We do not consider here cases of the large videoconferencing rooms where participants are sitting far from the screen or use a screen with poor resolution. In this situation the absence of direct eye contact is not so noticeable but one can hardly call this creative a high degree of social presence.)

Network bandwidth and latency are also very important issues for mutually immersive social presence. Majority of those who experienced videoconferencing knows how annoying audio-video asynchrony can be. While audio lagging by 200 ms becomes annoying, video lagging audio starts to become annoying at 50 to 100 ms delay. Unfortunately, current broadcast technologies do not allow avoiding latency.

Current videoconferencing systems, therefore, have technological drawbacks that do not allow interactive communication similar to real life.

1.2 State-of-the-art videoconferencing systems

First we want to summarize some of the aspects [6] that make the experience of remote locations immersive (create social presence for the participants):

- Direct eye contact is preserved
- Wide visual field
- Both remote participants appear life size
- Possibility for participants to see the upper body of the interlocutor
- High quality image and correct colour reproduction
- Audio with high S/N ratio
- Directional sound field
- Minimization of the video and audio signal asynchrony

– Availability of a shared working space.

However, it must be emphasized that it is technically very difficult to combine all these aspects in one videoconferencing system.

Actually a number of videoconferencing systems providing direct eye contact already exist. Examples are: Reciprocal Video Tunnel (B.Buxton and T.Norman), Ex'ovision/Eye Catcher (Netherlands), Teleportec (UK), Virtue Videoconferencing System (Germany-UK-Netherlands), MAJIC (Japan), Presence-box (Sweden), etc. They use different technologies allowing eye contact, such as a combination of the screen and a half-silvered mirror, image-processing technologies, rendering a "hologram" of the image, using a special screen (a thin transparent film with a number of very small hexagons printed on both faces). All these technologies were carefully analyzed in order to find out their advantages and disadvantages.

The disadvantage of such systems is, while preserving direct eye contact, they do not combine all factors creating social presence for participants listed above. Moreover, some of successful solutions are very expensive.

2 Design alternative

The key idea in the design alternative called Direct Eye Contact VCS is the introduction of a half-silvered mirror, together with a large back-projection screen (see Figure 1).

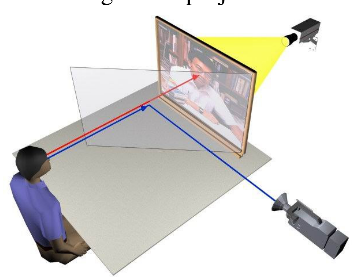


Fig.1. Design layout of the Direct Eye Contact VCS

The half-silvered mirror transmits some of the light, coming from one direction, but also reflects parts of it. This means that the viewer can watch the screen image behind the mirror, while his face is reflected by it in the direction of the camera. The camera is not visible to the other viewer. The back-projection screen allows the viewer watching the image projected on the screen from the opposite side of the viewer. These two characteristics, in addition to the specific implementation are the cornerstones of the system.

During the conference session, the viewer watched the image projected on the screen, gazing at the eyes of the interlocutor. The camera installed at the peripheral side of the table captures the mirrored image of the viewer. Via this relatively simple construction, the participants appeared to be always gazing at the eyes of each other, e.g. having eye contact. It is worth mentioning, that the large back projection screen allows showing life-size images, and the table can serve as a shared working space.

Benefits:

– Direct eye contact between the remote participants
– Maintaining eye contact while moving within the field of view of the camera
– Availability of a shared working space
– Life size image on the screen
– Participants see an upper body of the interlocutor
– Relatively cheap solution in comparison with state-of-the-art VCS
– Avoiding left-right reversal of the image

Drawbacks:

– Multiple participants are not possible
– The half-silvered mirror reduces luminance and contrast of the image

3 Measurement of social presence

In this paper we refer to social presence as to a feeling of being together during the communication with someone. All different approaches of presence measures can be divided into two general categories: subjective measures and objective corroborative measures.

3.1 Subjective measurements of social presence

Since presence mostly concerns the subjective experience of a person, the most widespread approach is to use subjective assessment methodologies such as interviews and questionnaires. They allow evaluating the user's affective judgments with respect to the system and its ability to support pleasurable, informal communication.

An example of such questionnaires is the IPO Social Presence Questionnaire (IPO-SPQ) developed at the Eindhoven University of Technology. [3] This questionnaire was filled in by the participants of the experiment. The IPO-SPQ is measurement scale consists of two parts: the Osgood's semantic differential technique and the subjective attitude statements with a 7-point agree-disagree Likert-type scale.

1. The Osgood's semantic differential technique is a popular method for the measurement of social presence. It is based on series of bipolar scales, such as "natural – unnatural, passive – active, personal–impersonal". For IPO-SPQ participants rated the communication media using a 7-point scale. The media that support a high degree of social presence are typically judged as personal, natural, friendly, warm or sociable.

2. As far as user's comments about media frequently relate to social presence [10], it allows using subjective attitude statements about system quality in attitude test. This approach was used in the second part of the questionnaire. The evaluation is based on the 7-point Likert-type scale (see Table 1).

Table 1. Example of the usage of the statements about the system in IPO-SPQ

How often do you feel in control over the interaction with your interlocutor?

Never	1	2	3	4	5	6	7	*Always*

3.2. Objective measurements of social presence

Subjective measures are potentially unstable [4]. That is why the most fruitful approach to measuring presence is likely to combine both subjective measures and objective corroborative measures.

Objective measures of presence focus on behavioural or physiological responses to media. "Nonverbal communication and intimacy behaviour seem to be of particular importance for supporting informal, mediated communication. The nonverbal channels seem to be less controllable than the verbal channels (i.e., they are more likely to "leak" information about feelings)" [3]. Other studies indicate factors of interpersonal intimacy: physical distance, smiling, eye contact, gestures, touching, vocal cues (e.g. tone of voice), turn-taking behaviour in dialogues, use of space and some other. Unfortunately, this approach is time consuming (thus not widely used), and does not incorporate applicable ready-to-use measuring technique.

Based on these factors and on the analysis of the video recordings of the experiments performed a checklist for the analysis of the social responses was created. The checklist allows performing a quantitative and a qualitative analysis of social responses that plays an important part in the daily interpersonal communication. Some examples of the responses contained at the checklist are provided below:

- Facial expressions: frequency of smiling, indicating surprise, etc.
- Head and body movements: frequency of nodding, indicating agreement or disagreement, shrugging, moving towards the interlocutor, etc.
- Gestures: pointing, expressing reaction, asking for reaction, attracting attention, drawing in the air while explaining, attempts to reach objects on the side of the interlocutor or pass objects to him, etc.
- Eye contact: frequency and duration
- Vocal cues: indicating "I am here", "I am listening" (mumbling while thinking, "mmmm", "uhu", "hm", etc.
- Fun: frequency of laughing and joking
- Turn-taking behaviour: frequency of interruptions

The hypothesis was: the system that provides a high degree of social presence should encourage using non-verbal and some verbal communication cues more often than the system providing a lower degree of social presence.

4. Design of the experiment

Two experiments were conducted in order to evaluate the degree of social presence provided by the Traditional VCS and by the Direct Eye Contact VCS. A comparative analysis of the systems was accomplished afterwards.

Twenty-six subjects took part in the experiments, in 13 pairs. Each couple took part only in one of the experiments. The duration of the test for each participant was about 40 minutes. All test sessions were videotaped for further analysis.

Videoconferencing set up description: Participants were seated in separate rooms equipped with a videoconferencing system. They were sitting behind a table in front of the screen and could see part of the interlocutor's table on the screen as well as the interlocutor sitting behind it. Microphones and loudspeakers provided audio communication. We used analogue cameras for the experiments because it allowed us to avoid video signal delays.

For the first experiment large TV sets were located on the table in both rooms. The camera was fixed on the top of the TV. One of the disadvantages of mounting the camera on top of the screen is that they do not provide a sense of looking into each other's eyes for the interlocutors. We call this system Traditional VCS because it is currently the most popular videoconferencing solution.

For the second experiment the room set up was identical to the previous one; only the videoconferencing system itself (TV set and camera) was removed and exchanged by the Direct Eye Contact VCS described above in the section 2.Design alternative.

Task description: There were two tasks to be carried out to the participants: object-oriented and social communication. Tasks were identical for the participants in each experiment, but the videoconferencing set up (condition) was different (see description above).

The object-oriented task consisted of construction with blocks. One of the participants (the instructor) was told to instruct the other participant (the builder) in rebuilding a block building on a toy foundation in accordance with the example provided and visible only to the instructor [2]. The participants were allowed to communicate using both verbal and non-verbal cues as they wished. Only the builder could manipulate the blocks. The instructor could observe the building and the interlocutor on the screen. The building consists of blocks of different colors, sizes, and shapes.

The second task was designed to evoke communication between the participants. They were asked to solve logical puzzles. One example of the puzzle is: "They loved each other very much and got married. They lived only for each other and didn't care about anything else. At the end they became millionaires. How could this happen?" The answer is: before that they were billionaires. The trick is that it is almost impossible to guess an answer from the description. So the person is forced to ask questions about the case. The participant who knows the answer is allowed to reply only "yes", "no" or "it does not matter".

5 Statistical analysis of the results

5.1 Evaluation of the IPO-SPQ measurement scales responses

A reliability analysis of the items of each response variable was performed. It appeared that the items in each scale are consistently measuring the same quality, as far as a reliability coefficient is 0.89.

A t-test for independent samples was run on the differences between conditions for different groups of subjects. The result of the test show that the difference between groups is significant ($t = -2.47$, $p = 0.012$). That means the group of subjects who used the Direct Eye Contact VCS evaluated the system more positively (mean: 5.79) than the group of subjects who used the Traditional VCS (mean: 5.04).

5.2 Video recordings analysis of the tasks performing

The video recordings of the experiment were analyzed according to the checklist for the analysis of social responses. Frequencies of the facial expressions, head and body movements, gestures, interruptions, vocal cues, frequency and duration of the eye contact, frequency of laughing and joking were counted for every participant and registered in the table. The result of the *t-test* for independent samples showed that the differences between groups during the accomplishing tasks are mostly significant at the chosen significance level 5% or less. Statistical results are presented in Table 2 and Table 3.

Table 2. Duration of eye contact (sec) averaged over subjects in the object-oriented and social communication tasks (N=12).

	Object-oriented task		**Communication task**	
Indicators	Condition 1	Condition 2	Condition 1	Condition 2
Total eye contact duration	50.6	99.2	351.8	423.4
Average eye contact duration	2.1	2.6	3.1	4.1

Table 3. Frequencies of behavioral indicators averaged over subjects in the object-oriented and social communication tasks (N=12).

	Object-oriented task		**Communication task**	
Indicators	Condition 1	Condition 2	Condition 1	Condition 2
Facial expressions	3.3	8.8	13.5	34.8
Head and body movements	1.8	7.2	12.2	36.0
Using gestures	9.2	17.3	11.8	27.3
Eye contact	26.5	40.0	117.9	106.0
Use of vocal cues	10.3	11.8	8.7	17.7
Laughing and joking	1.4	4.9	13.5	34.8
Turn-taking behavior	0.4	1.8	8.4	20.4

Object-oriented task performing: There is a significant difference between the groups in using all above mentioned categories of social responses (indicators) except the frequency of the use of vocal cues. Additionally the following social responses were noticed only at the group of the participants who used the Direct Eye Contact VCS (condition 2):

- Head movements in order to indicate direction
- Shrugging
- Moving towards the partner
- Attempts of the physical contact with the interlocutor in spite of the screen or attempt to reach objects at the side of the interlocutor or pass objects to him.

Social communication task performing: There is a significant difference between the groups in using all above mentioned categories of social responses (indicators) except of the frequency of eye contact. Attempts to gain physical contact with the interlocutor in spite of the screen as well as attempts to reach objects at the side of the interlocutor or pass objects to him were noticed only in the group of the participants evaluating the Direct Eye Contact VCS.

Based on the *t-test* results we can conclude that the participants of the evaluation of the Direct Eye Contact VCS (condition 2) used non-verbal communication and some verbal cues significantly more often then the participants of the evaluation of the Traditional VCS (condition 1).

5.3 Users' comments about the videoconferencing systems

Traditional VCS: Majority of the users described the system as nice, quite close to the "normal" communication, but mentioned "something was wrong". Most of them noticed the problem of the eye contact absence. They reported that this made them feel uncomfortable, and they stated that if their interlocutor would be somebody they did not know so well, the discomfort would be significant. All participants realized that only if they looked directly at the camera, their interlocutor perceived them looking directly at his/her eyes. Nevertheless they preferred to look at the screen to see reactions of the interlocutor. The illusion of a shared working space was mentioned as a benefit of the system.

Direct Eye Contact VCS: All participants were very enthusiastic about the system. The most often used comment was the communication was very natural and similar to the real life.

Almost all subjects mentioned they did not feel as if they were in different rooms. There was a strong illusion of them being together at the shared working table and they thought that the mirror or the screen was the only thing separating them from each other. Another interesting fact is that the people were thinking about the possibility of a physical contact during the videoconferencing. Some participants really believed that they could reach the legs of the interlocutor under the table as well as they could pass objects to each other. Moreover, two participants mentioned they really felt that interlocutor (who was not a very close friend) moved so close to them that sometimes they had a need to move away from him. It was just like in a daily communication when people defend their interpersonal space. The users of the Traditional VCS never mentioned anything like that.

6 Conclusions

The purpose of the study was to design a videoconferencing system that evokes a high level of social presence, creates a feeling of being together for the remote participants. In order to achieve this, we made a profound analysis of the literature and all existing at the moment videoconferencing systems. In our design we aimed to combine all known factors creating social presence for the participants and avoid disadvantages of the existing systems. The most serious challenge was to achieve direct eye contact and at the same time to add all other factors influencing immersive communication, such as wide visual field, life size image of participants, high quality image, correct color reproduction, etc.

Evaluation of the Direct Eye Contact VCS was conducted in comparison with the Traditional VCS. A combination of subjective and objective measurements was used in the design of the experiment. Although objective measures of social presence might provide precious information, it is time consuming (thus not widely used), and does not incorporate applicable ready-to-use measuring technique. Therefore we have developed a checklist for analysis of social responses for video data analysis.

Statistical results showed that the group of subjects who used the Eye Contact VCS evaluated the system more positively than the group of subjects who used the Traditional VCS. The participants of the evaluation of the Direct Eye Contact VCS used non-verbal and some verbal cues in communication much more often than the participants of the evaluation of the Traditional VCS in both task-oriented and social communication tasks. Examples of the non-verbal and verbal cues are: facial expressions, head and body movements, using gestures, frequency and duration of the eye contact, use of the vocal cues, laughing and joking, turn-taking behavior in the dialog.

Summarizing all findings we can conclude that the Direct Eye Contact Videoconferencing system provides higher degree of social presence then the Traditional Videoconferencing System.

References

1. Buxton, W., Moran, T. EuroPARC's Integrated Interactive Intermedia Faculty: early experience. *Proc. of the IFIP WG 8.4. Conference on multi-user interfaces and applications.* Amsterdam: Elsevier science publisher B.V. (North-Holland), (1990), 11-34.
2. Cremers, A. Reference to objects: an empirically based study of task-oriented dialogues. *PhD thesis*, Eindhoven Univerity of Technology, the Netherlands,1996.
3. De Greef, P., IJsselsteijn, W. Social presence in a home tele-application. *Cyberpsychology and behaviour,* volume 4, November, 2001.
4. Freeman, J., Avons, S.E., IJsselstein, W.A. Effects of sensory information and prior experience on direct subjective ratings of presence. *Presence: Teleoperators and Virtual Environments*, (1999), 8:1-13.
5. Issacs, E., Tang, J. Studying video-based collaboration in context: from small workgroups to large organizations. *Video-mediated communication,* (1997), 173-197.
6. Jouppi, N., Pan, M. Mutually-immersive audio telepresence. *Audio Engineering society, conventional paper*, 113th Convention, 2002.
7. Kendon, A. Some functions of gaze direction in social interaction. Acta Psychologica, (1967), 32: 1-25.
8. Okada, K., Maeda, F., Ichikawaa, Y., Matsushita, Y. Multiparty videoconferencing at virtual social distance: MAJIC design. *Computer Supported Cooperative Work.* (1994), 385-393.
9. Prussog, A., Muhlbach, L., Bocker, M. Telepresence in videocommunications. *Proc. of the Human factors and ergonomics society 38th annual meeting*, (1994), 25-38.
10. Short, J., Williams, E., Christie, B. The social psychology of telecommunication. London, 1976.

Magical Mirror: multimedia, interactive services in home automation

Luigi Ceccaroni and Xavier Verdaguer

:tmtfactory>
Edificio Forum de la Tecnología, c. Marie Curie s/n, 08042 Barcelona
luigi@tmtfactory.com, xavier@tmtfactory.com

Abstract. Several academic and industrial research groups are working to deploy home-automation systems and to improve their user interfaces. Based on user studies and prototype implementation, this paper presents the development of an innovating appliance that incorporates interactive services of leisure and information, offered through a high-quality user interface on the surface of a mirror. This appliance is thought to be both integrated in the programmable network of a home and used as an independent element. The prototype offers basic services, like interactive television, the presentation of personalized weather data, news and selections of musical tunes, and will incorporate some advanced functionalities, like an intelligent service of personal motivation and a reminder service.

1 Introduction

Modern man is basically sleepwalking if he is not multitasking. His lifestyle has evolved in such a way that optimizing time is the most important thing. The newest innovations show that we can -and therefore must- be online while watching TV, be e-mail—ready while driving the car, be taking calls while ascending Mount Everest, be watching television while shaving in front of the mirror. Most evenings most first world's kids do their homework on the computer while instant messaging friends and talking on the phone. The stereo is typically playing and they are probably downloading MP3s and DivX movies. Occasionally their foot extends to gently connect with the dog. There may be something wrong with this picture, but the kids are definitely not sleepwalking. The pressure to be wired -everywhere, always, fast- is hard to ignore [6].

Before the increasing interest of the industry and the market for the scope of home automation, the concept of intelligent human-habitat, managed by machines, has left the science-fiction domain to become, in the last years, a necessary investment-area for the companies that try to lead the sector in the mid term. But, is home automation really a necessity for consumers? It’s too easy, of course, to be a skeptic, to regard each new invention with a sense of wise detachment and, while praising its state-of-the-art capabilities, wonder aloud if we really need it. After all, nobody wants to risk missing real progress.

2 Related work

Several projects and public initiatives worldwide are dealing with ambient intelligence (e.g., [1] [4] [7] [9] [10] [12] [13]) and a few of them are developing interactive mirror-displays on which, based on personal preferences, different applications can be run. In particular, the HomeLab [14] at Philips Research [15] is working on a project which is very similar to the subject of this paper: the Magical Mirror project.

Several other groups dealt with the development of dialogue systems in diverse domains, not only home automation, with projects such as: SmartKom [16] on the design of new interfaces, which aims to help reducing the hesitations people presently feel upon using information technology and thus to make a contribution to the user friendliness and user centeredness of technology in today's information society; TRINDI [17] on task oriented instructional dialogue, which built a computational model of dialogue moves; Jupiter [18], a conversational system that provides up-to-date weather information over the phone, and other projects at MIT's Spoken Language Systems; the Conversational Interfaces project [7], whose aim is to build a general purpose dialogue system which supports multi-modal (i.e., speech and gesture) activity-oriented dialogues with devices, applications and services, and other projects at Stanford University's Center for the Study of Language and Information; and August [4], a Swedish spoken dialogue system.

3 Overview

Often, really useful technologies are not considered a need in a first phase (e.g., telephones, computers, PDAs) and part of the challenge in this project is demonstrating a technology to help creating new needs. In the information society, the emphasis is shifting from traditional computer interfaces (screen and keyboard) to computational modules embedded in appliances, hand-held devices and information systems, which often have to interchange significant amounts of multimedia content.

To be sure, current technologies that transmit images and voice fall far short of anything remotely realistic [6]: at present, home automation generally involves jerky video and canned sound. In Magical Mirror we have to deal with these limitations in order to develop an interactive appliance, which is able to provide the following realistic services, controlled by voice, in Spanish (English and Catalan, in the future), and presented with the support of a hand-designed, high-quality, advanced visual interface (see Fig. 1):

- S1. customized weather data
- S2. customized information on traffic and public transport
- S3. customized news (stock market, sports)
- S4. customized music management
- S5. interactive television (integration with the TVFinder service [2])
- S6. Internet access and search
- S7. personal agenda
- S8. home automation (control of other devices: lights, heating, oven)
- S9. communication among users (voice-message recording, email)

Fig. 1. Magical Mirror's high-quality interface (BCNMedia and **:tmtfactory>** are trademarks of the same company.)

4 Architecture

Magical Mirror is a meta-service, with the layered structure shown in Fig. 2, which is suitable to be integrated into a home automation network. Multiple infrastructures of communication exist that are candidates to form the backbone of a home-automation environment. It does not seem that, in the short term, any of them is going to constitute a broadly accepted standard and, in any case, it seems that there is a strong dissociation between the network of devices control and the network of data distribution (video, audio, and communications between computers). Nevertheless, it is important that the home-automation environment appears, before the user, as one interface. It would not be acceptable that he has to use an interface in order to handle home devices (lights, heating, alarms, oven), another one for communications devices (telephone, fax, email) and a different one for audio and video devices (TV, music). In order to obtain this unity in the interface, a high level of interaction, among the heterogeneous infrastructures of communications that will support it, is needed.

4.1 Device layer

At the physical-device level, interactivity is possible through a reception-emission system with the following characteristics:

- reception: microphone arrays (for voice recognition), micro-camera (for image recognition [future work]), touch screen (for user input [future work]).
- emission: retro-projection of images and videos, retro-projection of text, micro-speakers for voice synthesis and music.

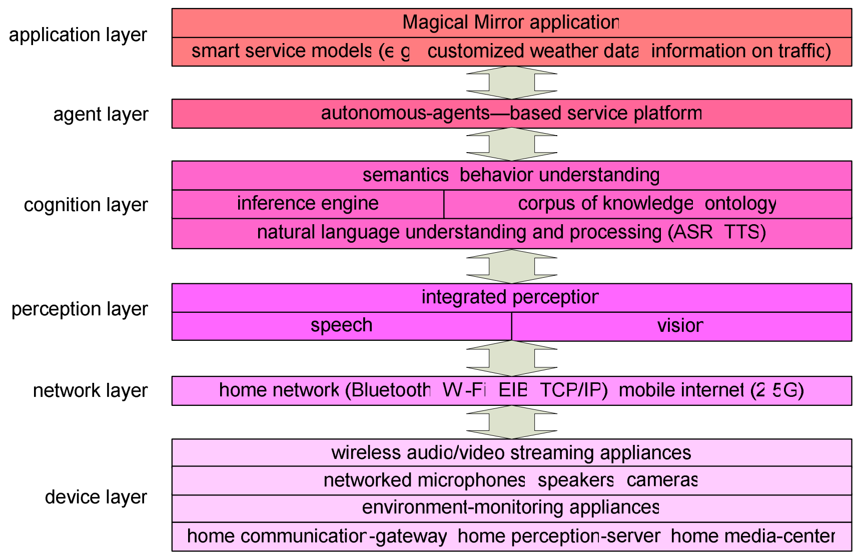


Fig. 2. Layered structure of the Magical Mirror service

4.2 Network layer

As home technologies expand to encompass computer and communication networks, standards and specifications are quickly multiplying. In Table 1 a list of the main ones developed by alliances and working groups (non proprietary) is presented. In Magical Mirror, the control of the different devices that compose the home-automation environment is, for the moment, simulated and will be mainly based on equipment with the following connection standards: Bluetooth (between PDAs and Magical Mirror), IEEE 802.11g and IEEE 802.15 (among Wi-Fi--enabled computers and devices). With the purpose of connecting all the devices that integrate the environment and to provide a suitable base of operation, a hardware architecture that is able to manage and integrate the different components, independently of the network to which they are connected, will be needed. In addition, it seems very suitable to provide means of connection between the Web world and the home-automation environment. As solution, we will work at the integration of different networks using the TCP/IP protocol. Among the advantages of having access by means of the TCP/IP protocol to elements connected by different networks, we can enumerate:

- Extending the scope of home automation to Web services and making available a global, remote access.
- Having access to the home-automation environment by means of WAP (wireless application protocol)-based applications, used in the mobile telephony, which will allow possibilities of ubiquitous control.
- Guaranteeing that we can have all kinds of connected equipment, without being restricted to a specific interconnection protocol.

Standard, specification	Media	Description
Bluetooth [http://www.bluetooth.com]	RF	Bluetooth is the codename for a technology specification for small form-factor, low-cost, short-range radio links between PDAs, PCs, mobile phones and other portable devices. The Bluetooth Special Interest Group is an industry group consisting of leaders in the telecommunications and computing industries that are driving the development of the technology and bringing it to market.
EIB [http://www.eiba.com]	Twisted Pair	The European Installation Bus is a home-automation system developed under the auspices of the European Union with the objective of balancing the imports of similar technologies that were taking place from the Japanese and the North American market. It links sensors and actuators to building systems that control HVAC (heating, ventilating and air conditioning), security, access, and life safety.
HomePlug Alliance [http://www.homeplug.com]	Power Line	Created to set a technology specification for home power-line networking and to promote its wide acceptance in the marketplace. The alliance's objective is to enable and promote rapid availability and adoption of cost effective, interoperable and specification-based home power-line networks and products enabling the connected home.
IEEE 802.15.4 [http://www.ieee802.org/15/pub/TG4.html]	Wireless	The IEEE 802.15 TG4 is chartered to investigate a low data rate solution with multi-month to multi-year battery life and very low complexity. It is intended to operate in an unlicensed, international frequency band. Potential applications are sensors, interactive toys, smart badges, remote controls, and home automation.
Konnex [http://www.konnex-knx.com]	All	The common association of EIBA (European Installation Bus Association), BCI (BatiBUS Club International) and HES (European Home System).
WECA [http://www.wirelessethernet.org]	RF	Wireless Ethernet Compatibility Alliance's mission is to certify interoperability of IEEE 802.11 high-rate products and promote that standard for the enterprise, small business and the home. IEEE 802.11 offers connectivity at speeds up to 11 Mbps.
ZigBee [http://www.zigbee.org]	Wireless	The ZigBee Alliance is an association of companies working together to enable reliable, cost-effective, low-power, wirelessly networked, monitoring and control products based on an open global standard.

Table 1. Main home-automation/network standards and specifications, developed by alliances and working groups, deployable in Europe

Through the protocols described, Magical Mirror can be connected to different devices that can be found in a home, becoming, in a flexible way, an integral part of a home-automation environment (see *modality **c*** below).

4.3 Perception and cognition layers

Voice and image feeds are collected in the perception layer and processed (voice only, at the moment) in the cognition layer. The main challenges during perception are:

- automatic speaker-identification;
- voice-silence detection;
- robustness with respect to the environmental noise (basically, water noise, echo and other voices).

Research and commercial tools are used for automatic speech recognition (ASR) and text to speech (TTS) processing. VoiceXML (VXML)[1] is used in service models to represent grammars, speech acts and dialogues. Here is a simple example of VXML encoding and a sample human-computer interaction (translated from Spanish to English) in the case of customized music management:

```
<form>
  <field name="artist">
     <prompt>Whom of these artists World you like to lis-
             ten to?: Anticappella? Aphex Twin? Eiffel
             65? Elio e le Storie Tese? o Leftfield?
     </prompt>
     <grammar        src="artist.grxml"
                     type="application/srgs+xml"/>
  </field>
  <block>
     <submit next="http://www.tmtfactory.com/art2.asp"/>
  </block>
</form>
```

A field is an *input* field. The user must provide a value for the field before proceeding to the next element in the form.

C (computer): Whom of these artists World you like to listen to?: Anticappella? Aphex Twin? Eiffel 65? Elio e le Storie Tese? o Leftfield?

H (human): Mogwai.

C: I didn't understand what you said. (default message.)

C: Whom of these artists World you like to listen to?: Anticappella? Aphex Twin? Eiffel 65? Elio e le Storie Tese? o Leftfield?

H: Aphex Twin

C: (continues in document art2.asp)

4.4 Agent layer

The highest layers of the architecture are based on independent agents which interact among them and with the user, and which accomplish the intelligent control of diverse devices, sensors as well as actuators (in phase of study for *modality* ***c***). In the project, the design of the architecture of independent agents for the implementation of home automation has been carefully studied. Agent-based architectures present, potentially, a series of fundamental advantages, such as modularity, reusability, scalability and dynamic reconfiguration. Each agent has a basic functionality and in addition

[1] See [http://www.voicexml.org].

is equipped with certain *decision and adaptation* capabilities, in such a way that it can improve its efficiency on the basis of information that receives directly from other agents or indirectly through three kinds of central support knowledge-components, a *base of facts and knowledge* that is common to all the agents, a **domain ontology** (*DomOnt*) and a number of **service ontologies** (*ServOnt*). An ontology is a model of the world in which the different agents of the system work. The ontology is made up of a hierarchy of entities, which represent buildings, rooms, elements that compose the rooms, users, tasks that these users are accomplishing, and any other information of interest for the community of agents which belong to the home-automation environment. Agents use the *DomOnt* as a mechanism for obtaining data they need about the environment and, in turn, deposit in the ontology their results so that other agents can take advantage of them.

While the *DomOnt* contains a representation of the environment and (in its instances) of its state, the *ServOnt* models the information flow that is exchanged among the different agents part of the environment, influencing the language and the communication protocols between them. The agents channel the information according to the flow model contained in the *ServOnt*. For example, if an agent deduces that the user requires that a certain flow of video (the output of a camera) is transferred from the hall to the mirror of the bath, the only thing it has to do is to change some parameters of the model of information flow contained in the *ServOnt*. The agents controlling the camera in the hall and the computer in the bath respond to this change in the ontology one emitting the video flow towards the computer in the bath and the other beginning to emit this flow in the Magical Mirror.

An important element of the Magical Mirror project is to allow the addition to the home-automation environment of any sort of new agents and devices developed by third parties using the same standards. With this aim, we use OWL [3] and OWL-S [9] (both based on XML) as the languages of definition of the *DomOnt* and the *ServOnt*, respectively. This assures that the requirements that a new agent will have to fulfill to join the home-automation environment will be minimum, standardized and of easy implementation. With respect to the agent systems' architecture, the general requirements are: (1) compliance with the FIPA standard of communication and interaction, and (2) ability to use OWL ontologies.

5 Personalization

In Magical Mirror, due to the presence of a number of different services, personalization is a long and challenging process to be carried out. The personalization methodology that we use is defined by the following steps:

1. Creation of domain models with formal semantics, via ontologies:
    a. user domain;
    b. television-programs domain;
    c. music domain;
    d. weather domain;
    e. …

2. Creation of a recommendation algorithm, which associates instances of the user-domain ontology with instances of the service-domain ontologies (e.g., in the case of television, in the form of personal programming guides). This step includes a certain degree of learning and constitutes the core of the personalization process.
3. Creation of an exploration algorithm (a search engine) for the different services, which takes into account the user profile.

Domains are formalized using ontologies: the static part of each domain is represented by a subset of *DomOnt*, while the dynamic part of service domains is represented by a *ServOnt*. User profiles, i.e. instances of the user ontology, are created through a Component for Creation of User Models (CCUM), which reuses a module developed for the TVFinder system [2]. In Fig. 3, an example of the interface for the capture of user preferences is shown. In this particular case the user is asked to explicitly express his preference with respect to the *ingredients* (such as love, action and suspense) of his favorite TV series. Explicit preferences are just one kind of preferences used by the CCUM; other kinds include stereotypical and implicit preferences.

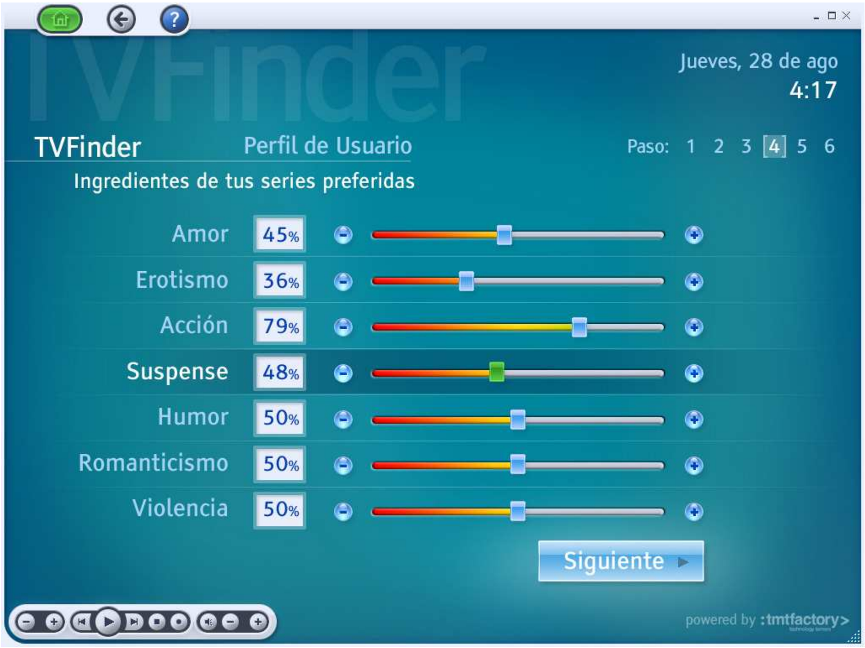


Fig. 3. Interface representing a step in the creation of the user profile (Copyright © 2003, :tmtfactory>)

A code is provided to each user in order to access his personalized interface with the Magical Mirror (e.g., music libraries and TV channels). The identification of the user is important also for improving the voice, image and gesture recognition.

6 Modalities

Magical Mirror can work in three modalities:

[a] independent;
[b] connected to the Internet (in particular, to an online data-base);
[c] connected to the Internet and to a home-automation network.

The data-base includes audio-visual contents and preferences of the users. After a process of initialization and content upload, the way to update the system depends on the modality of the mirror:

[a] learning;
[b] updates from a content portal + learning;
[c] updates from a content portal and the home-automation network + learning.

7 Natural-language dialogue management

Among the necessary technologies to be integrated in the Magical Mirror, so that it is possible to provide the type of personalized services described above, we are building a preliminary speech interface. To this aim, the main artificial-intelligence technique used is an agent for dialogue management, i.e. an agent for natural-language conversation, aware of the profile and preferences of the user.

One of the objectives of the project is that the developed system is able to maintain robust dialogues in natural language. By *robust dialogue* we mean one in which the user of the system can use general expressions (i.e., not restricted to a predetermined language of commands) to access the functions and services of the mirror. Naturally, this objective is excessively ambitious if we pretend that the scope of these dialogues is an ample domain without restrictions. Nevertheless, if the domain is sufficiently restricted, the complexity of the problem is tractable with the present technology of natural language processing. Therefore, a set of dialogue domains has been identified that corresponds with the services offered and is sufficiently restricted to allow such processing.

The architecture of the system is organized around the *VoiceXML framework* and the documents that comprise the *dialogue manager*. These documents are generated with dynamic content coming from the *data bases* of the system: one for the data of the services and one for the user profile. The choice of VoiceXML turns the dialogue manager independent of both the signal-processing technology and the logic used in the application.

The value of the technology being developed does not primarily consist of an advance in each one of the specific techniques of interaction (intelligent processing of the language, agents), but rather in their correct integration to form a coherent whole. Therefore, the process of design and development of a correct architecture of integration of all the modules constitutes an essential part of the project.

8 Task identification by means of artificial vision

As future work, we plan to integrate image-processing techniques which will allow more complex decisions; in particular: (1) identification of the context in which the user carries out her tasks and (2) gesture recognition. The first one consists of determining the number of individuals that are in the room and their arrangement. This information will have to be congruent with the rest of the information received by the home-automation network as well as complementary to it. Other agents can then use this information to adjust the behavior of the network. On the other hand, the analysis of gesture consists of making a suitable segmentation so that the user will be able to control the behavior of the system with clear, natural gestures. This capability of interaction will complement the agent for dialogues in natural language.

Furthermore, the personalization of the interaction will increase the robustness, because the mirror will more easily respond only to the voice and gestures of the person that is logged in to use the Magical Mirror.

9 Use case

Fig. 4. Magical Mirror at a public exposition

A demonstration version of Magical Mirror was shown at various expositions around Catalunya (Spain), in 2003, to check the reaction of the public (see Fig. 4), and received a positive feedback. We hand-designed graphical interfaces on Magical Mirror for interactive television, a multimedia player, weather data and news services. Since we could not yet control the actual applications by voice commands, we simulated control with a laptop (connected to the interface of the mirror) that responded to commands and signals from the users by playing multimedia content that closely matched the response expected of the actual applications. Most people attending the expositions, including kids and seniors, interacted with the mirror speaking in complete comfort.

The average time spent interacting with it was about 5 minutes per person; longer in the case of kids. The attribute that attracted the greatest deal of attention was not the function as information-services provider, but rather the capability of differentiating among interlocutors. In other words, the high degree of personalization and humanity was the most appreciated feature identified by the public. Visitors could listen to their favorite songs and see images corresponding to their specific requests. The importance of the use of stereotypes was absolutely clear at the moment of the mirror's deployment. About 80% of the visitors requested the same artists and type of images; and these artists corresponded to the ones at the top of the charts of the season. Sport, and specifically soccer, was often the object of search requests.

10 Conclusions

A prototype of a device that incorporates interactive services of leisure and information, offered to the user through a natural interface in form of language and the visual superposition of images on the surface of a mirror, is being developed. The main services, controlled by voice, in Spanish, and presented with the support of an advanced visual interface are: customized weather data, customized news, customized music management, interactive television, Internet access and search. All services are in various phases of research and prototypical development. Nonetheless, a comprehensive prototype, called Magical Mirror, was deployed at various public expositions, where people could interact lively with it. In general, the public showed a positive disposition at adopting a similar mirror at home, given that the price were not much higher then the one for a traditional appliance, demonstrating that people did not consider the apparatus as something completely new and different, and that the additional technology was installed in a satisfactorily transparent way. Additional user studies are in progress to confirm this.

Acknowledgements

This work was conducted as a part of the Magical Mirror project. The interactive television interface was also conducted as part of the TVFinder project [2]. Various people at **:tmtfactory>** did a portion of the work on the prototype interface. This work was funded in part by grant no. FIT-070000-2003-626 from the Spanish Ministry for Science and Technology, through the PROFIT program. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the Spanish Ministry for Science and Technology.

References

[1] Aarts, E. H. L., Roovers, R.: IC Design Challenges for Ambient Intelligence. Proceedings of Design Automation & Test in Europe (DATE 2003), Munich, Germany (2003)

[2] Ceccaroni, L., Verdaguer, X.: TV finder: una aproximación semántica a la televisión interactiva. Proceedings of the workshop on Ubiquitous computation and

ambient intelligence of the Conference of the Spanish Association for Artificial Intelligence (CAEPIA 2003), Donostia, Spain (2003)

[3] Dean, M., Schreiber, G., Bechhofer, S., van Harmelen, F., Hendler, J., Horrocks, I., McGuinness, D.L., Patel-Schneider, P.F., Stein, L.A.: OWL Web Ontology Language Reference. W3C Proposed Recommendation (15 December 2003)

[4] Ducatel, K., Bogdanowicz, M., Scapolo, F., Leijten, J., Burgelman, J-C.: Scenarios for Ambient Intelligence in 2010. ISTAG final report for the European commission (2001) Available at [http://www.cordis.lu/ist/istag-reports.htm] (921kb), accessed 23 Apr 2004

[5] Gustafson, J., Lindberg, N., Lundeberg, M.: The August spoken dialogue system. Proceedings of Eurospeech'99, Budapest, Hungary (1999)

[6] Ignatius, A.: We have contact. Time, June 4 (2001)

[7] IST Advisory Group: Ambient Intelligence: from vision to reality. ISTAG draft consolidated report for the European commission (2003) Available at [http://www.cordis.lu/ist/istag-reports.htm] (225kb), accessed 23 Apr 2004

[8] Lemon, O., Gruenstein, A., Battle, A., Peters, S.: Multi-tasking and Collaborative Activities in Dialogue Systems. Proceedings of 3rd SIGdial Workshop on Discourse and Dialogue, Philadelphia (2002) 113-124

[9] Lindwer, M., Marculescu, D., Basten, T., Zimmermann, R., Marculescu, R., Jung, S., Cantatore E.: Ambient Intelligence Visions and Achievements: Linking Abstract Ideas to Real-World Concepts. Proceedings of Design Automation & Test in Europe (DATE 2003), Munich, Germany (2003)

[10] Marzano, S., Aarts, E.: The New Everyday: View on Ambient Intelligence. Uitgeverij 010 Publishers. ISBN: 9064505020 (2003)

[11] McIlraith, S., Martin, D. Bringing Semantics to Web Services. IEEE Intelligent Systems, 18(1) (2003) 90-93 OWL-S's latest version available at [http://www.daml.org/services/owl-s/], accessed 23 Apr 2004

[12] Myers, B. A.: Graphical User Interface Programming. In: Tucker, A. B. (ed.): Computer Science Handbook. 2nd edn. CRC Press, Inc., Boca Raton, FL, USA. To appear on 25 Jun 2004, ISBN: 158488360X (2004)

[13] Nichols, J., Myers, B., Harris, T. K., Rosenfeld, R., Shriver, S., Higgins, M., Hughes, J.: Requirements for Automatically Generating Multi-Modal Interfaces for Complex Appliances. Proceedings of the Fourth IEEE Inter-national Conference on Multimodal Interfaces, Pittsburgh, PA, USA (2002) 377-382

[14] Philips HomeLab [http://www.research.philips.com/InformationCenter/Global/FArticleDetail.asp?lArticleId=2753&lNodeId=931&channel=931&channelId=N931A2753], accessed 23 Apr 2004, is part of Philips Research Laboratories

[15] Philips Research [http://www.research.philips.com], accessed 23 Apr 2004. Prof. Holstlaan 4, 5656 AA Eindhoven, The Netherlands

[16] SmartKom project: Dialog-based Human-Technology Interaction by Coordinated Analysis and Generation of Multiple Modalities [http://www.smartkom.org], accessed 23 Apr 2004

[17] TRINDI project: Task Oriented Instructional Dialogue [http://www.ling.gu.se/projekt/trindi/], accessed 23 Apr 2004

[18] Zue, V., Seneff, S., Glass, J., Polifroni, J., Pao, C., Hazen, T.J., Hetherington, L.: Jupiter: A Telephone-Based Conversational Interface for Weather Information. IEEE Trans. on Speech and Audio Processing, 8 (1) (2000) 85-96

Personalized Information Delivery in Dynamic Museum Environment by Implicit Organizations of Agents

Tsvi Kuflik, Paolo Busetta, Loris Penserini, Paolo Bresciani and Massimo Zancanaro

ITC-irst, Via Sommarie 18, POVO (TN), Italy.
{kuflik,busetta,penserini,bresciani,zancana}@itc.it

Abstract. Today's complex distributed systems call for novel methods for modeling, design and implementation. The "active museum" being developed within the PEACH project is an example of such a system. In order to model the system, a new pattern called *Implicit Organization* was defined that supports the coordination requirements of the agents that compose the system. This pattern was applied within the PEACH museum's visitor guide; its implementation is based on the "channeled multicast" capability offered by the *LoudVoice* infrastructure. The system analysis and design, reported in this paper, demonstrates the need and use of the *Implicit Organization* pattern. We discuss the planned implementation of PEACH using a general purpose environment that supports the development of such highly distributed and dynamic systems.

1. Introduction

Nowadays, complex distributed applications that are emerging in areas such as e-Business, e-Government, and the so-called *ambient intelligence* (i.e., "intelligent" pervasive computing [7]) call for novel methods for modeling, design and implementation. The mobile visitor guide system being developed within the framework of the PEACH project[1] is an example of such systems.

Although many research projects are exploring the possibilities offered by Personal Digital Assistants as multimedia guides (among others, [6,8,12,15]), there have been very few attempts to investigate the architectural issues emerging from the embedding of a mobile guide into an overall "intelligent museum". In an "intelligent museum," different devices—mobile or fixed—have to interact with a variety of services. It's not just that the number of devices in the museum cannot be easily predicted, but also that the available services may change over time. Moreover, it is desirable that services be provided in a user–adapted and context–adapted way.

[1]http: //peach.itc.it. This work was supported by the PEACH project, funded by the Autonomous Province of Trento.

Generally speaking, active environments have some characteristics that make them substantially different from traditional computing and HCIs:

- Multiple users may be in a single place, interacting with different applications simultaneously.
- The set of users changes dynamically over time.
- Users are unaware that the environment is formed by many components. Therefore, they interact with the environment as if it were a single, monolithic system.
- Services are provided by a variable set of components that can join and leave the environment and can be running anywhere.
- Services provided by components can (partially) overlap; therefore, these components need to coordinate. They might need to decide, for instance, who provides a specific service, and how.

In the context of the PEACH project, we are investigating the concept of "active museums". An active museum is a form of "active environment" [10], and can be seen as a large scale multi-user, multi-media, multi-modal system. Adopting techniques based on the notion and use of autonomous software agents— at both the design and implementation levels—is central to providing the functionalities and qualities we expect from our system. Using the PEACH system, museum visitors can request information about exhibits. This may be provided by a variety of information sources and media types (museum servers, online remote servers, etc.), and presented by a variety of clients (e.g., hand-held devices such as PDAs, desktop PCs, and more).

Let us consider the following scenario, illustrating parts of the above: A visitor requests the presentation of some artifact at an exhibition. The primary interaction device used by the visitor is a PDA (that corresponds to what we will call a *User Assistant* agent). Nevertheless, while walking around, the visitor may approach some presentation devices that are more comfortable and suitable to handling the presentation than her PDA, e.g., in terms of pixel resolution or audio quality. How can the presentation be generated and presented to her? We assume that there are different *Presentation Composers* agents capable of building presentations for the visitor, and that each *Presentation Composer* relies on *Information Mediator* agents to provide the information required for the generation of the presentation generation Moreover, we also assume that each *Presentation Composer* is able to proactively propose the best services (in terms of available or conveniently producible presentations) to the *User Assistant* (which interact with the *Presentation Composers* on behalf of the visitor), as well as to other presentation devices in the environment. As well, we expect that all the services are "dynamically validated". In other words, because the environment and the user location are changing quickly, only appropriate services are considered.

As suggested by the above scenario, our "active museum" environment, poses new challenges for systems development, starting with system requirements analysis and modeling, through architectural and detailed design, and, eventually, implementation. Traditional agent patterns (such as *"Broker"* or *"Matchmaker*) require central coordination, and as such, are limited in their ability to support the requirements of the "active museum" (these limitations are discussed later). The authors suggest two novel agent patterns that address this issues: *Organizational Matchmaker* and *Implicit Organizations*. Previous work suggested a practical approach and infrastructure for effective, yet unreliable, *channeled multicast* communication, called *LoudVoice*

[2,3,4], to implement the idea of implicit organizations. This infrastructure allows agents to collaborate by means of *overhearing*, a technique suitable to highly distributed and dynamic environments such as our "active museum".

This paper demonstrates how, by using the patterns mentioned above, we bridge the gap between requirements analysis and design of such a system. We show how *implicit organizations* support the requirements of the PEACH museum visitors guide, and how it can be implemented using the *LoudVoice* infrastructure.

In our work, we adopt the *TROPOS* agents-development methodology [1].

The rest of the paper is organized as follows: Section 2 provides an introduction to *TROPOS.* Section 3 briefly presents the limitations of traditional agent patterns. The new architectural patterns used are presented in section 4. the idea of *implicit organizations* is presented in section 5, and it's implementation and *LoudVoice* infrastructure are presented in section 6. Section 7 presents the PEACH museum visitors guide modeled using *TROPOS,* and Section 8 presents the design of *Implicit Organization* to be implemented over the *LoudVoice* infrastructure. Section 9 summarizes our conclusions.

2. An Introduction to TROPOS

TROPOS adopts high-level requirements engineering concepts founded on notions such as actor, agent, role, position, goal, softgoal, task, resource, belief and different kinds of social dependencies between actors [1,5]. An actor represents any active entity, be it individual or collective, or be it human or artificial. Thus, an actor may represent a person or a social group. Or it may represent an artificial system, such as an interactive museum guide, or any of its components at different levels of granularity (e.g. software and hardware modules or software agents). Actors may be further specialized as *roles* or *agents*. In particular, an *agent* represents a physical instance of actor that performs activities assigned to it. A *role*, instead, represents a specific function that, in different circumstances, may be executed by different agents. We say that the agent *plays* the role. Actors are used in TROPOS to describe different social dependencies and interaction models. In particular, Actor Diagrams describe the network of social dependencies among actors. An Actor Diagram is a graph, where each node may represent an actor, a goal, a softgoal, a task or a resource. Links among nodes may be used to form paths of the form: *depender* $\rightarrow$ *dependum* $\rightarrow$ *dependee*, where the *depender* and the *dependee* are actors and the *dependum* is a goal, a softgoal, a task or a resource. Each path between two actors indicates that one actor depends on the other for a goal/softgoal/task/resource (the *dependum*) so that the former may attain it. The type of the dependum describes the nature of the dependency: goal dependencies are used to represent delegation of responsibility for fulfilling a goal; softgoal dependencies are similar to goal dependencies, but their fulfillment cannot be defined precisely (For instance, appreciation is subjective, or the fulfillment can occur only to a given extent); task dependencies are used in situations where the dependee is required to perform a given prescriptive activity; and resource dependencies require the dependee to provide a resource to the depender. Finally, TROPOS spans four phases of software engineering activities [1,5]: early requirements, late requirements, architectural design, and detailed design. Its key premise is that agents and goals can be used as fundamental concepts for all these four phases, as well as for the implementation phase, particularly if agent-oriented programming is adopted.

3. "Organizational Matchmaker" and "Implicit Organization" Agents Patterns

As suggested by the above scenario, our "active museum" environment, poses new challenges for systems development, starting with system requirements analysis and modeling, continuing through architectural and detailed design and ending with implementation. Such a scenario calls for architectural flexibility in terms of dynamic group reconfiguration.

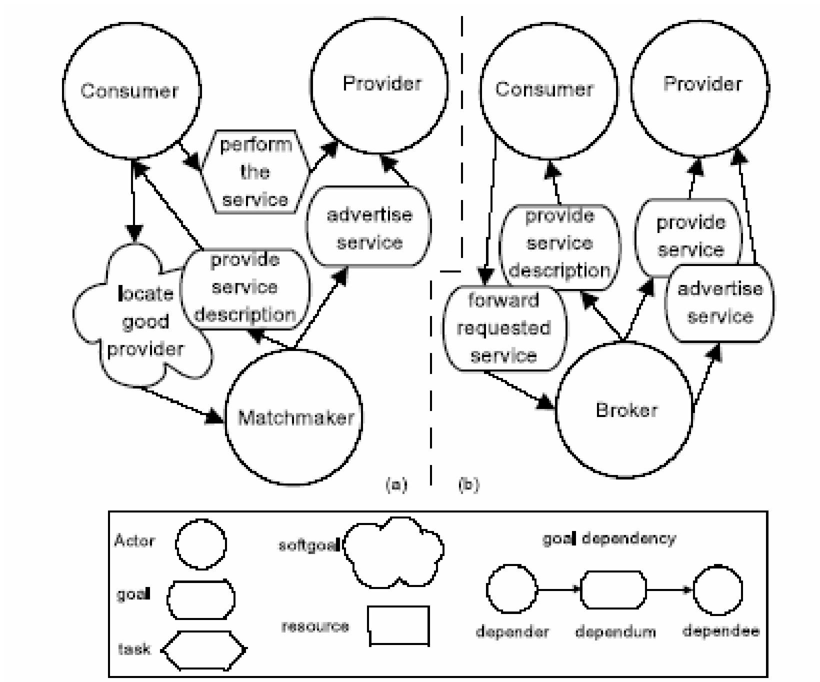


Fig. 1. "Broker" and "Matchmaker" Agent patterns.

Traditional approaches allow for intentional relationships and request/response communication protocols among single agents only, and not among group of agents [9, 11], as presented by Figure 1. More specifically, we may assume that the *User Assistant* starts an interaction session that triggers the involvement of a group of system actors with the same ability of *Presentation Composer,* which, in turn, triggers the involvement of a group of system actors (of unknown size) with the same ability as the *Information Mediator*. Each *Presentation Composer* relies on a *User Modeler* to correctly build a user-tailored presentation. Therefore, the architecture has to adopt group communication in order to support an "intelligent" pervasive computing model among user-assistant devices and the information/service providers. To cope with these new challenges, we can imagine that the system agents exploit a form of "implicit communication", where they can autonomously build up service-oriented organizations (SOO). In order to satisfy a request as best they can at the time. (We call *service oriented organization* a set of autonomous agents that, in a given location at a given time, coordinate to provide a service.) This is not possible by means of traditional approaches that adopt request/response based communication styles (e.g., [13]). In fact, as shown in Figure 1, using classical *matchmaker* and *broker* approaches, we inherently assume at the beginning that there is an advertise service dependency (e.g., based on a registration phase) that forces system actors to rely on a centralized computing model (more details can be found in [14]).

4. "Organizational Matchmaker" and "Implicit Organization" Agents Patterns

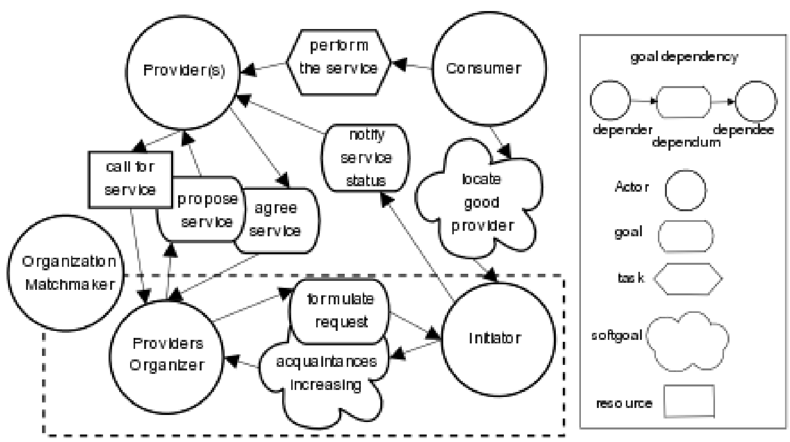


Fig. 2. Service-Oriented Organization pattern.

In dealing with pervasive computing scenarios, several works [7,14] have discovered new requirements that cannot be captured by means of classical *Matchmaker* and *Broker* patterns [9,11,13] in modeling our system requirements; hence, new architectural patterns are introduced, named *Service Oriented Organization* (SOO) and *Implicit Organization* patterns to cope with these requirements. In this section, we only give an overview of the new patterns, assuming classical Matchmaker and Broker patterns are known to the reader [9,11,13].

Figure 2 presents the SOO pattern as an extension of the traditional Matchmaker as TROPOS Actors Diagram. Here, the traditional matchmaker role is replaced by the ***Organization Matchmaker***, which is further decomposed into two system actors: the ***Providers Organizer*** and the ***Initiator***. The dependencies between ***Consumer*** and the ***Organization Matchmaker*** and between ***Consumer*** and the ***Provider***(s) reflect those of a traditional Matchmaker. The main novelty is the absence of any advertised service, typical of the traditional Matchmaker. In fact, our scenario calls for dynamic group reconfiguration (since agents come and go, and organizations may form and dissolve), and this calls for a proactive and, more importantly, dynamic capability of service proposal, based on real service requests or needs. This new requirement is fulfilled by the proposed architectural solution in which the ***Provider***(s) depends on the **Organization Matchmaker**, and more specifically on the ***Providers Organizer***, to obtain a Call for Service. On the basis of such calls, the ***Provider***(s) may signal which services they can currently provide. Thus, the ***Providers Organizer*** depends on the ***Provider***(s) for *proposed services* and, vice versa, the ***Provider***(s) depend on the ***Providers Organizer*** for the final agreement on service provisioning (Agree Service).

The Implicit Organization pattern improves the previous pattern. It minimizes the load on the communication between the ***Consumer*** and the other system actors. This may be required because of possible communication physical constraints such as wireless bandwidth or possibly discontinuous/unreliable connections. Moreover, this requirement supports and enhances the environment awareness, that is, a more system pro-activeness to satisfy user needs. In this case, it seems appropriate to exploit the implicit communication paradigm towards the adoption of Implicit Organizations [4]. That is, we can consider having Implicit Organizations playing a kind of broker role. In other terms, each time the system perceives the visitor's information needs, the system actors set up a SOO, which, in addition to the already presented matchmaking capabilities, can also manage all the service I/O process; namely, SOO is able to autonomously and proactively cope with the whole service providing process life cycle. Such system ability enhances the ambient intelligence awareness, a system requirement that cannot be captured by adopting traditional agent patterns, characterized by single actors that follow simple request / response communication protocols, as in [9,11,13]. Figure 3 introduces an Implicit Organization pattern as a refinement / adaptation of the SOO pattern. Here, the Provider(s) are part of the organization itself, which, as noted above, plays the role of an Organization Broker. Thus, the latter include both the Providers Organizer and the Provider(s) themselves (more details can be found in [14]). Another difference from

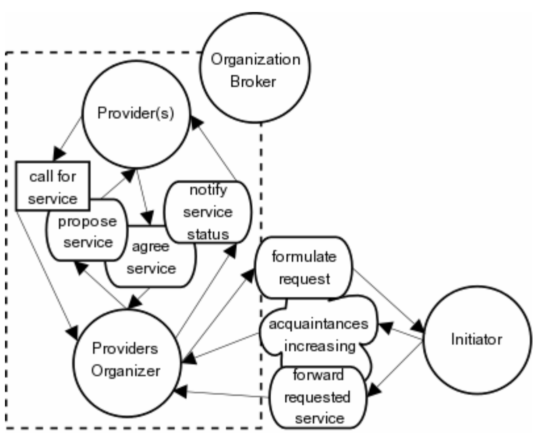


Fig. 3 Implicit Organization Pattern

Figure 2 is that the Implicit Organization pattern incorporates in the ***Initiator*** role both the role of a ***Consumer*** (e.g., a user or its visitor guide device) and the role of a system actor that perceives the user's needs. As already said, this is a consequence of the fact that, in an ambient intelligence some system actors concurrently play the consumer and initiator roles, which allows the system to enhance autonomy and pro-activity skills.

5. Implicit Organizations and Dynamic Group Communication in the Museum Environment

In order to illustrate the need and the idea of *Implicit Organization*, let us consider the following simplified scenario presented in the sequence diagram of Figure 4: A request for presentation is initiated by a ***User Assistant*** on behalf of its user. The request is addressed to the Implicit Organization of ***Presentation Composers***. Presentation composers have different capabilities and they require different resources. Hence every presentation composer requests user information from the Implicit Organization of ***User Modelers*** and presentation data from the Implicit Organization of ***Information Mediators***. In turn, the Implicit Organization of ***information mediators*** holds a conversation. Every member suggests the service it can provide. The "best" service is selected and returned, as a group decision, to the requesting ***Presentation Composer***. At this stage, the ***Presentation Composers*** request additional information from the Implicit Organization of ***User Assistants***, regarding the availability of assistants capable to show the presentation being planned. When all the information has been received,

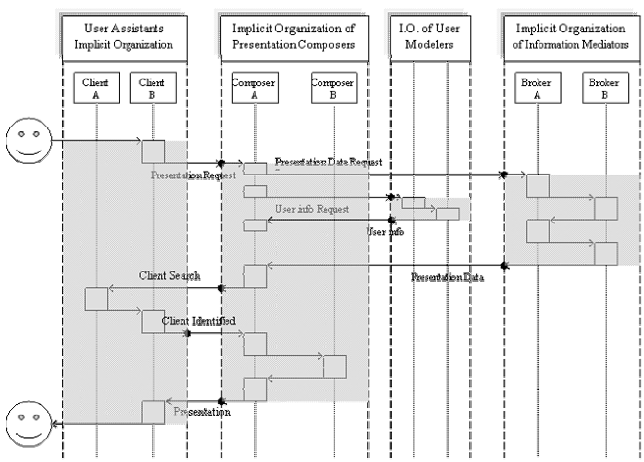


Fig. 4. Interaction of implicit Organizations (Between two vertical dashed lines there is an "Implicit Organization", its' name appears in the upper rectangle and the shaded rectangles represent interaction between its members. Requests are represented by arrows, where the dots represent "organizational border" – a request is sent to an organization, not to a specific member and so is also the response returned from the organization).

the Implicit Organization of ***Presentation Composers*** can reason and decide on the best presentation to prepare. This will be sent from the composers as a group response to the selected ***User Assistant***.

During this process another, virtual, Implicit Organization is formed, this organization is composed by the client that originated the request, the members of the Implicit Organization that provided the required partial services and the client that presented the information. When service is provided, the organization dissolves.

We are interested in three main broad classes of relationship among roles and agents that support them. **Redundancy:** multiple agents support a certain role in an equivalent way; that is, a request for a service from that role can be serviced by anyone of the agents. **Partitioning:** the services in the role are partitioned among the agents; that is, a service request can be fulfilled by at most one of the agents. Note that partitioning is often based not on the type of service being requested, but on its parameters; e.g., two agents may maintain user profiles with the same schema but for different users. **Coalition:** the services of the role can be provided by the group but there is no single agent that can do it by its own; thus, collaboration is necessary. As it has been illustrated earlier, the idea of Implicit Organizations provides a framework for all these relationships.

6. LoudVoice

The group communication infrastructure required by implicit organization, as modeled by the *Implicit Organization* Pattern, is implemented, by means of an experimental communication infrastructure called *LoudVoice*, which has been designed to support channeled multicast [2]. *LoudVoice* uses the fast but inherently unreliable IP multicast — which is not a major limitation in our domain, since the communication media in use are unreliable by their own nature. However, we had to deal with message losses and temporary network partitions by carefully crafting protocols and using time-based mechanisms to ensure consistency of mutual beliefs within organizations. The current implementation of the implicit organizations, formalized by means of the Join Intention Theory [3], exploits group communication and *Overhearing* (i.e., the ability to listen to messages addressed to others) to minimize internal communication.

Whenever a new agent joins an environment, a protocol is run to negotiate the coordination policy used to service requests by the organization the agent wants to be part of. The same happens when an agent leaves the organization. This allows the agents to propose whatever coordination they deem most appropriate to the current context and to their own capabilities. The negotiation protocol has been designed to be as efficient as possible, and allows an external entity (such as a network administrator or a management system) to influence its results. Four general-purpose policies, which exploit overhearing for their functioning, have been identified: *Plain Competition* (i.e., all agents try to service a request, and the first to finish “wins”), *Simple Collaboration* (as the previous one, but the first to finish merges results from all others before sending a response), *Multicast Contract Net* (an efficient variation of the well known contract-net protocol), and *Master-Slave* (required for situations where an external controller has to be elected to choose on behalf of the organization). Furthermore, it is possible to design application-specific policies.

A client interacts with an implicit organization by sending its request to the *role* of the organization on the appropriate application channel; one of the members will respond when the request has been serviced. Policy negotiation and internal coordination are performed on a dedicated *control* channel.

7. PEACH Planned Architecture

The museum visitor's guide system, currently being developed within the PEACH project can be modeled using the newly introduced agents' patterns and later on designed and implemented using the LoudVoice infrastructure. Figure 5, presents the result of the late requirements analysis of PEACH actors, following TROPOS [1]. The visitor has several soft goals that the museum visitor's guide system needs to satisfy.

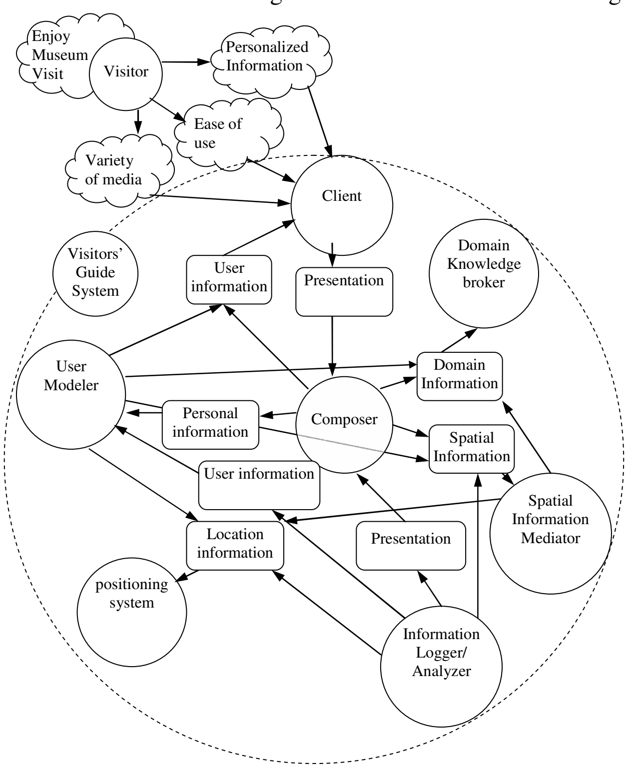


Fig. 5 PEACH actors diagram

According to the museum scenario briefly described earlier, each actor represents an Implicit Organization of agents; each of them is modeled by the *Implicit Organization* pattern. The following implicit organizations are identified: ***Positioning***– this implicit organization is responsible for reporting user current position periodically and/or by request. This organization can be composed of a WiFi based positioning system that continuously monitors user's movement, it may also include Infra-Red (IR) based positioning system – limited to sensing the user in the vicinity of an IR sensor and it can also include a positioning system based on dialogue with the user.

Spatial information mediator provides positioning relations between visitors and museum exhibits. For sake of simplicity, it's functionality is included in the positioning system for the rest of this presentation.

User modelers that may include different agents implementing user modeling approaches, such as the collaborative approach, a stereotypic approach and an adaptable, personal user model. ***Presentation* composers** may include agents that provide audio, text, slides or videos for presentation. ***Information brokers*** that provide the information required for presentation composition – pictures, text and other media, stored locally or remotely. ***Presentation clients*** - different clients with different presentation capabilities that may present information to users in the museum environment and ***Information logger/analyzer*** is a set of development and research support tools (will be ignored later for sake of simplifying the presentation).

In terms of the policies currently implemented with LoudVoice (see the previous section), the organization of User Modelers may adopt either a Simple Collaboration (to merge results from multiple modelers) or a Contract-Net, with bids as a function of the freshness of information available to the agents. ***Presentation Composer*** will adopt a competition policy – either Contract-Net, with bids based on quality of output given the devices available, or Plain Competition to enable quick reaction to user movements, especially when the museum is crowded. ***Information brokers***, by contrast, should adopt a collaborative policy such as Simple Collaboration, in order to return the greatest possible amount of information to queries. In the first version, in which we do not foresee support for group visits, (such as the same presentation being shown to a group of people simultaneously) ***Presentation Clients*** will also adopt a competition policy.

8. PEACH Dynamics and Design for LoudVoice

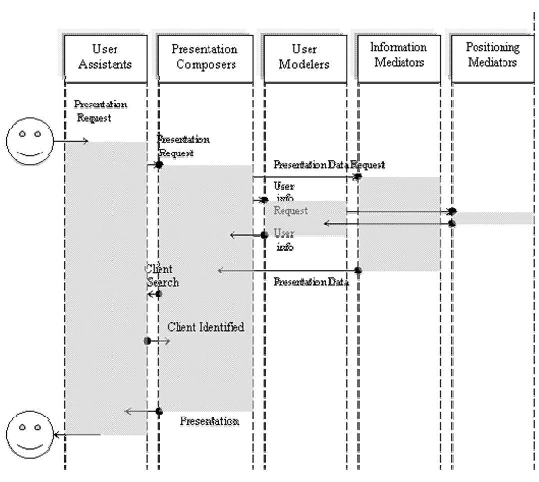


Fig. 6 PEACH implicit organizations dynamics example

The system architecture can now be defined in terms of *LoudVoice* infrastructure [2,3,4]. This representation can lead now to implementation of the various agents in the *LoudVoice* infrastructure. The museum visitor's guide system needs one "application channel" where all agents will be listening in order to monitor for service requests and provide services. Implicit Organizations' members negotiations will take place on separate control channels, one for every organization. Hence the *Overhearing* that is implemented as monitoring to channel multicast transmission over the application channel will enable all agents to gather information and learn about requests for services. Negotiations between organization members will take place on the organization dedicated control channel while the selected agent will, in turn, provides the required service over the application channel.

Figure 6 presents the sequence diagram with the various Implicit Organizations identified in PEACH (in the same notation used in Figure 3), for the same scenario of presentation generation. Every implicit organization has a dedicated control channel, while the application channel is represented by the arrows in the figure – representing messages transferred between members of the different organizations.

9. Conclusions

This work demonstrates how the complex requirements of a highly distributed system, situated in an active and highly dynamic environment, are modeled by the novel *Implicit Organization* agents pattern. It also shows that the idea of implicit organization enables the definition of a system architecture that in a later stage are implemented in the *LoudVoice* agents communication infrastructure. An Implicit Organization is a representative of a more general pattern that we called *Service Oriented Organization*. While the latter captures a large class of architectures that are emerging in service-oriented computing, Implicit Organizations apply to ambient intelligence or other scenarios in which a relatively limited number of agents have to react in real-time to environmental stimuli; other patterns are being applied to Service Oriented Organizations in environments such as the Business Grid (see, for instance, the concept of Service Domain in [16]).

The use of the novel agents' pattern of Implicit Organization provided the required means for system modeling for the Early and Late requirements analysis of the PEACH museum visitors guide system. Moreover, it provided the missing link that allowed continuing from the analysis to the design and implementation of the system.

Future work within the framework of PEACH project will demonstrate this complete cycle from requirements analysis to system implementation.

References

1. P. Bresciani, P. Giorgini, F. Giunchiglia, J. Mylopoulos, and A. Perini. TROPOS: An agent-oriented software development methodology. Journal of Autonomous Agents and Multi-Agent Systems (JAAMAS), Kluwer. To appear, 2004.
2. P. Busetta, A. Doná, and M. Nori. Channeled multicast for group communications. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems*, pages 1280–1287. ACM Press, 2002.
3. Paolo Busetta, Mattia Merzi, Silvia Rossi, Francois Legras, *Intra-Role Coordination Using Group Communication: A Preliminary Report* - International Workshop on Agent Communication Languages and Conversation Policies,

ACL2003 (in conjunction with AAMAS03), Melbourne, Australia, July 2003. Springer, LNAI 2922, pages 231-253
4. Paolo Busetta, Mattia Merzi, *An approach to the integration of peer-to-peer systems with active environments* - Workshop "Dagli Oggetti agli Agenti" (WOA'03), Villasimius (Cagliari), Italy, September 2003
5. J. Castro, M. Kolp, and J. Mylopoulos. Towards requirements-driven information systems engineering: The TROPOS project. Information Systems (27), pages 365–389, Elsevier, Amsterdam, The Netherlands, 2002.
6. K. Cheverst, N. Davies, K. Mitchell, A. Friday, and C. Efstratiou. Developing a context-aware electronic tourist guide: Some issues and experiences. In *Proceedings of CHI 2000*, Amsterdam, 2000.
7. K. Ducatel, M. Bogdanowicz, F. Scapolo, J. Leijten, and J-C. Burgelman. Scenarios for ambient intelligence in 2010. Technical report, Information Society Technologies Programme of the European Union Commission (IST), February 2001. http: //www.cordis.lu/ist/.
8. R. E. Grinter, P. M. Aoki, A. Hurst, M. H. Szymanski, J. D. Thornton, and A. Woodruff. Revisiting the visit: Understanding how technology can shape the museum visit. In *Proceedings of ACM Conf. on Computer Supported Cooperative Work*, New Orleans, LA, 2002.
9. M. Klusch and K. Sycara. Brokering and matchmaking for coordination of agent societies: A survey. In A. Omicini, F. Zambonelli, M. Klusch, and R. Tolksdorf, editors, Coordination of Internet Agents: Models, Technologies, and Applications, pages 197–224. Springer-Verlag, Mar. 2001.
10. Joseph McCarthy. Active environments: Sensing and responding to groups of people. *Personal and Ubiquitous Computing*, 5(1), 2001. Available at http: //www.inf.ethz.ch/vs/events/dag2001/.
11. M. Kolp, P. Giorgini, and J. Mylopoulos. A goal-based organizational perspective on multi-agents architectures. In *Proceedings of the Eighth InternationalWorkshop on Agent Theories, architectures, and languages (ATAL-2001)*, 2001.
12. E. Not, D. Petrelli, O. Stock, C. Strapparava, and M. Zancanaro. The environment as a medium: Location-aware generation for cultural visitors. In *Proceedings of the workshop on Coherence in Generated Multimedia at INLG'2000*, Mitze Ramon, Israel, 2000.
13. L. Penserini, L. Liu, J. Mylopoulos, M. Panti, and L. Spalazzi. Cooperation strategies for agent-based p2p systems. WIAS: Web Intelligence and Agent Systems: An International Journal, IOS Press, 1(1):3–21, 2003.
14. L. Penserini, P. Bresciani, P. Busetta, T. Kuflik. Architectural Patterns for Ambient Intelligence based on Service Oriented Organizations. Technical Report 16.01.04, ITC-IRST, Povo – Trento, 2004.
15. M. Zancanaro, O. Stock, and I. Alfaro. Using cinematic techniques in a multimedia museum guide. In *Proceedings of Museums and the Web 2003*, Charlotte, NC, March 2003.
16. F. Leymann, K. Güntzel, The Business Grid: Providing Transactional Business Processes via Grid Services. In *Proceedings of the First International Conference on Service Oriented Computing (ICSOC 2003),* Trento, Italy, December 2003. Springer, LNCS2910, 256-270.

A Multi-Dimensional, Unified User Model for Cross-System Personalization

Claudia Niederée, Avaré Stewart, Bhaskar Mehta and Matthias Hemmje

Fraunhofer IPSI
Dolivostr. 15, D-64293 Darmstadt
+(49) (0) 6151 869 831
{niederee, stewart, Mehta, hemmje}@ipsi.fhg.de

Abstract. Personalization is an established method for reducing information overload and facilitating targeted access to relevant information objects in an information system. However, the use of the information collected about a user and his current context in a user profile is traditionally restricted to a single system, although many user tasks require information handling activities that span several systems, like searching in different collections. The potential advantage of reusing profiles in other systems is increased by the advent of flexible service architectures for information systems like digital libraries that dynamically incorporate services with specific personalization characteristics. This paper discusses an approach in support of cross-system personalization. Building upon an ontology-based unified user context model, which describes the relevant dimensions of the user and his working context(s), this approach uses the metaphor of a context passport that accompanies users on their travel through the information space. When interacting with a system the relevant "context-of-use" is extracted from this context passport and is used for improved support of the respective information-related activity. The approach is discussed in more detail for the relationship dimension of the unified user context model.

1 Introduction

Personalization approaches rely on the fact that the user revisits the same system several times with similar requirements, since this is the prerequisite for collecting useful, user-specific data. An improved re-use and refinement of the information gathered in the user profile can be achieved when exploiting the same user profile in several systems. This is especially valuable when a user's interaction with an information system is part of a larger task that covers several such interactions possibly with different systems [16]. Examples of such information and knowledge intensive tasks are: writing a proposal, preparing a course or, leisurely planning a holiday. Such cross-system or cross-service personalization becomes even more important with the current trend to build upon architectures that are dynamic federations of services rather than a static integration (e.g. service architectures, Grid-based architectures and peer-to-peer architectures for digital libraries [1,17,18]. However, the reuse of user profiles requires an alignment of the personalization approaches. In more detail, the three basic challenges of personalization [28, 46] have to be reconsidered:

- How to model the user and his situation in an extensible and unified way that can be interpreted and used for personalization by different systems?
- How to collect data for filling and updating user profiles from user interactions with different systems based on the unified user context model?
- How to conceptually and technically enable different systems to make use of the collected user profiles for improved support of the user?

In this paper we present an extensible Unified User Context Model (UUCM) that can be used for modeling characteristics of the user and his situation, i.e. the user context [20], along different dimensions. In the UUCM several working contexts of a user can be modeled. The extensible set of UUCM facets, not only describes characteristics of the users themselves like interests and skills, but also aspects of the users' situation and environment. UUCM incorporates several dimensions such as Task, Relationship, and Cognitive Patterns dimensions. This supports flexibly modeling aspects of the user like the: tasks of a user and information objects related to the user. For easing the interpretation of user profiles based on the UUCM, we rely on Semantic Web Technologies for the representation of the user context model as well as the concrete user context profiles. The UUCM can not only be used to represent a user context model within a system, but also provides an intermediate format to exchange user profiles between legacy personalization systems. We illustrate our approach by discussing in more detail a use case in the context of community events that combines the relationship and task dimensions of the UUCM.

Our approach on cross-system personalization uses a context passport metaphor based on the UUCM. The user profile information is encapsulated in a context passport that accompanies the user when traveling through the information space. When performing an activity the relevant part of the context passport, what we refer to as the context-of-use, is selected and used to transform the activity in order to better support the needs of the user.

The rest of the paper is structured as follows: In Section 2 we discuss related work in the area of user and context modeling and its dimensions. Our approach to cross-system personalization is described in Sections 3, 4 and 5. Section 3 describes the Unified User Context Model (UUCM) together with an ontology for the UUCM facets and dimensions, whereas Section 4 presents a use case for the UUCM, combining the relationship and the task dimensions. Section 5 introduces an approach for cross-system personalization based on a passport metaphor. The paper concludes with a summary of the most important issues of the paper and some ideas for future work.

2 Related Work

Context can be thought of as the "extra", often implicit, information (i.e. associations, facts, assumptions), which makes it possible to fully understand an interaction, communication or knowledge representation [6]. Historically, context was either ignored or treated as a black box [33]. One effort to build context models for knowledge representation was carried out by Cyc [13], where the knowledge is structured into a large number of different contexts and rules are used to support the logical consistency of assertions for a given (closed-world) situation and importing assertions across different situations. A more general approach to context modeling can be found in [6], where the metaphor of a Box is introduced to systematically discuss different approaches to context modeling and contextual reasoning.

An important issue of context modeling is the question on which parameters or dimensions are to be taken into account for modeling context. Some researchers [30] consider context as an n-dimensional space, with each individual context as a region in this n-dimensional space. Lenat [30] identifies twelve dimensions that they consider sufficient to describe context. It should be noted however that a context may not be 100% specified by giving its coordinates in 12-dimensional space, but only approximated. In our work, the user has a task at hand and context here is likely to be more clearly defined as compared to the more general situation in knowledge representation. Dimensions like Culture, Sophistication, Granularity, Modality, Argument-Preference and Justification defined in Cyc are less relevant for user context modeling. Similarly, in our work, we consider a context model along a set of dimensions. We selected the dimensions Task, Cognitive Pattern, Relationship and Environment from the dimensions described in literature. Additionally, we define relevant subsets of the user context model as a user's working contexts distinguishing the different roles he plays. Context-of-use selectively considers only those facets of the user' context model which are relevant during a given information related activity. Our definition of context-of-use is in contrast to other researchers [21] who have defined the term more generally, covering the complete environment which characterizes the user's interaction with a computer system

Relation-based models of a user are information and community models that take into account the salient interrelationships of individuals in a cooperation or community context [35]. Having its roots in social theories, these systems use graph-based, or complex network structures to model interactions between human beings. Specifically, social network analysis (SNA) [64] extends and complements traditional social science by focusing on the causes and consequences of relations between people and among sets of people [15]. One approach to SNA is ego-centric network analysis. This approach focuses on an individual (or ego) and uses this individual's network of relations to understand the diverse factors contributing to his/her behavior and attitude [34, 41].

More general than social networks, relations as well relation types are considered of high significance in modeling users and information [58,60]. Relation types describe common properties for a class of relation and include, for example, containment relations such as: meronymic (or part-whole) and class inclusion [3,50,66,67] as well as non-containment relations such as thematic roles (or case relations). Having its roots in linguistics studies [4,32], the thematic roles relation types are significant in modeling a user's interaction with his environment because they represent a function, behavior, or assigned characterization that a participants plays in an association [54,58]. One type of participant from these classifications includes a determinant: an entity which is an active participant who initiates or determines the direction of process. Other types include immanent and recipient. In addition to ontology-based classifications, relational elements theories have been used to describe inherent properties of the relations themselves [9]. For example, a hierarchical property would be inherent for the relations *manages* and *contained-in*. Additionally, ongoing research in using relations include not only examining types for a single relation linking entities, but also types that apply to a relation built by either composition and transitivity of several relations or co-occurrences between relations [2,11,24,42]. The goal of such work is to discover patterns that would reveal a more complex relation in a network, which was derivable from the given ones.

From the domain of ego-centric social network analysis, we will be concerned with building relation based models of a single user and to this end consider an ego-centric approach. However studies have shown that persons, especially professionals, are tied into a manifold network of domain entity of different types [29]. Their connectedness within such a "web" is defined and affected by the work they do, the things and people they know, and the activities they engage in, etc. [42,65]. Such a socio-systemic context, necessitates a model more general than social theories. We, therefore focus not only on the relationships between people, but also relationships among resources within a given domain (of which people are just one type). In our research, a user's personal web is an important dimension of modelling a user.

Traditional Models of users are based on a mentalist paradigm [36, 48]. Mentalistic paradigms are based on characteristics of the user which we collectively refer to as cognitive patterns. These patterns represent user-specific aspects and include for example: interests, knowledge, preferences, misconceptions, or abilities [63].

Systems incorporating models of user's interest [16,26] have been widely used to selectively filter information on behalf of users from a large, possibly dynamic information source [5]. A common example of an interest based model is a collaborative filter which infers a user's interest and preferences from the ratings a user applies to an information item and from similarities between user's interests [27,45].

Despite studies which suggest that cognitive pattern models such as interest are insufficient data for accurate models of the user, it seems likely that these systems will continue to be adopted in the future [22]; therefore we consider, this traditional modelling dimension of the user.

Task models of user are considered important [25,57] based on the assumption that the goals of users (who participate in a task) can influence their information needs. When these needs are known in advance, a system can better adapt to its users [61] [62]. Based on these goal-driven theories for information related-activity, we consider the tasks an important dimension in modeling users and their context.

Environmental models are considered a key issue with respect to the interaction between human and computer because they describe the surrounding facts or assumptions which provide a meaningful interpretation (context) to a users' computer usage when their physical environment varies [53]. Furthermore, researchers have suggested that user's future usage scenarios will require more sophisticated support for changes that occur in a user's location and infrastructure. Such scenarios include: multi-computer usage: (e.g. a PC at work, a laptop on the go, and a PC at home); mobile computing: where a user carries a small information devices that can be temporarily connected to a network or ubiquitous information: where the information space can be accessed from information walls, kiosks, or desktops [16] and federated services: where collective information is dispersed among information sources.

Given the aforementioned trends and scenarios, environmental models are an important dimension in adequately supporting aspects of the users' situation and environment.

Generic user models are, in theory, systems which have, among other aspects, two major goals: 1) generality: which would allow a model of the user to be usable in a variety of application content domains; 2) expressiveness: in that the model is able to express a wide variety of assumptions about the user [26].

A number of factors contribute to the proposal in support of generic or unified user models. On the one hand, given the number of aforementioned dimension that is possible when modeling users, researchers have considered a generic approach to modeling users, because at present, there is no unified theory which systematically integrates all dimensions. On the other hand, current systems which typically model a user along a single dimension suffer from a limited view of users and a significant amount of potentially useful information about the user may be lost; thereby demanding a need for more robust models [7 ,55].

The trend towards dynamic federated services and multi-usage and ubiquitous environments, system designers can ease development and maintenance, increase reuse of user profiles and support the alignment of personalization approaches across systems by considering the use of generic user models in which a variety of different dimensions of the user are possible [26].

Finally, similar work in standards to build unified user-related models for dynamic information spaces in RDF [10] and the standardized RDF vocabulary CC/PP (Composite Capabilities/Preferences Profile, [8]). The work in CC/PP describes user preferences with respect to devices and user agents. We follow the standardization guidelines set forth in these areas.

3 Unified User Context Modeling (UUCM)

Adequate user modeling is the basis for every kind of personalization. Beyond properties of the users themselves, as they are traditionally captured in user models, information about the users' situation can be used for extended personalization support leading to user context models [20]. Cross-system personalization requires a unified approach to user context modeling that captures the relevant facts about users and their current situation in a way that can be shared by different systems. In order to deal with the variations in personalization approaches such a Unified User Context Model (UUCM) has to be flexible and extensible enough to capture the relevant dimensions of user and context modeling.

The UUCM not only is structured along different dimensions like tasks and cognitive pattern, but also captures the fact that the user interacts with systems in different working contexts by structuring the model accordingly.

3.1 Unified User Context Modeling

Two levels are distinguished in our approach for unified user context modeling. On the abstract level, the basic building blocks for the UUCM are defined: user context, user model facets, core properties for facet description, and user model dimensions. We use the term facet here to represent the different characteristics of the user not as it is traditionally used in knowledge representation to refer to the properties of a slot. This level defines a meta-model for the concrete dimensions and facets used in the description of the user context model. For the cross-system personalization approach, that we are aiming for, it is assumed that this user context "meta-model" is published as a shared ontology and all participating systems rely on this model.

On the concrete level, an extensible set of UUCM dimensions and facets is defined. This is not restricted to just users' interests, but also includes tasks and relations to other entities in the information space and respective user communities. UUCM facets and dimensions are described as part of an additional ontology that is shared by the components involved in the cross-system personalization. The UUCM meta-model, thus, can be combined with different UUCM facet and dimension ontologies to form concrete user context models that provide the schema for the construction of user context profiles.

Structure of the Unified User Context Model

The structure of the UUCM is summarized in Figure 1. A simple but flexible and extensible way of modeling the different facets of the user can be accomplished by the use of name/value pairs (cf. modeling of context by parameter value pairs in [6] Following this approach, a name/value pair is used to capture each facet of the user context model (e.g. user preferences) and new facets can be easily added.

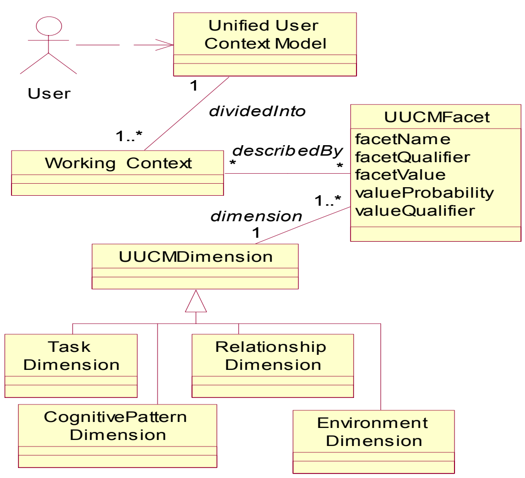


Figure 1: Structure of the UUCM

Since we want to use the UUCM in an open, cross-system personalization approach, qualification of names as well as values is a crucial aspect. Such a qualification binds facet names and values to vocabularies or ontologies. This eases interpretation of user profiles in a global context. In summary, each UUCM facet is described by the following properties:

facet name: Name of the UUCM facet to be described;

facet qualifier: A qualifier for the facet itself; this qualifier can be used to bind the facet to a defining vocabulary;

facet value: The values of the facet; The structure of the value may depend on the respective facet. In the general case, it is a value or a reference to another resource. (Our use of the term resource is comparable to that of RDF [49].)

value qualifier: A qualifier for the value(s) of the facet; this qualifier can be used to bind the value of the facet to a defining vocabulary (in contrast to the facet qualifier that qualifies the facet itself); In the case of a UUCM facet area-of-interest, for example, one might state that the ACM classification schema is used to specify the user's research interests.

value probability: A weight reflecting the probability of the facet value; this property of a facet can be used to express the reliability of facet values that are computed from analyzing user behavior.

facet dimension: Each facet is assigned to one of the dimensions covered by the UUCM; The UUCM facet area-of-interest, for example, is part of the UUCM cognitive pattern dimension.

Two aspects are used for structuring the UUCM. First, the UUCM is structured into several working contexts taking into account the different tasks and roles of a user (see Section 3.2). Secondly, the UUCM is structured along a set of UUCM dimensions that are discussed in more detail below. The UUCM structure can, thus, be summarized as follows (see figure REF): A user profile is divided into a set of working contexts, where each context is described by a set of UUCM facets. Each facet is assigned to one of the UUCM dimensions. We encode the UUCM as an RDF Schema augmented with OWL expressions.

User Context Dimensions and Facets

The UUCM just defines the principle way a user context profile is described and structured for cross-system personalization. For the description of concrete user context profiles the UUCM relies on ontologies (or vocabularies) for the UUCM dimensions, the UUCM facets and for the facet values.

In our approach, we use four UUCM dimensions (see Figure 1): The Cognitive Pattern dimension, the Task dimension, the Relationship dimension, and the Environment dimension. The selection of the dimensions is based on user models in existing personalization approaches and on context modeling approaches (see related work Section). For each of the UUCM dimensions, our UUCM facet ontology defines a set of UUCM facets, which describe the aspects of the respective UUCM dimension. These facets are presented together with the respective dimensions in the following. However, the UUCM is independent of the selected facet and dimension ontology. It can be combined with other sets of facets and dimensions. It is not the goal of this paper to give the ultimate set of facets and dimensions, but to discuss a flexible and extensible approach for unified user context modeling.

Within our facet ontology, the concrete facets are defined as subclasses of the general class UUCMFacet defined in the UUCM. They inherit all properties of the class UUCMFacet and define facet-specific restrictions like e.g. for the types of resources that are valid facet values. With this approach, there is a large flexibility with respect to which aspects are fixed for all instances of one facet (e.g. the facet name) and which can be selected individually for each facet instance (e.g. the value qualifier, if one wants to allow the use of values from different vocabularies). An alternative modeling approach is to make all facets instances of the general class UUCMFacet. This, however, gives fewer options for a systematic definition of specific types of facets.

The Cognitive Pattern Dimension

The cognitive pattern dimension describes cognitive characteristics of the user. It contains the facets that are traditionally used in personalization approaches. Based on an analysis of existing personalization approaches we selected the following facets to be included into our facet ontology:

- The facet *areas-of-interest* describing the interests of a user typically based on a controlled vocabulary or ontology of subjects (specified by the value qualifier).
- The facet *competence* with two facet subclasses *skill* and *expertise*.
- The facet *preference* that can be used to model preferences of the user.

Each of these facets may have several values. In this case the same facet is contained several times in the user context profile. Alternatively one may also enable the use of multi-value facets within the facet ontology. However, in this case all values have to share the same value qualifier.

The Task Dimension

When interacting with an information system, the user is involved in a task that determines his/her information needs and the goals of the performed activities. Tasks are described in (domain-specific) task models that structure tasks into subclass hierarchies. The user profile may refer to such task models.

We identified the following useful facets for the task dimension:

Current Task: this facet describes the task the user is currently involved in. This facet has a facet qualifier referring to a task model description, and the value qualifier refers to a concrete domain task model, whereas the facet value is a reference to a concrete task instance based on this task model. Using this approach, any appropriately described task model can be fitted into the UUCM.

Task Role: This facet describes the role of the modeled user in the current task. This facet has a value qualifier referring to an ontology of roles in the current task domain, and the facet value refers to a node in the chosen ontology.

Task History: This facet points to a history of tasks completed so far within the current working context. The task history helps to keep track of completed tasks and subtasks. This facet again is based on a task model (typically the same as the current task) and refers to a sequence of interrelated tasks.

Further task properties are currently considered for inclusion in the set of facets of the Task Dimensions. Since considerable work has been done in task modeling [39,52], the challenge here is not to identify adequate properties to describe tasks, but to decide, which of these properties are required as integral parts of the user context model.

The Relationship Dimension

The requirements and information needs of a user are also determined by the entities the user is related to. The facets of the relationship dimension are based on the relationships the user is involved in. Each facet represents one type of relationship. The facets of the Relationship dimension are, thus, based on one (or more) relationship type ontology (in our application example relationship ontologies from the scientific research community domain). The facet names are names of relationship types, the facet qualifier points to the respective relationship ontology and the facet value refers to the resource the user is related to via this relationship. The value probability, finally, gives a probability for the existence of this relationship. This dimension is discussed in more detail in Section 0 in the context of the UUCM use case.

The Environment Dimension

The environment dimension refers to those parameters which are typically used for context-awareness approaches. Facets like current time, location, device, language etc are parameters which influence and, thus, are important in understanding the interaction between the user and the computer. These aspects are also important in understanding the user's changing requirements in different scenarios. These facets include:
Time: Every working context would be valid in a certain time frame;
Location: This facet refers to the physical location of user;
Device: The device the user is using, e.g. PC, PDA, etc.
Language: The language of choice for the user;

These are only the most central facets of this dimension. Many other facets describing the environment might be important depending upon the specific application. However, the environment dimension is not in the focus of our work. We will rely on existing and upcoming work in this area. We included this dimension to show that it can be part of the UUCM in the same unified way as the other, more traditional user modeling dimensions.

3.2 User Context Modeling, Working Context, and Context-of-Use

In principle the user context can be described by a large set of facets. However, the user interacts with systems in different roles and is involved in different tasks in parallel, each of which is associated with a specific subset of the user context facets (e.g. private vs. job-related activities, see also [42]). To reflect this structuring the user context is divided into multiple working contexts (see Figure 1) grouping together user context facets that are related to and relevant for the same task and/or role of the user.

While accessing an information system and performing an activity to complete a task, a part of the current working context is extracted based on the relevance of the working context's facets for the planned activity (or activities). This subset of the working context is called the context-of-use. Section 5 describes the selection and the use of the context-of-use in more detail.

4 Personal Web Context: A UUCM use case

In this Section, we discuss the Personal Web Context (PWC) as a use case of the UUCM. The Personal Web Context combines the relationship and the task dimensions of the UUCM and outlines an approach for collecting information for the relationship dimension of the UUCM.

4.1 The Personal Web Context

The PWC is a model of a user, similar in structure to an ego-centric network (see Section 0), that describes the salient inter-relationships between the user and a set of interacting units that exist within a user's information space. The PWC represents the Relationship Dimension of the UUCM (see Section 3.1). The goal of the PWC approach is to analyze known relationships in order to infer and explicate implicit relationships between resources. This results in new "neighborhoods" that can be exploited for various kinds of services in the context of community events such as:

- Determining whom one should meet at a conference or trade fair in order to expand one's network of people [59];
- Recommending either relevant documents, or other people with whom to collaborate [23,38];
- Supporting a personalized conceptualization of an environment [40,51, 56];
- Supporting analytical activities that are crucial in business intelligence [31];

In the following, we discuss our approach to determine the PWC. We use the scientific research domain as an example application.

4.2 Information Model of the Personal Web

The underlying structure of the PWC is a domain ontology describing relevant entities in a domain. In our case, the domain ontology is built from the integration of existing ontologies describing the scientific community domain [37,44, 47]. Some examples of the domain resources in the ontology are: information objects (i.e. books, periodicals, and recorded presentations, technologies), people, organizations, and events. Typical relationship types in the domain are characterized by: a person being a member-of an organization; researchers who supervise graduate students; projects which produce a technology; a person author-ing a publication.

The domain ontology provides the basis for creating graph-based structures which we refer to as Resource Networks by defining the available resource and relationship types. Resource Networks are graph-based information models in which the nodes represent resources in the domain of various types and the directed edges indicate typed relationships between connected resources. The Resource Networks are used for several purposes. First, the Resource Network describes the relevant information space. Second, it is used as a starting point for expanding the PWC. Ideally, it uses the same or compatible relationship types as the UUCM for describing the Relationship.

Different approaches can be used to collect the data for a Resource Network from information spaces. On the one hand, metadata provided by the user like document authors, keywords etc. can be used to populate the Resource Network and, on the other hand, data for filling and updating the Resource Network can be collected by automatic analysis of information collections like DBLP [14].

4.3 Expanding the Personal Web Context

The goal of the PWC approach is to discover and expose the implicit relationships that link a user, by composition of relationships, to other resources that expand the PWC with new resources that were previously not in their PWC. The first step to extending the PWC is to extract pairs of relationships from the Resource Network which are connected by a common node. These pairs are called composition candidates (see Figure 2). After the extraction, we have a large set of composition candidates which can potentially contribute to expanding the Personal Web Context.

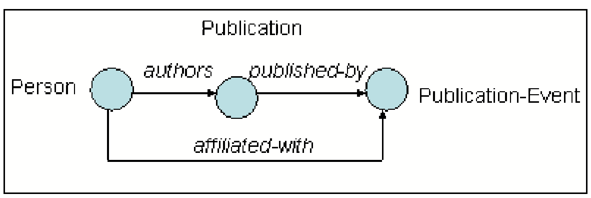


Figure 2: Composition Candidate

The next step in expanding the Personal Web is to consider a match-making process that supports discovering sequences of relations from which a single weighted relation can be compositionally inferred. For this purpose, composition rules are used. In the rules the relationships are represented as quadruples of the form (Subject, Predicate, Object, weight). Subject, Predicate and Object represent an RDF resource [49], Predicate represents the relationship types and weight represents the probability assigned to the Predicate linking the Subject to the Object. Figure 3 shows example rules which are discussed in more detail below.

In the match-making process, the rules are used: to decide which compositions candidates to use; to determine the type of composed relation and to compute the weight of the composed relation. A weight for a composed relation measures the likelihood that the composed relation holds. The measure for a composed relation is determined by: 1) the type of composed relations, and 2) the weights of the composing relations. The composed relationship acts as a surrogate for the sequence it replaces. If one of the resources involved in the composition is part of the PWC, the composition links portion of the user's existing PWC with entities in the information space. The approach we have taken to manage the set of composition candidates is to incorporate task-specific domain knowledge so we can exclude some of the composition candidates from the search space. Combining the task dimension provides us with further facts that increases the available contextual information and allows us to restrict the search process, for example by fixing the type of the terminus nodes, or only including nodes that are related based on a given subject matter.

The following example illustrates how the relationship and task dimension can be combined. Given a task: “Find potentials collaborators at this conference for a proposal in Grid computing”.

The additional facts that we get from including this type of task restrict the search process to finding: 1) the existence of a relation between the information seeker and a potential collaborator, 2) the strength of such a relation and, 3) a measure of the potential’s determinant role (see Section *0*) in the community. Additionally, since we are interested in proposal preparation, we focus only those relations in the network that exemplify a person as an active or influential community member for the subject Grid Computing. The additional context provided by the tasks, supports the use of example rules shown in Figure 3. For each rule in Figure 3 we make the following assumptions: all a_i are persons, all d_i are documents, all p_i are probabilities, and all e_i are events. Furthermore, we assume that an event, e_i, is a serial event devoted to a given subject matter.

Rule 1: attend-as-presenter

Assume: $n = d_1$.number-of-authors = number of authors for document d_1

$(a_1, \text{authors}, d_1, p_1) \land (d_2, \text{published-by}, e_1, p_2) \land (d_1 = d_2) \rightarrow (a_1, \text{attend-as-presenter}, e_1, p_3)$ where $p_3 = 1/n * (\text{MIN}(p_1, p_2))$

Rule 2: determinant-author

$(a_1, \text{publication-frequent}, e_1, p_1) \land (a_1, \text{publication-recent}, e_1, p_2) \rightarrow (a_1, \text{determinant-author}, e_1, p_3)$ where $p_3 = \text{MIN}(1, p_1 + p_2)$

Rule 3: meeting-presenter

$n = e_1$.number-of-participants = the number of participant in event e_1

$(a_1, \text{presenter}, e_1, p_1) \land (a_2, \text{presenter}, e_2, p_2) \land (e_1 = e_2) \land (a_1 \neq a_2) \rightarrow (a_1 \text{ meeting-presenter } a_2, p_3)$ where $p_3 = 10/n * \text{MIN}(p_1, p_2)$

Figure 3: Example Rules

In Rule 1, we assume that the number of authors for a document is known. Domain specific knowledge about publication activities allows us to assert with a certain probability, that one of the authors will attend the event to present the publication. This rule restricts the use of composition candidates to those that lead to persons who attend the event as a presenter.

In Rule 2, restricts the use relations that exemplify a person as an active or influential community member. One domain–specific criterion which can be used to determine influence is the frequency and recency of publications in an event which is devoted to a given subject. The relationships publication-frequent and publication-recent are inferred by other rules that use the author's relation and the connection between the document and the event (the published-by relation) as a starting point. Other criteria could include heading an organization or project.

Using Rule1 and Rule 2, we can discover and recommend the implicit relationships that link the information seeker (for whom the system knows will attend the event) to other prominent persons that have certain probability of attending the same event as a co-participant.

The use of the rules for the above examples can be sketched as follows: we insert a link of type attend into the PWC that links the user to the event, e_1, under consideration. As a next step, we use Rule 2 to determine the active community members for the event. In the next step, Rule 1 can be used to determine for each of the members, the probability of the active member attending the event in the case he has an accepted publication for the event. This results in a relationship attend-as-presenter with the respective probability. Finally, the relationship will be combined with the initially inserted relationship attend to a co-attend relationship between the user and the community member.

Rule 3, is representative of another kind of rule which allows us to estimate the probability of meeting a person at the event by chance. In this rule, we assume that the number of event participants is known and one meets about 10 people at a given event. These are example rules and we are currently developing further ones.

This example demonstrates that the use of relationship and the task dimensions for the UUCM and can be used to support a match between a systems users needs and the available items in an information space.

5 The Context-Passport Approach for Cross-System Personalization

This section introduces an approach for cross-system personalization based on a passport metaphor. The context passport is a compact representation of the user's current context model for cross system personalization. It also contains the activities chosen by the user to be performed in order to fulfill the tasks allotted. It contains ontologically-arranged information about the user's current tasks and related activities, his cognitive patterns (skills, area-of-interest etc), the environment (time, place, device used), and his personal web of the people and relationships involved, following the UUCM model for user context modeling (See Section 3).

In order to use the context passport for cross system personalization, the user takes the context passport and "presents" it to an information system (IS) that the user wants to use. Since the context passport is bound to a shared ontology, there is a chance the IS can partially interpret the context passport using a mediator architecture [19]. As a result of this partial interpretation, two flows of information are possible: one from the context passport to the IS, and secondly from the IS to the context passport. The first flow helps the IS to better understand what the user requires from the IS, since the context passport refers to the task model, activities and also other information about the user context model. The second flow arises due to the interaction between the IS and the user which changes the state of the user's context. The purpose of this flow is to update the context passport with the feedback from the interaction. The approach outlined here so far does not specify the details of second flow.

We describe the approach as follows: we start with the box metaphor for context modeling, and then we outline how to apply this metaphor for the context passport and the formalization for selecting activities and context-of-use. Finally, we describe the approach for using the context passport in a cross system personalization environment.

5.1 The Box Metaphor

In [21], the notion of context dependence is illustrated by introducing the metaphor of the box (see Figure 4).

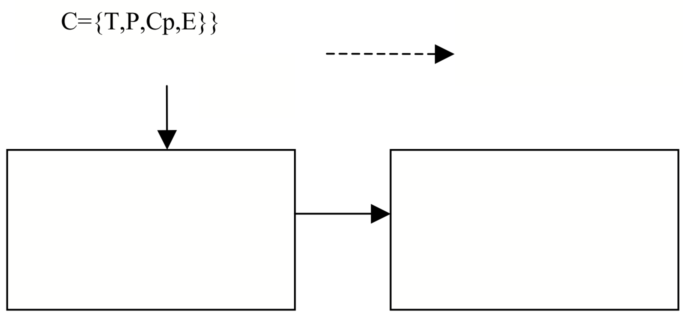
Sentence1
→ Sentence2

Figure 4: The Box Metaphor

A context dependent representation can be split into three parts: a) inside the box, a collection of linguistic expressions that describe a state of affairs or a domain; b) outside the box, a collection of parameters P1, P2… and finally, c) a value Vi for each parameter Pi. The intuition is that the content of what is inside the box is partially determined by the values of the parameters associated with that box. For example, in a context in which the date is 21^{st} April, 2003 and the location is Paris, any occurrence of 'today' will refer to 21^{st} April, 2003. Thus a statement like 'It will rain today' in this context is completed by the presence of date and location as a context parameter. The meaning of a context dependent representation is partly encoded in the parameters outside the box, and partly in the sentences inside the box. Reasoning mechanisms can be supplied for altering the balance between what is explicitly encoded inside the box and what is left implicit (i.e. encoded in the parameters). Intuitively, we can move information from the collection of parameters outside the box to the representation inside the box, and vice versa. These two operations are called push and pop to suggest a partial analogy with the operations of adding (pushing) and extracting (popping) elements from a stack. In one direction, push adds a contextual parameter to the collection outside the box and produces a flow of information from the inside to the outside of the box, that is part of what was explicitly encoded in the representation is encoded in some parameter. In the opposite direction, pop removes a contextual parameter from the collection outside the box and produces a flow of information from the outside to the inside, i.e. the information that was encoded in a parameter is now explicitly represented inside the box. This is a kind of decontextualization.

5.2 Using the Box Metaphor with the Context Passport

The context passport contains the facets along the four dimensions described in Section 3 which can each be formalized as a set, and facets for each dimension grouped in the relevant set as a key value pair. For the purposes of formalization, facet properties like facet qualifier, value probabilities etc are not taken into account. However these facet properties will be used in other steps of the approach.



We apply the box metaphor to the context passport as follows: instead of statements, the chosen activities are inside the box, the working context facets parameters are outside the box, along with their values. The UUCM dimensions and their facets determine the available parameters outside the box. Also, inside the box are the activities that a user might need to perform in order to fulfill allotted tasks. These activities are chosen from the task, or manually chosen by the user from a collection of activities. In the figure, the dimensions are represented by symbols T (Task dimension), P (Relationship dimension), E (Environment dimension) and Cp (Cognitive Pattern Dimension). Each of T, P, Cp and E are sets with key value pairs of their facets or attributes, along with their assigned values.

T={(t1,tv1),(t2,tv2), ...} t is a facet , tv is the related facet value of the Task Dimension,

P={(r1,re1),(r2,re2).....} , r is a facet, re is the related facet value of the Personal Web Dimension,

Cp={(Cp1,Cpv1),(Cp2,Cpv2).....} , Cp is a facet, Cpv is the related facet value of the Cognitive Patterns Dimension,

E={(e1,ev1),(e2,ev2).....} , e is an facet, ev is the related value of the Environment Dimension.

The interaction between the user and the IS using the context passport can be summed up as follows

1. The user goes to an information system and presents the context passport.
2. The IS can partially interpret the user’s requirements.
    a. Activities contained in the context passport which can be supported by the IS are selected.

b. The relevant context-of-use is extracted.

c. The activities are 'transformed' using the context-of-use.

These steps are described in detail in the following paragraphs.

3. The IS performs activities based on information derived form the context passport.
4. The feedback from the IS as a result of interaction is used to update the context passport and keep it up to date.

In a cross-system personalization environment, accessing a given information system may require the use of some part of the User's context. The context-of-use so required will depend on the activities that the user needs to perform. However, the given information system may not support all the activities that the user wishes to perform. Therefore, the applicable activities that can be supported have to be selected before.

Consider the Selection and Transformation functions as defined below:

Activity Selection function:

$S_1(C,IS) \rightarrow A$, $A=\{a_1, a_2 \ldots\}$

(IS = Information System, A = Activity list,

C = context passport)

Context Selection function:

$S_2(C, a_j) \rightarrow C'$(modified context)

Activity transformation function:

$F_a(S2\ (C, a_j)\ , a_j) \rightarrow a_j'$

After the activities supported by the current information system have been selected, the Context Selection function is used to identify which parameters from the context are relevant for this given activity. As introduced in the box metaphor, the selected parameters are then popped and put inside the box. Once the activity and the relevant parameters are known, the activity can be transformed and the transformed activity performed for the user. For example, if the activity is an information seeking activity, then popping the location from the user context can transform the information seeking into a geography based search (i.e. to find all hotels in the user's current city).

When the user now visits another information system, similar procedure is repeated. The activities not supported by previous system, but supported by this system can be selected by the Activity selection function. The context-of-use is then extracted from the context passport and the selected activities are transformed and performed. It is also possible that case this IS can support previously unfinished tasks or subtasks. The unfinished tasks are those which have been already performed on an information system, but the result was not satisfactory. In the detailed approach which is under development, the history of the unfinished activity will be used to trace the progress made so far while performing this activity.

What remains at this stage is the process of updating the context. While this is not clearly outlined at this moment, possible solutions involve developing a protocol for passing update information to the mediator which then updates the context passport. Most existing information systems would not be able to follow this protocol directly, so a wrapper-based architecture could be used to implement the update protocol. Newly developed systems supporting cross system personalization across a federation of systems or services, could implement this protocol directly and thus not require a wrapper. As standards like Web services become more popular and information systems start supporting them, the wrapper could combine the user interface and the context update feedback procedure. This might also make tracking user activities related to this system possible.

6 Conclusions and Future Work

In this paper we presented a Unified User Context Model (UUCM) that incorporates the various aspects that are relevant in capturing the characteristics of a user and his current situation in a flexible and extensible way. The UUCM provides a basis for the realization of cross-system and cross-service personalization approaches that enable the exchange and reuse of user profiles across system boundaries enabling improved support for multi-step information handling activities. The introduction of the UUCM, thus, is a contribution to the first question posed in the introduction of this paper. As a concrete use case for the UUCM we presented the Personal Web Context that combines the Task and the Relationship UUCM dimension. For the Personal Web Context we discussed ways to fill a relationship-based user context profile with data (collecting and updating user context profile data) and we also presented possible applications in the context of community events. Also, we will draw upon the inherent properties of the relations to determine probabilities that can be associated with each relation as well as the composed relations. The work presented in this part of the paper, thus, is a way to deal with the challenge addressed by the second question posed in the introduction of the paper.

Of course, the Unified User Context Model is only the first step on the way to cross-system personalization. As first ideas for the next steps in this direction and as a contribution to the third question raised in the introduction, we, therefore, presented an approach based on the context passport metaphor. The context-passport includes user and context information based on the UUCM. The relevant task-dependent context-of-use is selected from the context passport and applied for activity selection and transformation for better meeting the information needs of the user. The presented approaches are work in progress that still requires further investigation in several areas. Some important issues for further research and development activities in cross-system personalization support are.

- Development of further methods for the systematic collection data for the different dimensions and facets of user context profiles based on the UUCM; This work will take into account existing methods for collecting user profile data;
- Development and evaluation of further rules for the composition of relationships for the Personal Web Context approach based on an analysis of relevant domain knowledge
- For the context passport approach rules for the selection and transformation of information access activities based on the current context-of-use have to be developed.
- Investigation in methods for ensuring the adequate evolution of user context profiles; this includes methods the collection of new profile data in a cross system personalization context, but also the definition of facet-specific aging proc-

esses for the profile content. These activities will be based on existing work dealing with information decay [43].

- Evaluation of adequate visualization methods for the Personal Web Context and its interaction with the Resource Network of the examined information space.

Furthermore, we are working on the further refinement and a prototypical implementation of the context passport architecture.

Acknowledgments

This research is funded in part by the DELOS: a Network of Excellence on Digital Libraries Proposal no. G038-507618 Sixth Framework entitled: Technology-enhanced Learning and Access to Cultural Heritage.

References

[1] Alexandria Digital Library Project. (ADEPT) *Annual Progress Report Of The Alexandria Digital Earth Prototype Project.* 30th June, 2002.

[2] Anyanwu K, Sheth A.: *The p Operator: Discovering and Ranking Associations on the Semantic Web*. SIGMOD Record 31(4): 42-47 2002.

[3] Artale, A., Franconi, E., Guarino, N., and Pazzi, L. 1996. *Part-Whole Relations in Object Centered Systems: An Overview*. Data and Knowledge Engineering, 20(3): 347-383. Elsevier.

[4] Baker, Collin F. and Ruppenhofer J.: *FrameNet vs Frames vs. Levin s Verb Classes*. In Proc. of the 28th Annual Meeting of the Berkeley Linguistics Society, February 15-18, 2002.

[5] Baudisch P., *Dynamic Information Filtering.* GMD Research Series 2001, No. 16, GMD Forschungszentrum Informationstechnik GmbH, Sankt Augustin.

[6] Benerecetti, M., Bouquet, P., and Ghidini, C. *Contextual Reasoning Distilled*. Journal of Theoretical and Experimental Artificial Intelligence, 12(3):279--305, July 2000.

[7] Callan J., Smeaton, A., et. al.: *Personalization and Recommender Systems in Digital Libraries.* DELOS-NSF Workshop on Personalization and Recommender Systems in Digital Libraries, June 2003. (http://www.ercim.org/publication/workshop_reports.html).

[8] CC/PP: Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies. (http://www.w3.org/TR/CCPP-struct-vocab/).

[9] Chaffin, R., and Herrmann, D.J., Relation Element Theory: A New Account of the Representation and Processing of Semantic Relations. Memory and Learning - The Ebbinghaus Centennial Conference, edited by Gorfein, D.S., and Hoffman, R.R., Lawrence Erlbaum Associates, 1987, pp. 221-245.

[10] Cingil, I., Dognac, A., Azgin, A.: A Broader Approach to Personalization. Communications of the ACM, 43:8:136-139, August 2000.

[11] Cohen, P.R., Loiselle, C.L.: Beyond ISA: Structures for Plausible Inference in Semantic Networks Proceedings AAAI88, St. Paul, MN, August, (1988).

[12] Crabtree, B.I., Soltysiak, S.J.: Identifying and Tracking Changing User Interests. International Journal on Digital Libraries, 2:1:38-53, (1998).

[13] Cyc.org. (http://www.cyc.com/cycdoc/course/what-is-a-context.html).

[14] DBLP Digital Bibliography & Library Project (http://dblp.uni-trier.de/).

[15] Egonet. (http://survey.bebr.ufl.edu/EgoNet/).

[16] Fink, J., User Modeling Servers - Requirements, Design, and Evaluation, Dissertation vorgelegt zum Erwerb des akademischen Grades Dr. rer. nat. Datum der mündlichen Prüfung: 15. July 2003.

[17] Foster I., Kesselman C., Nick J., Tuecke S.. Grid Services for Distributed System Integration. Computer, 35(6), 2002.

[18] FreeLib, http://p2pdl.cs.odu.edu/.

[19] Gerding G., Kuhn, W., A Functional Approach to a Wrapper- Mediator Architecture, (http://musil.uni-muenster.de/documents/FuncMediation.pdf).

[20] Goker A. and Myrhaug H.I. User Context and Personalization. European Conference on Case Based Reasoning, Workshop on Case Based Reasoning and Personalization, 4-7 September 2002.

[21] Giunchiglia, F. & Bouquet, P. . Introduction to contextual reasoning. An Artificial Intelligence perspective, in B. Kokinov (ed.), Perspectives on Cognitive Science, Vol. 3, NBU Press, Sofia, pp. 138–159. Lecture Notes of a course on “Contextual Reasoning” of the European Summer School on Cognitive Science, Sofia, 1996.

[22] Hanani U., Shapira B., Shoval P., Information Filtering: Overview of Issues, Research and Systems. User Modeling and User-Adapted Interaction 11(3), 2001, 203-259.

[23] Huang, Z., Chung, W., Ong, T.H., Chen, H.: A Graph-based Recommender System for Digital Libraries. In Proc. of the second ACM/IEEE-CS joint conference on Digital libraries, pp 65-73, Oregon, USA 2002.

[24] Huhns, M.N., Stephens, L.M.: Plausible Inferencing Using Extended Composition. IJCAI Vol. 2, 1420 – 1425, 1989.

[25] Kaplan, C., Fenwick, J., Chen, J.: Adaptive Hypertext Navigation Based On User Goals and Context. User Modeling and User-Adapted Interaction 3. Kluwer Academic Publishers, The Netherlands, 1993, pp. 193-220.

[26] Kobsa, A., Generic User Modeling Systems, User Modeling and User-Adapted Interaction Journal 11: 49-63, 2001.

[27] Konstan, J.A., Miller, B.N., Maltz, D. Herlocker, J.L. Gordon, L.R., Reidl, J.: Applying Collaborative Filtering to Usenet News. Communications of the ACM, Vol. 40 No. 4, (1997) 77-87.

[28] Kramer J., Noronha S., and Vergo J.. A User-Centered Design Approach to Personalization. In ACM Computing Surveys, 08 2000.

[29] Kraut, H., Galegher, J., Edigo, C.: Intellectual Teamwork: Social and Technological Bases for Cooperative Work, Lawrence Erlbaum, Hillsdale, NJ (1990).

[30] Lenat D.B.. The dimensions of Context Space. (http/www.cyc.com/publications), 1998.

[31] Lesser, E.L., Storck, J., Communities of Practice and Organizational Performance, IBM Systems Journal “Knowledge Management”, Volume 40, No. 4, 2001.

[32] Levin, Beth. 1993. English Verb Class and Alternations: A Preliminary Investigation. Chicago: University of Chicago Press.

[33] McCarthy, J., and Hayes, P. J. Some Philosophical Problems from the Standpoint of Artificial Intelligence. Machine Intelligence 4, Edinburgh University Press. (1969).

[34] McCarty C. Structure in Personal Networks, Journal of Social Structure 3:1 (2002).

[35] McDonald D.W.: Recommending Collaboration with Social Networks: A Comparative Evaluation. In Proc. of the ACM 2003 Conference on Human Factors in Computing Systems (CHI'03), Vol. 5, No.1 (2003) 593-600.

[36] McTear M. F., User Modeling for Adaptive Computer Systems: A Survey. Artificial Intelligence Review, 7(3-4):157–184, August 1993.

[37] Middleton S., Alani, H., Shadbolt, N.R., Rource, D.C.: Exploiting Synergy between Ontologies and Recommender Systems. In Proc. of the Eleventh International World Wide Web Conference Semantic Web Workshop, Hawaii, USA, (2002).

[38] Middleton S., DeRoure, D., Shadbolt, N.: Capturing knowledge of User Preferences: Ontologies in Recommender Systems. In Proc. of the ACM K-CAP'Oi. 2001. Victoria, Canada: ACM Press (2001).

[39] Motta, E., Reusable Components for Knowledge Modeling. IOS Press, Amsterdam. November 1999.

[40] Nardi, B.A. et al. Integrating Communication and Information Through ContactMap. Communications of the ACM,45(4). (2002)205-242.

[41] Newman M. E. J., Ego-centered networks and the ripple effect. Social Networks 25, 83-95 (2003).

[42] Neuhold, E. J., Niederée C., Stewart A.: Personalization in Digital Libraries : An Extended View. ICADL 2003: 1-16.

[43] Niederée,C., Steffens, U., Schmidt J.W., Matthes, F.. Aging Links. In J. Borbinha, T. Baker, editors, Proceedings of the Fourth European Conference on Research and Advanced Technology for Digital Libraries (ECDL 2000), September 18 - 20, 2000, Lisbon, Portugal.

[44] Ontology of SCIENCE: http://protege.stanford.edu/ontologies/ontologyOfScience/ontology_of_science.htm.

[45] Pazzani M.: A framework for Collaborative, Content-based and Demographic Filtering. Artificial Intelligence Review, 13 Vol. 5-6, 393-408, (1999).

[46] Pednault, E. P. D., Representation is everything. ACM 43,8 (XXX-Month 2000), 80-83.

[47] Protégé Ontologies Library http://protege.stanford.edu/ontologies/ontologies.html

[48] Pohl W., LaboUr - machine learning for user modeling, In Proc. of the 7th International Conference on Human-Computer Interaction). Volume B, pages 27-30, Amsterdam, 1997. Elsevier Science.

[49] RDF/XML Syntax Specification (Revised) W3C Recommendation. (http://www.w3.org/RDF/).

[50] Rumbaugh, J. Relations as semantic constructs in an object-oriented language, Conference proceedings on Object-oriented programming systems, languages and applications, 1987, 466-481.

[51] Sack, W., Conversation Map: A Content-Based Usenet Newsgroup Browser. The 2000 International Conference on Intelligent User Interfaces(UIU 2000),ACM Press, 233-340.

[52] Schreiber, G., Akkermans, H., Anjewierden, A., De Hoog, R., Shadbolt, N., Van de Velde, W., Wielinga, B., Knowledge Engineering and Management - The Common-KADS Methodology. MIT Press, pp. 455, Cambridge, Massachusetts 2000.

[53] Schmidt, A., Beigl, M., Gellersen,. There is more to context than location. In Proc. of the International Workshop on Interactive Applications of Mobile Computing (IMC98), Rostock, Germany, 1998.

[54] Shapira, B., Shoval, P., Hanani, U., Stereotypes in Information Filtering Systems. Information Processing and Management, Volume 33, Issue 3, Pages: 273 – 287, (May 1997).

[55] Sharma, A., A Generic Architecture for User Modeling Systems and Adaptive Web Services. IJCAI 2001 Workshop on E-Business & the Intelligent Web, Seattle, USA, 2001.

[56] Smith M.A., et al., Conversation Trees and Threaded Chats. ACM 2000 Conference on Computer-Supported Cooperative Work (CSCW’00) (Philadelphia, PA, 2000) 97-105.

[57] Souchon, N., Limbourg, Q., Vanderdonckt, J., Task Modeling in Multiple Contexts of use. in Proceedings of 9th International Workshop on Design, Specification, and Verification of Interactive Systems DSV-IS'2002 (Rostock, 12-14 June, 2002), Lecture Notes in Computer Science, Vol. 2545, Springer-Verlag, Berlin, 2002.

[58] Sowa, J. F., Knowledge Representation: Logical, Philosophical, and Computational Foundations. Brooks Cole Publishing Co., Pacific Grove, CA, 2000.

[59] Tomobe, H , Matsuo, Y., Hasida, K., Ishizuka, M., Social Network Extraction of Conference Participants. WWW 2003 poster. (http://www2003.org/cdrom/papers/poster/p092/p92-tomobe.html).

[60] Trigg R., A Network-Based Approach to Text Handling for the Online Scientific Community , PhD Dissertation. (http://www.workpractice.com/trigg/thesis-chap4.html).

[61] Tyler, S.W. and Treu, S., An interface architecture to provide Adaptive Task-Specific Context for the user. International Journal of Man, Machine and Studies, 1989(30): p. 303-327.

[62] Vassileva, J., A Practical Architecture for User Modeling in a Hypermedia-Based Information System. In Proc. of the 4th International Conference on User Modeling, pp 115-120, UM'94, Cape Cod, MA, 14-19 August 1994.

[63] Wahlster, W. and Kobsa, A.., User models in dialog systems. User Models in Dialog Systems, pp 4–34. Springer, Berlin, Heidelberg, 1989.

[64] Wasserman, S., Galaskiewicz, J., Eds.: Advances in Social Network Analysis Sage. Thousand Oaks, California, (1994).

[65] Wenger, E., Snyder, W.: Communities of Practice: The Organizational Frontier. Harvard Business Review, January-February (2000).

[66] Wierzbicka, A., Apples are not a "kind of fruit": The semantics of human categorization. American Ethnologist, 313-28, (1984).

[67] Winston, M., Chaffin, R., and Herrmann, D.. A Taxonomy of Part-Whole Relations. Cognitive Science, 11: 417-444, 1987.

A Generic Architecture for User Interface Personalization

Jing-Hua Ye and John Herbert

Department of Computer Science
University College Cork
College Road
Cork, Republic of Ireland
jhy1@student.cs.ucc.ie j.herbert@cs.ucc.ie

Abstract. In this paper, we present research efforts aimed at the development of system support for automated user interface (UI) adaptation to different devices. We introduce a generic software architecture for the run-time adaptation process covering UI fragmentation, personalization and transcoding. The adaptation software is realized by a Java-based framework, which combines the advantages of manual and automated adaptation and supports the addition and removal of modular adaptation mechanisms. This framework is based on a generic UI markup language (XUL) allowing the manual addition of semantic meta-information at design time. This software system has been evaluated on simple case studies by doing various experiments, for example, cost of fragmentation and speed comparison of SAX and DOM transformation. Finally, we also demonstrate how link personalization, user-based personalization and context personalization are supported in a general modular fashion.

1 Introduction

The fastest growing community of web users is mobile visitors who browse the web with various (mobile) hybrid computing devices such as the wide spectrum of PDAs, smart phones, and tablet PCs available. About 95% of mobile phones sold today are "web-ready" and it has been predicted that within the coming two years there will be over one billion mobile devices with access to services and information through the web. The design of UIs has become one of the major challenges for the new hi-tech portable devices. Mobile devices lack the capabilities of their desktop counterparts on account of suffering from physical resource constraints (e.g. smaller screens, lower bandwidth networks etc). The design and the development of UIs for mobile computing will be very time consuming, error-prone or even doomed to failure despite the continously physical improvements on those mobile devices.

Personalizing applications or web sites attracts more and more attention in human-machine interaction. The UI on small wireless devices has only very limited capabilities, thus having the possibility of personalizing is vital for a user accessing applications or the web sites on these devices. Allowing a developer to concentrate on the functional aspects of an application without worrying about presentation on different devices should have significant benefit.

A main obstacle in supporting UIs on heterogeneous devices is the variety in display and memory size. Whereas a complete UI can be easily displayed on a desktop PC at once, the memory and display constraints of a handheld device require splitting the UI into smaller UI fragments.

Fragmentation must not be performed arbitrarily and also must not break the logical structure of the UI. In order to preserve this property during the fragmentation process, semantic information about the logical structure of the UI must be contained within the XUL i.e. every generic XUL interface element must have an additional "breakable" attribute to clarify whether it can be split or not. Göbel and Sven Buchholz have exploited the DOM model to tailor a dialog so that it is displayable on handheld devices [17]. Our contribution is that we have developed a new fragmentation method that fragments a sophisticated UI by using SAX APIs to index each abstract XUL interface component so that it can be displayed on the target devices. This method includes six phases: indexing phase, classifying phase, redundant breakable compounds removal phase, linking phase, number of fragments determination phase, and writing phase. Even though this fragmentation process provides UI developers with the convenience of UI development for multiple devices, there is still a cost involved in this process and those costs have been quantified as shown in the evaluation section.

This paper presents a software framework providing an innovative generic software architecture for a run-time adaptive UI development system.

2 Overview of Related Work

Some different approaches have been developed to support forms of UI adaptation. Proxies such as Web $Clipping^{TM}$ and Portal-to-go services that automatically convert HTML to WML content for potable devices [12, 7, 21, 13]. Most systems take XML-described interface content and transform it into different HTML or WML formats depending on the type of target device [12, 21]. Intelligent, adaptive and component-based UIs support user and task adaptation [5]. Furthermore, some recent new developments for multi-device UIs [21, 7, 6] utilize generic, device-independent UI descriptions.

Since the advent of the WWW and the emergence of the XML, a number of different markup languages have surfaced for creating UIs for different devices, for instance, UIML [3, 11, 2], XForms [20], AUIML [14], XIML [8], XUL and so on.

XUL[22, 10] is a presentation specification for creating lightweight, cross-platform, device-independent UIs. The layout and look-and-feel of XUL applications can be altered independently of the application definition and logic due to the fact that XUL provides a clear separation between the client application definition and programming logic, presentation and language specific widgets. The main limitation of using XUL is that it lacks a way of describing UIs not defined by distinct graphical objects (e.g. Voice, Speech) and is more limited in

Fig. 1. The adaptive UI development system general architecture

scope than UIML.The compelling reasons of choosing to use XUL for UI design instead of inventing a new kind of description language are:

Re-usability: XUL employs a powerful overlays mechanism to override small pieces of a XUL file without having to resupply the whole UI, and to reuse particular pieces of the UI.

Extensibility: New content for XUL widgets, additional event handlers to a XUL widget, and new interface properties and methods can be defined by applying XBL with XUL.

Portability: XUL can be ported easily as a GUI interface meta-library to other programming languages such as Java,C,and C++ and so on. For instance, the jXUL open source project will integrate XUL into the Java platform.

3 Architecture and Implementation

The overall architecture (cf fig.1) is conceptually split into three tiers: the first tier is client inputs, which contains the XUL, XML and the corresponding XSL files. Each file in the client inputs is fed into the adaptation software component as the second tier, which proceeds various adaptation processes. The outcome of the adaptation software will be a set of device-specific UI language files (device-specific outputs), which forms the last tier of the architecture.

We have implemented a prototype of our adaptation software architecture and this is presented in figure 2. It is implemented by using J2SE1.2.4 Java compiler and makes use of the Xalan-XSLT-Processor [1] by the Apache group as well as the JAXP (Java API for XML Processing) [18] package from Sun Microsystems.

3.1 Client Inputs

The clients can use a normal text editor to create the XUL, XML and the corresponding XSL files at design time. Each file is so-called specification. The following specification are required as the inputs of the adaptation software:

UI Specification: developers specify UIs using a high-level markup language (XUL) based on the logical structure of the UI. In this specification, developers need to explicitly specify each XUL element can be split of not by giving a yes/no value to the 'breakable' attribute of each element.

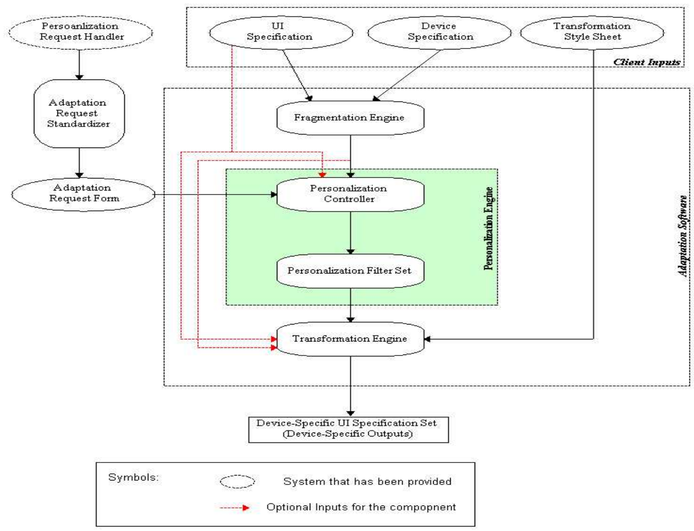


Fig. 2. Upgraded generic architecture for UI personalization

Device Specification: all constraints on the target device (i.e. memory capacity, screen size, resolution etc) on which the UI will be displayed are specified in standard XML format.

Transformation Style Sheet: all generic XUL interface elements in the UI specification can be transformed to a device-specific default format. There is a style-sheet for each type of device. Currently we have implemented default style sheets for HTML and XHTML for experimental purposes. The selection of a style sheet is based on the client's preferences. In short, it specifies the rules of transformation in terms of XSL.

3.2 Fragmentation Engine

The fragmentation engine (cf fig.2) manages the fragmentation process which paginates a single UI into a set of coherent UIs, each of which is capable of being displayed on the target device. The UI specification is forwarded to a cache, which manages the caching of UI fragments. The fragmentation decision-making machine compares the size of a UI that specified in the UI specification with the display size of the target device, which is specified in the device specification via SAX filters. If the size of the UI exceeds the display size of the target device, the fragmentation processor will be triggered to decide how many fragments need to be formed. Once the number of fragments is determined, each UI fragment can be formed by using split processor, which fetches the device properties from the fragmentation decision making machine and employs the SAX parser or SAX Writer to write the UI elements into the corresponding files in terms of

XUL element tags. Finally, store each written fragment into the cache until the completion of the fragmentation process.

3.3 Transformation Engine

The transformation engine (cf fig.2) takes responsibility for transforming either each fragmented page or the UI specification to a device-specific one. Currently, numerous transcoding techniques have been developed. General speaking, there are two standard approaches to transform an XUL UI into a platform-specific UI: the DOM transformation engines and SAX transformation engines. Our transformation engine encompasses both types of engines, with the DOM transformation engine being set as the default engine. Internally, This component comprises four possible different architectures:

1. Using a DOM transformation engine to transform each fragmented page. In this case, the following constituents are obtained:
   DOM Builder: It is responsible for building a DOM tree for each page in memory.
   DOM Navigation Processor It takes responsibility for adding navigation links as nodes on the DOM tree of each page.
   DOM Parallel Transformation Engine: Each page can be transformed to the device-specific format in parallel by DOM transformers.
2. Using a SAX transformation engine to transform each fragmented page. In this case, two subcomponents are supplied:
   SAX Navigation Processor: It takes responsibility for adding navigation links by means of using SAX Writer or SAX parser to write additional <link> elements to each page. Each <link> element contains a "src" attribute that points to the destination page.
   SAX Parallel Transformation Processor: Each page can be transformed to the device-specific format in parallel by SAX transformers.
3. Using a DOM transformation engine to transform the UI specification explicitly. In this scenario, the UI specification is parsed by the DOM parser, which constructs the corresponding DOM tree in memory. Subsequently, the result of the DOM parser will be transformed by the DOM transformer with supplied transformation style sheet to a device-specific UI specification.
4. Using a SAX transformation engine to transform the UI specification explicitly. The SAX transformation engine exploits a filter pipelined architecture. The UI specification and a reader are encapsulated as a Source object. The transformer sets up an internal object as the content handler for filter2, and tells it to parse the input source; filter2, in turns, sets up as the content handler for filter1, and tells it to parse the input source; filter1, in turns, tells the SAX parser to parse the input source. The SAX parser does so, generating SAX events which it passes to filter1; filter1, acting in its capacity as a content handler, processes the events and does its transformations. Then, acting in its capacity as a SAX reader (XMLReader), it sends SAX events to filter2; filter2 does the same, sending its events to the transformer;s content handler, which generates the output stream.

3.4 Personalization Engine

Personalization is an important factor for a successful UI. Kramer et al.[9] defined personalization:"Personalization is a toolbox of technologies and application features used in the design of an end-user experience. Features classified as 'personalization' are wide-ranging, from simple display of the end-user's name on a web page, to complex catalogue navigation and product customization based on a deep model of a user's needs and behaviors."

In general, the personalization of a UI can be divided into six categories:link personalization, user-based personalization, content personalization, context personalization, authorized personalization and humanized personalization. The link personalization, the user-based personalization and the context personalization are supported in this engine. The link personalization involves selecting links that are more relevant to the user and changing the original navigation space. The user-based personalization gives users the flexibility to tailor transcoding to the preferences of each individual user. Context personalization involves personalizing navigational contexts when the same information can be reached in different situations.

The personalization request handler (cf fig.2) acts as an administrator that monitors and detects the users' behaviors on the UIs and the contextual conditions of the UIs, e.g. the network performance. Furthermore,it gets direct feedbacks from users.

The personalization request handler sends requests in various formats to the personalization engine (cf fig.2). There is a necessity for standardizing those requests before the personalization engine acts on them. Thus the role of the adaptation request standardizer is to standardize the requests sent by the personalization recommender into the adaptation request form, which will be expressed in XML.

The personalization engine is a vital component for managing the personalization of adaptive UIs, and employs a controller-filter architecture. It comprises of two modules - the personalization controller and the personalization filter set. The personalization controller scans through the entire adaptation request form using a SAX parser, then informs the appropriate filter located in the personalization filter set, which is an interface, to execute a correct method.The personalization filter set contains four types of filters. The reduction filter reduces the resolution of each UI component in the UI to what the target device can support. The most frequent access UI components in the UIs, which are selected by the selection filter, will be displayed on the target device. All normal graphical UI components can be converted into different style that have same effect as the original components, e.g. convert all image files into a simple link, which point to somewhere else that has enough memory or display capabilities to display them; convert a standard menu bar into a simple list format etc, by using conversion filter.The shortcut link filter adds shortcut links to all pages or a particular page by exploiting MiniPath model [4],which is a combination of Naïve Bayesian mixture models and mixtures of Markov models [15].

Notations	
SAX-SAX	Using the SAX transformation engine to transform each fragmented page.
SAX-DOM	Using the DOM transformation engine to transform each fragmented page.
Single DOM	Using the DOM transformation engine to transform UI specification explicitly.
Single SAX	Using the SAX transformation engine to transform UI specification explicitly.

Table 1. Summary of notations

4 Evaluation

To illustrate some of the adaptation capabilities of our prototype we have developed a simple internal college book ordering online demo document. A small Mozilla application for this demo rendered on a PDA and a desktop computer (HTML) is shown in the appendix. A one-page PC presentation is paginated into a 6-page PDA presentation. The entry point to the dialog flow of the PDA is the main index page after transforming those fragmented pages into HTML. Navigation to the main index page is done by using the "main index page" link inserted by the framework for each fragment. A user can navigate between the fragments by means of "previous page" and "next page" links inserted by the adaptation system.

4.1 Evaluation of SAX and DOM Transformation Engine

In order to evaluate our architecture we have measured the performance for processing the college ordering online demo application. The notations will be used for evaluating the performance of this architecture in the table 1. The simulation of this simple case study is conducted under the following conditions:

- A 2.4GHz Pentium 4 processor DELL PC with 256MB DDR RAM running SUSE 8.0
- Using Xalan-XSLT-Processor to perform the transcoding process
- Compiled with J2SE1.4.2 compiler
- used JAXP1.2.4 package
- used Mozilla 1.6 web browser to illustrate the results of this simulation

The total processing time for each possible process is shown in the table 2. These results are the average of 1000 simulations. The above experimental results have shown that:

1. Even though SAX-DOM is faster than SAX-SAX by 0.0051 second, using a standard method of estimating statistical significance, the t-test [16] value(t = 0.511) indicates that the difference between them is not statistically significant. After fragmentation, users can choose either DOM transformation

process type	total processing time
SAX-SAX	0.6618 second
SAX-DOM	0.6567 second
Single DOM	0.1431 second
Single SAX	0.1165 second

Table 2. simulation results

or SAX transformation. SAX-DOM spent 88.58% time in the transformation over the total processing time. and SAX-SAX spent 88.82% time in the transformation over the total processing time. These figures have shown that SAX-SAX and SAX-DOM have spent most of the time on the transformation. Further optimisation of the transcoding will lead to a significant speed-up while preserving the advantages of flexibility.

2. Comparing transforming without fragmentation and transforming with fragmentation shows that there is a significant difference between them. This indicates that fragmentation slows down the entire process of adaptation. How to avoid fragmentation during the process of adapting the UI and how to shrink a UI for wireless devices efficiently is a topic that needs further exploration.

4.2 Evaluation of Fragmentation Engine

Method:

1. Design small elements: Each small element contains a various number of generic XUL interface components (i.e. XUL UI element tags), the number being in the range from 1 to 70. In this experiment, 26 small elements have been designed for the purpose of this experiment.
2. Design basic components: This experiment makes use of 26 small elements, some of them are randomly combined together to form a single basic component according to the needs of the number of fragments.
3. Fragment: The basic component is fragmented, and then each fragment is transformed by the DOM or SAX transformation engine. The number of fragments is increased in increments of 5 (i.e. 5 fragments, 10 fragments, 15 fragments and so on) until the computer stops the process automatically.
4. Establish a control group: The basic elements under identical experimental conditions (e.g. with the same increments in the number of fragments) are not fragmented, but are transformed with DOM and SAX transformation engines separately.

Results:

1. Figure 3 demonstrates that the processing time of pagination increase as the number of fragments grows. The processing time of fragmentation and the number of fragments have a direct proportional relationship.

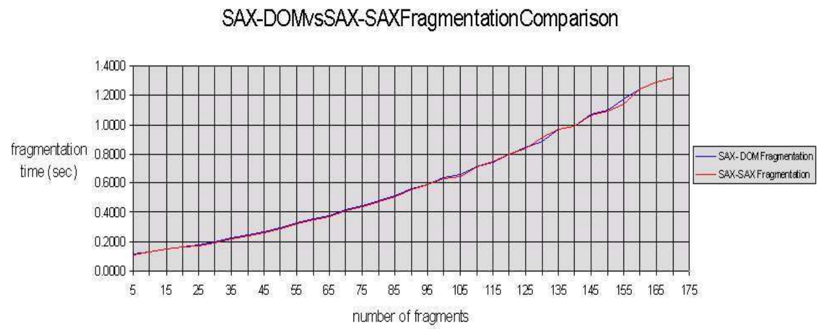


Fig. 3. Comparing the fragmentation time between SAX-DOM and SAX-SAX

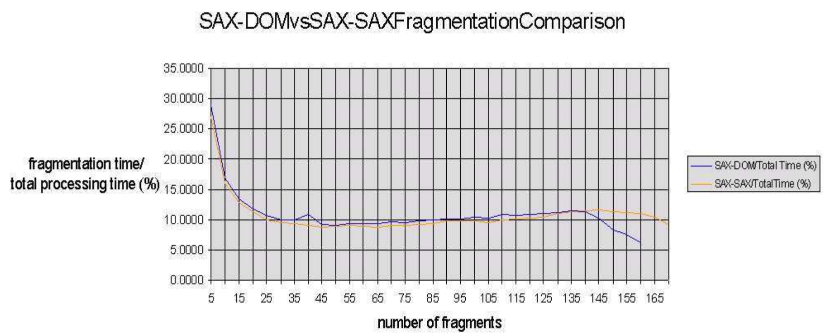


Fig. 4. Comparing the processing time of fragmentation over total processing time

2. Figure 4 shows that the time consumed on the fragmentation process is roughly 10% of the entire adaptation process time.
3. In this experiment, SAX-DOM (use the DOM transformation engine to transform each fragment) has successfully sliced a UI into 160 fragments and each fragment is transformed into a concrete UI. Likewise, SAX-SAX (use the SAX transformation engine to transform each fragment) has fragmented a UI into 170 fragments and each one is transformed into a concrete UI. By way of comparison, a UI containing 600 fragments (non-sliced) can still be transformed into a concrete UI for the desktop PC. This indicates that the adaptation process with fragmentation consumes 3.5 more times space than the one without fragmentation.

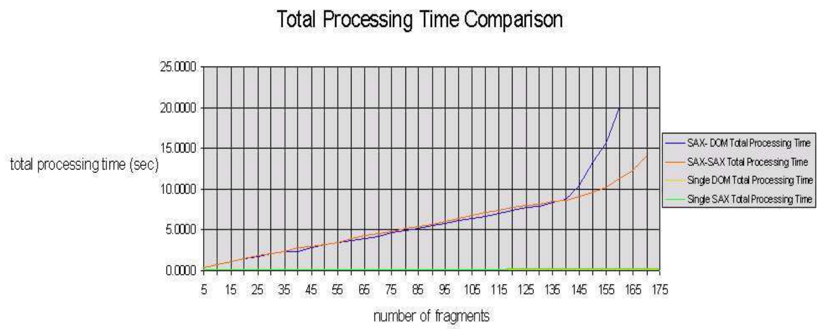


Fig. 5. Total processing time comparison

4. Figure 5 shows that the time spent on the adaptation process with fragmentation is much longer than the time spent on the adaptation process without fragmentation.

5 Conclusion and Future Works

This paper presents a Java-based generic architecture constructing thin-client UIs that adapt to multiple display devices. This architecture supports multiple-platform services access and UI tailoring.

The prototype implementation of this architecture will reduce memory consumption during the pagination process due to the nature of the SAX parser which means that it does not need to build the entire logical tree structure of the UI in memory in advance. Using DOM to fragment a very sophisticated UI, the size of whose tree exceeds the amount of free memory, will not succeed. This case is unlikely to occur in this framework. The evaluation of this framework has revealed that there is no significant speed difference between the SAX and DOM transformation engines after pagination. Inevitably, the speed of transforming a single page without fragmentation is much faster than with fragmentation.

Personalizing an application or a web site is becoming more and more essential, especially on wireless devices. The architecture has already successfully supported fundamental UI adaptation to multiple devices and UI tailoring.We demonstrate how the link personalization, the user-based personalization and the context personalization are supported in a general, modular fashion.

Ongoing developments include a "look-and-feel" consistency checking mechanism and "dynamic UI validation" mechanism into the adaptation framework. Integrating additional mechanism into the existing architecture can be done easily by introducing new engines due to the extensibility of this architecture. Furthermore, we will evaluate the prototype of this architecture with a more sophisticated case study.

References

1. The Apache Software Foundation: Xalan-Java version 2.6.0. (Feb 18th 2002) http://xml.apache.org/xalan-j/
2. Marc Abrams, Constantinos Phanouriou, Alan L. Batongbacal, Stephen Williams, Jonathan E. Shuster: UIML:An Appliance-Independent XML User Interface Language. In: WWW8, (May 1999), Toronto
3. Marc Abrams, Jim Helms: User Interface Markup Language (UIML) v3.0 Draft Specification. Harmonia Org, (2002) http://www.uiml.org/
4. Corin R.Anderson,Pedro Domingos,Daniel S.Weld: Adaptive Web Navigation for Wireless Devices. In: 17th International Joint Conference on AI (IJCAI-01) (2001) http://www.the4cs.com/ corin/research/pubs/ijcai01-talk.pdf
5. Grundy, J.C. and Hosking, J.G.: Developing Adaptable User Interfaces for Component-based Systems. In: 1st Australian User Interface Conference, (2000), Canberra, Australia Proc. IEEE P17
6. Grundy, J.C. and Zou,W.: An Architecture for building multi-device thin-client web user interfaces. In: 14th Conference on Advanced Information Systems Engineering Toronto, Canada (May 29-31 2002) P728–P732
7. Han, R., Perret, V., and Naghshineh, M.WebSplitter: A unified XML framework for multi-device collaborative web browsing. In: Proc. CSCW 2000 Philadelphia (Dec 2-6 2000)
8. Jacob Eisentein, Jean Vanderdonckt: Applying Model-Based Techniques to the Development of UIs for Mobile Computers. In: Proc. the 6th international conference on Intelligent user interfaces (2001)
9. Kramer Joseph, S. Noronha, and J. Vergo: A User-Centered Design Approach to Personalization. In: ACM. Num: 8 Vol: 43 CACM(Aug 2000) P45
10. Lori MacVittie: Conjure Up XUL for Browser Flexibility. Business Applications workshop (March 5 2001) http://www.networkcomputing.com/1205/1205ws4.html
11. M.Abrams, C.Phanouriou: UIML:An XML Language for Building Device-Independent User Interfaces. In: XML'99 Philadelphia (1999)
12. Marsic,I.: Adaptive Collaboration for Wired and Wireless Platforms. In: IEEE Internet Computing (July/August 2001) P26–P35
13. Pal Corp. Web Clipping services. (2001) http://www.palm.com
14. Pedro Azevedo,Roland Merrik: OVID to AUIML- User-Oriented Interface Modelling. In: IBM technical report (2000)
15. Ramesh R.Sarukkai: Link prediction and path analysis using Markov chains. In: 9th International WWW conference (2000)
16. TexaSoft Org: Statistics Tutorial: Comparing Two Groups In: tutorial, (2004) http://www.texasoft.com/2groups.html
17. Stedd Göbel,Sven Buchholz: Device Independent Representation of Web-based dialogs and contents. In: Proc. of the IEEE YUFORIC 01 (2001)
18. Sun, Inc.: The JavaTM Web Service Tutorial v1.2.4. In: (2003) http://java.sun.com/
19. Sun Microsystems, Inc.: XSLT-Compiler. In: (2002) http://www.sun.com/xml/developers/xsltc/
20. T.V.Raman: XForms:XML Powered Web Forms. publisher: Addison Wesley (September 2003)
21. Van der Donckt, J., Limbourg, Q., Florins, M., F., and Macq, B.: Synchronisesd model-based design of multiple user interfaces. In: HCI-IHM'2001 Workshop

on Multiple User Interfaces over the Internet, Lille, France (September 2001) 10–14
22. Vaughn Bullard,Kevin T.Smith: Essential XUL programming. publisher: John Wiley & Sons, Inc. (2001)

A Internal Book Ordering Online Demo

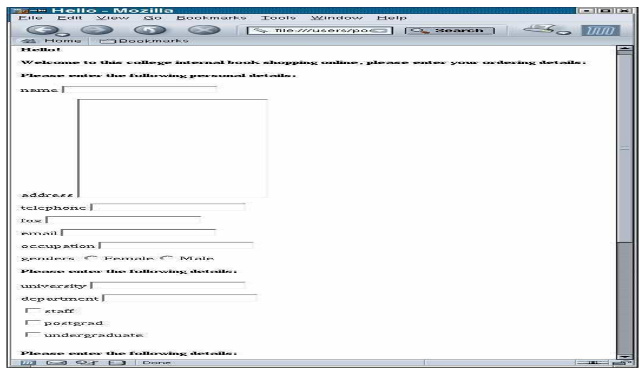


Fig. 6. Demo Application Rendered On a Desktop PC

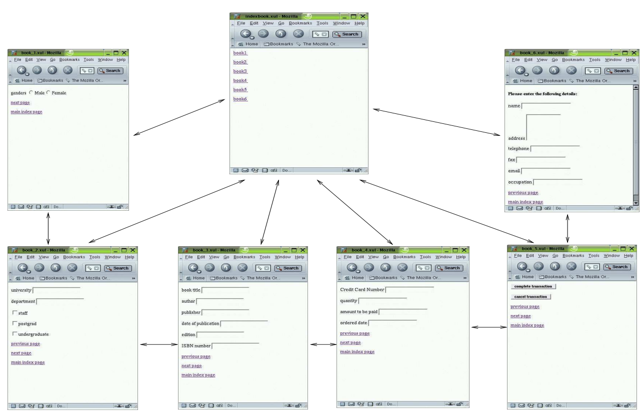


Fig. 7. Demo Application Rendered On a PDA

User Interface Design for Multi-platform Interactive Sports Content Broadcasting

Emmanuel Papaioannou[1], Erik Borälv[2], Athanasios Demiris[1], Niklas Johansson[2], Nikolaos Ioannidis[1]

[1]Intracom S.A., New Technologies Dep, 19002, Peania, Athens, Greece
{paem, dema, nioa}@intracom.gr
[2]Uppsala University Dept. of Information Technology, 751 05 Uppsala, Sweden,
{Erik.Boralv, Niklas.Johansson}@it.uu.se

Abstract. The new generation of television viewers is currently being confronted and becoming acquainted with a series of technological developments in the realm of consumer electronics and gaming that raise their expectations for similar advances in TV broadcasts. The MELISA platform aims at the cross-media broadcasting of sports events featuring interactive advertising and sports-related games over digital television and next generation mobile network infrastructures. The platform provides services for optimal presentation of complex interactive real time video content, for advertisement and an advanced real-time gaming (betting) engine in at least two different client platforms. User interface design is a major issue in a complex end-to-end solution having to cater the needs of users ranging from broadcasting professionals to end-users. Especially in the case of interactive gaming there are numerous challenges in the user interface design, in order to deliver to all categories of devices (and end users) equal quantity and quality of information. In this paper we present the overall system architecture and philosophy and then focus on user interface design issues both for the routine work of broadcasting professionals as well as the end users, owners of different types of consumer devices (such as PDAs and interactive TV Set Top Boxes).

1 Introduction

The MELISA platform aims to establish the infrastructure to support the virtual value chain for sports events broadcasting over wireless and digital television networks, by offering valuable, revenue building services, real-time video content, for advertisements and an advanced real-time gaming.

An end-to-end solution for the authoring and delivery of this new media content in real-time must provide the infrastructure necessary for the management, generation and broadcasting a multitude of services related to cross-media sports broadcasting.

MELISA introduces authoring tools for production, encoding and playback of rich interactive multimedia content in MPEG-4 for a variety of devices over wireless and digital television networks. The User Interface Design is based on basic understanding of human skill, cognitive aspects of human-computer interaction and on detailed knowledge about the specific needs and requirements of the end users in an interactive and time critical environment.

The Melisa clients provide the viewer with enhanced interactive content for real-time betting, advertising and in game information enhancing the user experience. User ergonomics and cognitive principles play an import role in the user interface design since the design itself is a significant acceptance factor for any new services on modern consumer devices.

2 Melisa end-to-end Broadcasting Platform

The Melisa platform is an end-to-end system that allows the creation, distribution and viewing of enhanced interactive video content in MPEG2 and MPEG4 format. The Server Platform provides a range of tools for collection and processing of XML information and video streams, from a variety of input sources.

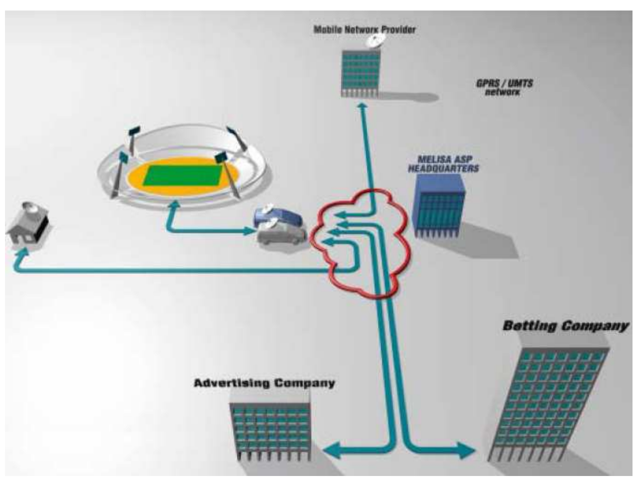


Fig. 1. MELISA Information flow and involved entities

The tools that comprise the MELISA system address multiple entities, as shown in Figure 1: a broadcasting organization, a wireless network operator, a betting company (mainly generating and updating the odds), an advertisement company and the MELISA ASP. The latter is the point of integration for all services and information, and is represented in the figure by a building representing their headquarters, as well as a dedicated van attached to the Outside-Broadcasting Van of the broadcasting organization. The role of the MELISA ASP may be played by any other involved entity in the overall scenario, e.g. the broadcaster themselves.

The service offers the viewers/mobile phone users, the ability to receive additional content for optimal, sports broadcast related, information visualization. This information leads on the one hand to more confidence in participating in betting services. Additionally it allows for the so called embedded advertisement, i.e. an opportunity to be informed in a non intrusive way about products in pre-selected categories. The challenge for MELISA is to achieve an easy to use and appealing way for the diversity of consumers using interactive television and next generation mobile phones, to equally enjoy the MELISA service.

2.1 System Components

The central entity of the Melisa system is the Melisa Application Server where all the information is collected and processed for transmitting. The Visual Enhancement and Capturing Unit captures the video content and provides enhanced content by real-time generation of visual enhancements that help the viewer quantify the context of the game, e.g. by overlaying graphics that denote the distance between two athletes.

A Template editor in the Melisa Application Server allows the creation of Visual Enhancement Templates prior to an event, thus reducing the processing needs during the live broadcast. The broadcast director is aware of the available pre-defined enhancements that can be instantiated during an event using the Work Flow Application. The Work Flow Control is the application that receives and controls all the available information. The Work Flow Control allows the viewing and managing of all content authoring activities during a broadcast investing only very limited time and effort.

The Lottery and the Advertising System are physically separated from the Melisa system. Both communicate with the Work Flow Control to provide the system with up to date betting and Advertising content that is updated in real-time. The betting and information is used to display real-time bet options to the viewer during a sports event. The viewer can then place a bet in real-time via a return channel to the Melisa Server. The Advertising content is used in conjunction with other visual enhancements such as player or car tracking. The car or the player becomes an active hotspot that the user can select and activate an advertisement possibly with links to e-commerce websites.

The Melisa Application server provides additional tools for scheduling Sports events, managing athlete and team information, as well as annotation tools for the generation of statistical information from scheduled event.

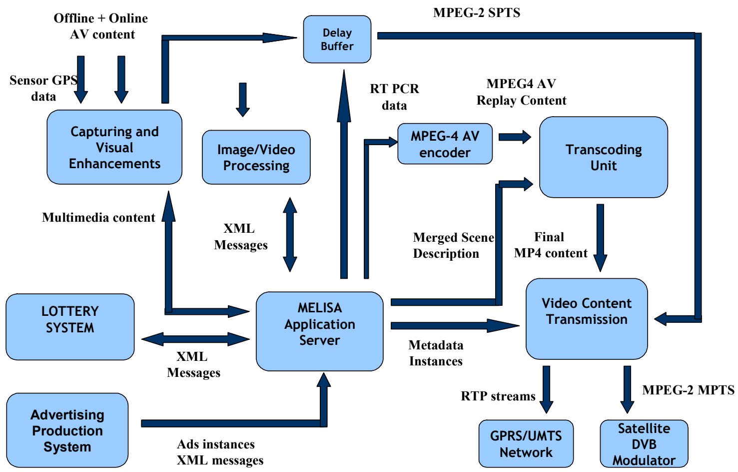


Fig. 2. Overall Melisa Server Design

The Application server processes the collected content and redirects it for Transcoding to MPEG-4 interactive content. The Video Content Transmission module generates the video streams for the Set Top Box and mobile phone clients. The derived MPEG-4 scene and accompanying MPEG-7 metadata are multiplexed with the original MPEG-2 Single Program Transport Stream (SPTS) and broadcasted via DVB Satellite Networks to the Set Top Box. In the case of the mobile phones, MPEG-4 RTP streams are generated and transmitted.

The Set Top Box (STB) client features DVBS reception, and MPEG-2 decoding capabilities. The received DVBS MPEG-2 stream carries the original MPEG-2 video stream, the MPEG-4 video that contains all the interactive elements and enhancements, such as the betting dialogs, object tracking information, advertising panels etc., as well as MPEG-7 metadata that contain the scene description information, event information for profiling, and additional statistics information concerning participating teams and players . The MPEG-4 video is overlaid on MPEG-2 video to provide the viewer with the desired interactivity. Every viewer that uses that STB has a locally stored user defined profile that utilizes the MPEG-7 metadata to perform content filtering thus personalizing the viewing experience. This personalization of the viewing experience is further extended from a single viewer level to a group viewer level with the notion of group viewing profiles, something extremely important for a multi-user medium such as television. As suggested by [9] advertising is ideal area for use of this personalization; dynamic insertion of adverts during, for example, ongoing football matches may both be technically feasible as well as economically sound. [10]

The PDA or Wireless client receives an MPEG-4 RTP stream that contains the video content and the interactive elements as well as the MPEG-7 metadata information as per the Set Top Box.

3 MELISA Platform User Interface Design

The requirements of a diverse system such as MELISA are difficult to meet with a single strategy. Many existing approaches include forming a set of design rules or templates, such as style guides or platform guidelines.[8]

The server-side tools display a set of varying profile, ranging from elementary administrative form-based input dialogues to real-time production. As expected, these tools address different needs. As in any project concerned with design and implementation of a novel system this raises a number of issues. In a novel system the context of use and users is not known or is difficult to make a precise description of. It makes the standardized way of testing usability troublesome.

It has been previously shown, in [11], that usability aspects alone are not sufficient for success when integrating technology with home life. With a larger view on interaction and functionality of technology, we also have to consider aspects of flexibility and adaptation to household activities.

There have been attempts to overcome the aforementioned problems. A number of solutions have been suggested, by considering the home aspects and social space of computing [13]. The theoretical frameworks for such studies always emphasize the structure of the social situation [12],[14]. Working in the well known area of sports broadcasting and betting enabled us to draw conclusions from existing knowledge. For example, betting companies know their user groups, or clients, in some detail. Thanks to this prior knowledge it is possible to form groups of users and define them as our specified users.

A second attempt was made by defining *personas*. Personas are often used instead of requirements, but in a quite different way. The persona is a fictional user, with a made-up life. Making personas is an additional activity. It doesn't prohibit compiling requirements or scenarios the traditional way. On the contrary, it needs this kind of input in order to form good and relevant personas [1].

The Personas method was first presented by Alan Cooper in 1999. A persona is "A precise description of our user and what he wishes to accomplish."[2] The idea with personas is not to define one generic user or to describe a list of tasks or duties. Instead it will give a short presentation of a couple of archetypical users, life-like characters driven by personal motives. The persona should describe the flow of someone's day, his or her skills, attitudes, environment and goals and answer questions like which pieces of information are required at what points in the day? And why are they using the product.

Cooper emphasizes that a persona is a tool for communication and design within the group of designers, software developers, managers, customers and other stakeholders. The aim is to give simple, but good enough description of the user to make it possible to design the system rather than a precise description or a complete theoretical model of a user.

Although the outcome of defining personas was partly unsuccessful, it was valuable as it served as a tool to examine the characteristics of the system and its use [16]. The value was perhaps only for the designers, and not the whole development group. Although we as a design team believe in personas as a helpful method of practice design, it is not just a matter of going ahead. The problems occurred because of the development group's lack of understanding of what personas are and how to use them, since only few project members have HCI knowledge. Other projects have experienced the same difficulties because the know-how of the method was not sufficiently integrated with existing knowledge and practice [15].

Fig. 3. The initial design of the annotation tool

3.1 Melisa Server User Interface Design

The server side application consists of preparatory tools and production tools used during live broadcast. It is vital that the production tools work efficiently since the most important aspect is the speed of use. The error rate has to be kept low since there are few barriers between the use of the tool and what finally is broadcasted [7]. The real-time aspect of the system implies that it is highly likely that any user errors will be broadcasted undetected.

The server side applications are to be treated as interfaces for experts [3]. Thereby it is accepted with a longer adaptation period. Ease of learning for the novice users is not regarded as important as long-term speed and effectiveness [17].

The design process therefore uses a number of iterations to find the desired level of performance. In Figure 3 we see the initial design of the annotation tool for football. It is used to record the events that are significant for football into a database.

In a number of redesign stages we are able to examine the properties and dynamics of the design. The result is a redesign of the interface without the design flaws. Most of the iterations use inspection techniques intended to find possible trouble areas [6].

We also use a predefined checklist that contains dos and don'ts. Some aspects that are treated in the checklist are:

1. Disposition of the screen area
2. Color codes
3. Icons and buttons
4. Placements

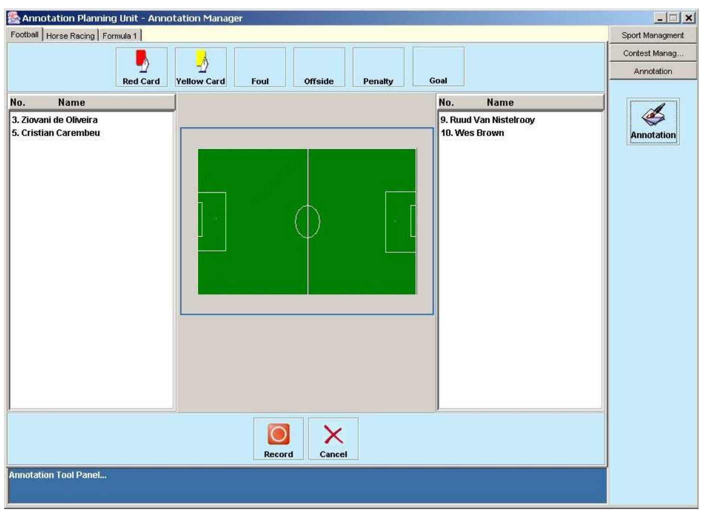


Fig. 4. The current design of the annotation tool, football view selected

The checklist is project internal, based upon experience in design [5]. The checklist is an effective tool for bringing development up to a certain standard. It cannot work as the only design aid, as checklists (even good ones [18]) contain general knowledge that needs interpretation. As Tullis has shown, design is not a question of common sense [19].

Using the combined results from the inspections and the checklist we are able to modify our design to perform better, and result in fewer operator errors. Figure 4 shows the latest design of the same annotation tool. It has been improved to perform faster, perhaps even using touch display technology for direct and fast interaction.

3.2 Melisa Set Top Box Client User Interface Design

The problems we face in regards to the Set-Top Box (STB) are associated with the expanding field of use of cable television. As cable television allows us to do more than just transmit video, we face the limitations imposed by the current STB technology. The intended STB use was to receive video signals and convert them to something we can view on our TV set. As the possibility to communicate with the sender to send back information exists, we are now faced with the need to include this functionality in the STB remote control. We also have an increasing need to store data into the STB – viewer profiles can be used to control what kind of advertisements we are shown. If our profile tells the STB we have a baby in the family we might get more information about family products compared to someone that has a viewer profile indicating a single family status.

Figure 5 shows a typical user interface design of the STB. As we can see, this design has a number of problems: (i) the design is PC-like in the sense that it is best used using the interaction tools available on such a platform and (ii) low density of information due to low screen resolution.

A STB is always controlled by a remote control. The latest types of remote controls include some kind of pointer to allow for improved interaction, but the functionality is still in all aspects very limited. It is very hard to provide a satisfactory solution even for simple tasks such as entering text.

Being fazed with interactive tools with such limitations one has to focus on redesigning the problem areas. Many techniques are known from the desktop domain:

- We can let the user choose from predefined values instead of having to enter values.
- We can use focus groups to navigate the user interface instead of the freeform navigation, as used by the mouse.
- We need to put more effort in providing better default values. A more intelligent way of calculating the default values may bring many benefits.
- We can provide ways to re-use information already present or entered. A solution like copy & paste is easy to use using a remote control and may facilitate many operations.

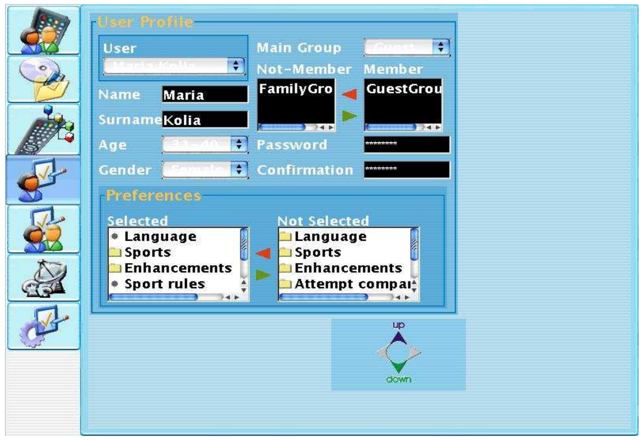


Fig. 5. The initial MELISA STB interface design

The navigation and operation of the STB is being tested using a prototype. The prototype is built using Flash as this allows integration of video, sound and interaction.

Building the prototype made us aware of the difficulties to design a User Interface for only remote control interaction. Because the content of a broadcast is not just static background information, we must ensure that all dialogue paths are short and possible to cancel if needed.

4 Conclusions

When developing computer systems for skilled professionals as well as for non-professional users, the role of the user interface is essential. Inadequately designed interfaces will result in low user acceptance, high costs, inefficient work, and possibly negative stress (cognitive overload). Especially in the cases of consumer devices, such as interactive television, correct designs of the user interface are a challenge determining almost exclusively the success or failure of a new service.

The Melisa Platform gives us the opportunity to explore several aspects of User Interface Design. Dealing with User interface design for such a diverse type of applications ranging from administration and data management tools to end user interactive video playback applications, gave us the opportunity to realize that we may have to change the way we think of interaction especially when the interaction tools have very limited capabilities. A way to overcome the limitations of the platform's interaction capabilities is to rely more on the locally stored user and group profiles.

References

1. Stefan Blomkvist. The User as a Personality - Using Personas as a Tool for Design. Working paper, Uppsala University (2002)
2. Alan Cooper. The inmates are running the asylum, Macmillan Publishing Company Inc. (1999)
3. Jakob Nielsen. Novice vs. Expert Users. Jakob Nielsen's Alertbox, (2000), http://www.useit.com/alertbox/20000206.html
4. Gulliksen, J., Göransson, B., Boivie, I., Blomkvist, S., Persson, J. and Cajander, Å. Key principles for user-centred systems design. In Behaviour and Information Technology, 22, 6, (2003) 397-410
5. E. Nygren. Graphical user interfaces, some hints. (available in swedish only). Report no. 60, CMD, Uppsala University (1995)
6. Nielsen J, Mack R.L Usability Inspection Methods. John Wiley & Sons. (1994)
7. Reason JT. Generic Error-Modelling System (GEMS). A cognitive framework for locating common human error forms. In: J. Rasmussen, K. Duncan & J. Leplat eds. New Technology and Human Error. Chichester & Wiley (1987)
8. Olsson E, Göransson B, Borälv E, Sandblad B. Domain Specific Style Guides - Design and Implementation, Proc of the Motif & COSE International User Conference, Washington D.C. (1994) 133-139
9. Konstantinos Chorianopoulos and George Lekakos and Diomidis Spinellis, Intelligent user interfaces in the living room: usability design for personalized television applications, International Conference on Intelligent User Interfaces, Miami (2003) 230 – 232
10. Van Meurs, L. Zapp! a study of switching behavior during commercial breaks. Journal of Advertising Research, 38(1) (1998)
11. Jon O'Brien and Tom Rodden and Mark Rouncefield and John Hughes, At home with the technology: an ethnographic study of a set-top-box trial, ACM Trans. Comput.-Hum. Interact. (1999) 282-308
12. Venkatesh, A. Computers and other interactive technologies for the home. Commun. ACM 39, 12, (1996) 47–54
13. Moses, J. The computer in the home. In The Computer Age: A Twenty-Year Review, M. Dertouzos and J. Moses, Eds. MIT Press, Cambridge, Mass. (1981)
14. Rob Kling, Social Analyses of Computing: Theoretical Perspectives in Recent Empirical Research, ACM Computing Surveys (CSUR), v.12 n.1, (1980) 61-110
15. Åsa Blomquist and Mattias Arvola, Personas in action: ethnography in an interaction design team, Proceedings of the second Nordic conference on Human-computer interaction, Aarhus, Denmark, (2002) 197-200.
16. Sinha R., Persona development for information-rich domains, CHI '03 extended abstracts on Human factors in computing systems, Ft. Lauderdale, Florida, USA (2003)
17. Alison L. Kidd and Martin B. Cooper, Man-machine interface issues in the construction and use of an expert system, Int. J. Man-Mach. Stud. Vol. 22, no 1, (1985) 91-102
18. Brykczynski B., A survey of software inspection checklists, ACM SIGSOFT Software Engineering Notes, vol. 24, Issue 1, (1999)
19. Tullis, T. S. Is user interface design just common sense? In M. J. Smith, & G. Salvendy (Eds.) Human-Computer Interaction: Application and case Studies. Advances in Human Factors/Ergonomics, 19A, (1993) 9 - 14.

Supporting Personalized Interaction with an ECA in Public Spaces

Addolorata Cavalluzzi, Giovanni Cozzolongo, Berardina De Carolis, Sebastiano Pizzutilo

Intelligent Interfaces
Department of Informatics
University of Bari
{cavalluzzi, cozzolongo, decarolis, pizzutilo}@di.uniba.it

Abstract. In this paper, we present the first results of a research aiming at developing an intelligent agent able to interact with users in public spaces through a touch screen or a personal device. The agent communication is adapted to the situation at both content and presentation levels, by generating an appropriate combination of verbal and non-verbal agent behaviours.

1 Introduction

This research is based on prior work of various research teams and of our group on dialog simulation, emotion modelling and personalization in ubiquitous computing [1,2]. In our opinion, an ECA (Embodied Conversational Agent) represents a creative expression of a natural and engaging interface between users and services of smart environments [5].
In this paper we discuss, in particular, the personalization of conversations with an embodied intelligent agent in public spaces, where interaction may be performed using a public kiosk or through a personal device. The last option allows to overcome to limit typical of public space interaction such as lack of personalization in presenting the information and I/O problems due to the noisy nature of the environment.
In our approach, personalization and adaptation of the agent behavior is made according to the functional architecture of our system shown in Figure 1. This architecture includes two main components: the first one (on the central and right side of the figure) represents the Environment Agent architecture and is installed on a PC, the second one (on the left side of the figure) runs on the PDA and is responsible for handling personal interaction.
When a user move activates a goal, the plan that best suits to achieve the most urgent and important goal in the given context is then applied. Once the agent move has been planned, every communicative act is rendered with an XML string compliant to APML (Affective Presentation Markup Language: [4]). This string combines the text with tags

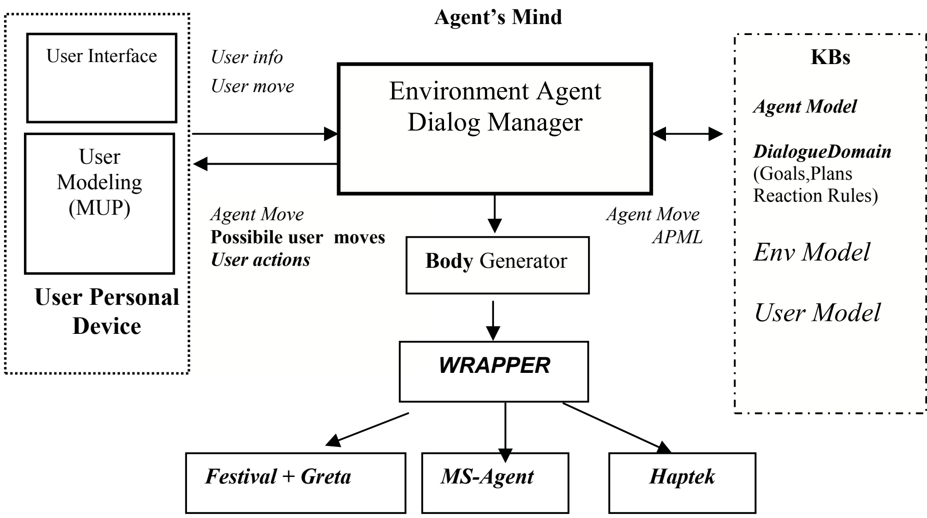


Fig. 1. Functional components of our conversational system

which specify the 'meanings' that the Body Generator Module has to render with a combination of body signals (verbal and non verbal behaviours). As mentioned previously, the agent embodiment can differ according to several factors like the type of environment, the role, the target audience (average age and cultural factors). We may couple the tagged agent move with different bodies, by using appropriate wrappers. So far, we developed APML-wrappers for Greta [8], Haptek [6] and MS-Agent technology [9].

On the other side the personal component enables the user to interact with the environment agent by means of a personal device. It includes two sub-components, aimed at modeling the user and providing an interface for exchanging messages (dialog moves) with the agent. As far as user modeling is concerned, we adopt a mobile approach [7], in which the user information is stored and modeled on the personal device. When the user approaches one of the active points in which the environment agent is displayed, this component provides information about the user preferences (xml-annotated user info stored in the Mobile User Profile) that are relevant for that particular domain. The environment agent can then make its own reasoning about these preferences and adapt interaction accordingly. This approach has the advantage to leave data about the users under control of the users themselves.

During the interaction, the user may ask information to the agent in that particular domain, by using the interface of his/her personal device. The Agent answers by using a multimodal presentation strategy in the kiosk and by sending, at the same time, to the PDA its output move in written form. Possible user answers or reactions which are

enabled in that phase of the dialog are sent as well to the PDA, to be displayed as interface icons; the user may also make a different move by means of a text-based input, which is parsed by a simple keyword recognition method.
An example of system functioning will be shown at the Workshop.

REFERENCES

1. de Rosis F., De Carolis B., Carofiglio V. and Pizzutilo S. Shallow and inner forms of emotional intelligence in advisory dialog simulation. In H Prendinger and M Ishizuka (Eds): "Life-like Characters. Tools, ffective Functions and Applications". Springer, in press.
2. De Carolis B., Pizzutilo S. and Palmisano I. D-Me: Personal interaction in Smart Environments.In P Brusilovsky, A Corbett and F de Rosis (Eds): Proceedings of User Modeling '03.Springer LNAI 2702. 2003.
3. De Carolis B., Pelachaud C., Poggi I. and Steedman M. APML, a Mark-up Language for Believable Behavior
4. Gratch J., Rickel J., Elisabeth Andre, Norman Badler, Justine Cassell, Eric Petajan (2002) Creating Interactive Virtual Humans: Some Assembly Required. IEEE Intelligent Systems 17(4): 54-63.
5. HAPTEK: http://www.haptek.com
6. Kobsa A., Generic User Modeling Systems. UMUAI vol. II nos.1-2 pp.49-63. Kluwer Academic Publisher. 2001.
7. Pelachaud C., Poggi I., Subtleties of Facial Expressions in Embodied Agents, Journal of Visualization and Computer Animation, to appear, 2002 .

Halo: A Virtual Periphery for Small Screens Devices

Patrick Baudisch[1]

Visualization and Interaction Research.
Microsoft Research
+1 (650) 812 4656
baudisch@microsoft.com

As users pan and zoom, display content can disappear into off-screen space, particularly on small-screen devices. The clipping of locations, such as relevant places on a map, can make spatial cognition tasks harder. Halo is a visualization technique that supports spatial cognition by showing users the location of off-screen objects. Halo accomplishes this by surrounding off-screen objects with rings that are just large enough to reach into the border region of the display window. From the portion of the ring that is visible on-screen, users can infer the off-screen location of the object at the center of the ring. In our user study, participants completed all four types of map-based route planning tasks faster when using Halo then when using an arrow-based visualization technique.

Keywords

Halo, visualization, peripheral awareness, off-screen locations, hand-held devices, spatial cognition, maps.

3. Introduction

People use maps in a number of tasks, including finding the nearest relevant location, such as a gas station, or for hand-optimizing a route. Using a map, users can easily compare alternative locations, such as the selection of restaurants shown in Figure 1a (as indicated by the barn-shaped symbols). Users can see how far away a restaurant is from the user's current location, and whether it lies close to other locations the user considers visiting. When users are required to use a zoomed-in view, however, for example to follow driving directions (Figure 1b), relevant locations disappear into off-screen space, making the comparison task difficult[2]. Comparing alternatives then requires users to zoom in and out repeatedly—a time-consuming process that can hardly be accomplished on-the-fly. Especially on small-screen devices, such as car navigation systems or personal navigation devices, this can severely limit a user's capability with respect to spatial cognition tasks.

---

[1] The work reported in this paper was carried out during the author's affiliation with Xerox PARC, now PARC Inc.

[2] See also the concept of *desert fog* in zoomable interfaces [3].

4. Halo

Halo [1] addresses this issue by virtually extending screen space through the visualization of the locations of off-screen objects. Figure 2a shows a map navigation system that is enhanced with Halo. The figure shows the same detail map as Figure 1b, but in addition the display also contains the location information contained in Figure 1a. The latter is encoded by overlaying the display window with translucent arcs, each indicating the location of one of the restaurants located off screen. Figure 2b shows how this works. Each arc is part of a circular ring that surrounds one of the off-screen locations. Although the arc is only a small fragment of the ring, its curvature contains all the information required for inferring the ring center, which is where the off-screen object is located. Although the display window shown in Figure 2a by itself contains no restaurant, the display informs users that there are five of them in the periphery and that the one located southwest is closest.

Figure 1: The problem: In order to make route decisions, users need to see the alternatives (a), but when drilling down to street information, relevant locations disappear into off-screen space (b).

Figure 3 shows how ring sizes are governed. As the map is panned, the restaurant moves from on-screen to off-screen. As the restaurant icon reaches the border region of the display window, a ring grows under the icon. As the restaurant moves further off-screen, ring radii are recomputed dynamically, so that the ring is always just big enough to reach into the border region of the display window while never occluding the display's central region.

Halo combines many of the advantages of other visualization techniques that have been used to display large documents on small screens. Unlike overview + detail visualizations [5] it offers a single view, that allows users to inspect detail information without losing context. Unlike fisheye views [2], Halo displays are not distorted, which is especially important for map-related tasks. Unlike techniques that use arrows to point to off-screen locations (such as City Lights [4]), Halo provides full information about the location of off-screen objects, not only their direction.

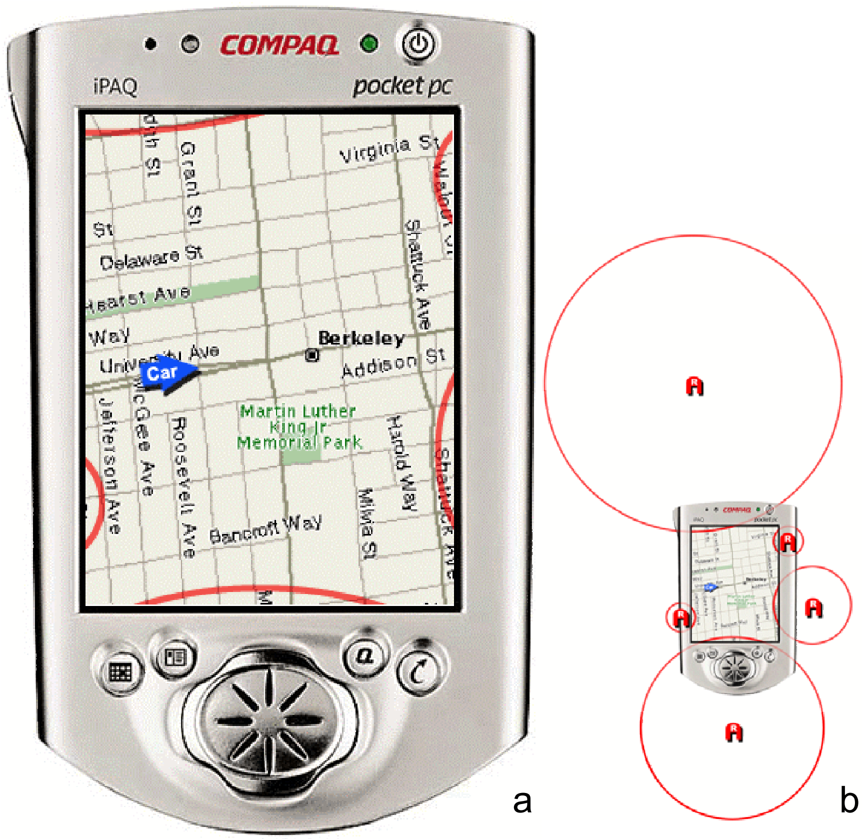


Figure 2: (a) Enhancing the map from Figure 1 with *Halo* shows where in off-screen space the five restaurants are located. (b) How it works: each off-screen location is located in the center of a ring that reaches into the display window.

Figure 3: As this location is panned out of the display window, a ring emerges from its center. The ring grows as the location is panned further away.

5. Halo implements a streetlamp metaphor

The metaphor behind Halo is to represent off-screen locations as abstract "streetlamps" that cast their light onto the ground/map. This metaphor allowed us to derive four important properties for Halo. First, a streetlamp creates an aura, a large artifact which allows observers to infer the lamp's existence even if it is not in view. Second, the aura created on the map is round, allowing users to conclude the center location, even if only a small part of the ring is actually visible. Third, light overlays itself onto objects without occluding them; overlapping auras from multiple lamps aggregate nicely by adding up light intensities. Fourth, the fading of the aura with distance provides an additional visual cue about the distance of the streetlamp. An intense aura indicates a lamp located nearby; a weaker aura indicates a more distant lamp. The final design (Figure 2), has undergone three modifications. First, in order to make it easier to perceive the halo curvature, we replaced the smooth transition at aura edges with a sharp edge. Second, to minimize occlusion of window content and overlap between auras, we replaced the disks with rings. Third, we inverted the color scheme resulting in dark halos on a light background in order to better accommodate typical map material, which used a light background. The concept of fading arcs representing more distant locations was implemented by rendering the short arcs that represent nearby locations as nearly opaque and the longer arcs representing more distant location with increasing translucency.

6. User study

We conducted a user study comparing Halo with a visualization technique using arrows to point to off-screen locations [1]. Participants completed four tasks: locating off-screen locations, picking the closest off-screen location, arranging off-screen locations into the shortest possible traversal, and avoiding traffic jams. Subjects achieved significantly better task completion times in all four tasks when using the Halo interface (16-33% faster). For all four tasks, the majority of subjects expressed a preference for the Halo interface.

7. Conclusions

In this paper, we presented Halo, a visualization technique providing users with location awareness of off-screen objects. Halo provides a single non-distorted view of a document, overlaid with location information for the off-screen locations. Unlike arrow-based visualizations, Halo does not require explicit distance annotation; the distance information is encoded in the arcs themselves and directly incorporates the scale of the scene. Demo and video are at www.patrickbaudisch.com/projects/halo.

In future work, we plan to explore the application of Halo in the area of real-time tasks, such as simulations or highly interactive games where Halo arcs will be used to continuously update users about the location of moving objects in the user's periphery.

Acknowledgments

Thanks to Ruth Rosenholtz, as well as Polle Zellweger, Jock Mackinlay, Lance Good, and Mark Stefik.

References

1. Baudisch, P. and Rosenholtz, R. Halo: A Technique for Visualizing Off-Screen Locations. In *Proc. CHI'03*, pp. 481-488.
2. Carpendale, M.S.T., Montagnese, C. A. Framework for Unifying Presentation Space. In *Proc. UIST'01*, pp 61–70.
3. Jul, S. and Furnas, G. Critical zones in desert fog. Aids to Multiscale Navigation In *Proc. UIST 99*, pp.97–106.
4. Mackinlay, J., Good, L., Zellweger, P., Stefik, M., and Baudisch, P. City Lights: Contextual Views in Minimal Space. In *CHI'03 Extended Abstracts*, pp 838-839.
5. Plaisant, C., Carr, D., & Shneiderman, B. Image-Browser Taxonomy and Guidelines for Designers, *IEEE Software*, 12(2):21–32, 1995.

Author Index