Proceedings
of the International Workshop on

Ubiquitous and **De**centralized **U**ser **M**odeling

UBIDEUM 2007

11th International Conference on
User Modeling, UM 2007

Corfu, Greece, June 26th, 2007

Shlomo Berkovsky, Keith Cheverst, Peter Dolog, Dominik Heckmann,
Tsvi Kuflik, Jerome Picault, Phivos Mylonas, Julita Vassileva

Preface

Currently, issues relating to ubiquitous and decentralized user modeling are gaining more and more attention from research groups representing the user modeling community, the HCI community, the multi-agent systems community, and the ubiquitous computing community. Additional communities that are active in this area include those concentrating on web services, service-oriented architectures and innovative information systems. Many of these communities are moving away from a technically focused approach and starting to consider more user-centered approaches to context awareness and adaptation is among their central research problems. The goal of this workshop was to bring together both academic and industrial researchers from these communities to discuss new and innovative approaches to ubiquitous and decentralized user modeling, to enhance the exchange of ideas and concepts, to determine the veins along the research should proceed and to go one step further towards achieving the widespread adoption personalization in ubiquitous computing.

Ubiquitous user modeling implies new challenges of scalability, scrutability, privacy and trust. Furthermore, new issues of decentralization and integration have to be addressed. Topics of interest included:

- Generic user modeling in mobile and ubiquitous computing
- Context aware ubiquitous user modeling (in mobile and distributed environments)
- Construction and acquisition of distributed user models
- Semantic web approaches for user modeling (i.e. user model ontologies)
- Privacy, security and trust in decentralized user modeling
- Personalized and adaptive applications and interfaces in decentralized and ubiquitous environments
- Case studies, user experience and evaluation of ubiquitous and decentralized UM approaches
- Distributed architectures and interoperability of personalized applications like recommender systems, adaptive hypermedia, e-

learning, adaptive navigation guides, personalized shopping guides, etc.

- Service-oriented architectures for decentralized and ubiquitous user modeling and adaptive systems
- Dynamic changes and their implications on the adaptive services in decentralized and ubiquitous environments
- Knowledge modeling, integration and management for personalization in constrained environments
- Reasoning methods in constrained environments
- Personalized content authoring, delivery and access in mobile environments
- Personalized multimedia applications
- Ubiquitous access to personalized applications
- Challenges for user personalization in mobile/distributed environments

The papers accepted for presentation at the workshop represents the above diversity of topics in that area, they vary from abstract, theoretical frameworks to systems architectures, in various specific application areas.

The organizers

Shlomo Berkovsky, Keith Cheverst, Peter Dolog, Dominik Heckmann, Tsvi Kuflik, Jerome Picault, Phivos Mylonas, Julita Vassileva.

Acknowledgements

Many people contributed to the success of the workshop. We owe special thanks to the members of the program committee for their efforts in reviewing the workshop submissions.

Lora Aroyo, the Netherlands
Yannis Avrithis, Greece
Mathias Bauer, Germany
Joerg Baus, Germany
Bettina Berendt, Germany
Boris Brandherm, Germany
Francesca Carmagnola, Italy
Pablo Castells, Spain
Nadja de Carolis, Italy
Pedro Concejero, Spain
Cristina Gena, Italy
Anind Dey, USA
Tatiana Gavrilova, Russia
Geert-Jan Houben, Belgium
Antony Jameson, Germany
Judy Kay, Australia
Afred Kobsa, USA
Yiannis Kompatsiaris, Greece
Alexander Kroener, Germany
Daniel Kudenko, United Kingdom
Andreas Lorenz, Germany
Gord McCalla, Canada
Marcin Paprzycki, Poland
Myriam Ribiere, France
Ilaria Torre, Italy
Manolis Wallace, Greece
Ingrid Zukerman, Australia

Table of Contents

Abstract of Invited talk

Theories

Implementations

Business Issues Related to Mobile and User Modelling

Makram Bouzid – Jean Millerat

Motorola Labs
Parc Les Algorithmes, Saint-Aubin, 91193 Gif-sur-Yvette Cedex, France
[Makram.Bouzid, Jean.Millerat]@motorola.com

Abstract. Personalising or intelligently adapting an application to user's behaviour and context requires gathering and processing some information about the user. Data collected for user modelling may contain personal and sometimes confidential information. Hence, privacy issues raise with the use of personalisation and context-awareness technology. Is personal data collected and used appropriately, with full awareness and consent of the user? Is the user granted the adequate access rights to manage their personal data? Local privacy regulations have a strong impact on business operations and vary a lot from country to country. Another related issue is that of the security of collected user data; is it kept secured, uncorrupted, with appropriate tracking mechanisms and protection from unauthorized access? The ownership of that data can also be questioned: user profiles may describe how the user relates to their environment (social, digital or physical). As such, it contains both information about the user and their environment. Does such information fully belong to the user? Or to tuples of users involved in a given transaction or communication? Can it be owned by the content provider? By the network or service operator? Or by the device manufacturer? Last but not least, beyond legal and security matters, personalisation systems are only accepted when they are of good-enough quality and accuracy. In order to be accepted by users, their efficiency must exceed some quality threshold for users to accept them. By adopting personalisation systems, they not only put some of their privacy at risk but also their time and attention as they may receive irrelevant recommendations from poorly performing personalisation systems, as well as their money since the cost of the system is at the end somehow supported by end-users. When is a personalisation system good enough for being compliant with the users' expectations in terms of quality of service as well as in terms of cost? The economic viability of content-related business models may vastly rely on the accuracy allowed by the current state of the art in personalisation systems. For instance, how far can mobile advertisements targeting go in proposing business tradeoffs between the need of the advertisers to achieve high efficiency in advertising campaigns, the need for privacy of users and their hostility toward being distracted by irrelevant ads on their personal mobile device? Is there an appropriate price for letting users abandon their privacy and accept reading poorly targeted ads? Can advanced user modelling systems do anything better than that? All these issues will be presented and discussed through our talk planned during the UbiDeUM'2007 workshop (International Workshop on Ubiquitous and Decentralized User Modelling).

Providing Context-Aware Personalization through Cross-Context Reasoning of User Modeling Data

Shlomo Berkovsky[1], Lora Aroyo[2], Dominik Heckmann[3], Geert-Jan Houben[4], Alexander Kröner[3], Tsvi Kuflik[1], Francesco Ricci[5]

[1] University of Haifa, Israel
{slavax@cs,tsvikak@is}.haifa.ac.il

[2] Technical University of Eindhoven, The Netherlands
l.m.aroyo@tue.nl

[3] German Research Center for Artificial Intelligence, Germany
{heckmann,kroener}@dfki.de

[4] Vrije Universiteit Brussel, Belgium
Geert-Jan.Houben@vub.ac.be

[5] Free University of Bolzano/Bozen, Italy
fricci@unibz.it

Abstract. Existing personalization systems base their services on user models that typically disregard the issue of context-awareness. This work focuses on developing mechanisms for cross-context reasoning, i.e., inferences linking user model data in two different contexts. That reasoning process can augment the typically sparse user models, by inferring the missing information from other contextual conditions, and can better support context-aware personalization. Thus, the proposed approach improves existing personalization systems and facilitates provision of more accurate context-aware personalized services.

1 Introduction

During the years, personalization research yielded a number of techniques that facilitate adapting personalized services to the user's interests, needs and constraints. Those are expressed by the User Models (UMs) 7 that constitute an essential input for every personalization technique. However, current personalization techniques suffer from a severe limitation. User preferences represented by the UMs are generally valid only in a specific application and in a specific context, which is typically disregarded by the state-of-the-art personalization systems. Here, considering various contextual conditions can prove essential for providing accurate personalization.

For example, consider a task of recommending radio music for a user during his/her daily driving from home to work. Although the user's music preferences are quite steady, different types of music may be recommended as a function of his/her mood, presence of other people, or traffic and weather conditions. Hence, there is an emergent need for *slicing* the general preferences represented by the UM according to various contextual conditions. This will allow considering the contextual aspects and providing the user with context-aware personalization.

On the one hand, context-aware personalization may significantly improve the accuracy and the usefulness of the provided service. On the other hand, the information stored in the UMs may not suffice for providing accurate context-aware personalization. This deficiency follows from the above slicing of the general UMs that splits the available user information according to the appropriate contextual conditions. Hence, any attempt of inserting the context-awareness dimension into the state-of-the-art personalization systems implies developing a reasoning mechanism, which will facilitate inferring the essential parts of the UMs across various contextual conditions.

This work focuses on developing mechanisms for cross-context reasoning for UMs, which can be applied for the purposes of the subsequent context-aware personalization. The core element of these mechanisms is referred to as user *experience*. By experience we denote an explicit or implicit feedback provided by a user as a result of consuming (i.e., using, purchasing, viewing, reading, listening, browsing and so forth) a certain item in a certain context. Figure 1 schematically illustrates the experience components. For example, a user may rate a pop-music radio program listened when driving alone on a rainy morning by assigning it *4* stars on a *5*-stars scale. In this case, the experience relates for the user, *4* stars to the content of the pop-music radio program in the context of a rainy weather and being alone. The overall collection of such experiences is regarded as the UM. Given such UM, the goal of the cross-context reasoning mechanism is inferring the essential parts of the UM for the purposes of generating context-aware personalization for future user experiences.

Our approach is based on semantically-enriched descriptions of the experiences. This means that all the components of the experience (users, items and contexts), are described using semantic schemata. These schemata facilitate applying various cross-context reasoning mechanisms, that augment the sparse parts of the UM by inferring the missing information from past experiences in other contextual conditions. The inferred UM data is further used for providing context-aware personalization services.

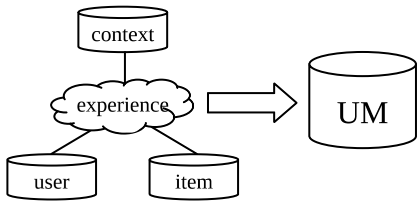

Fig. 1. Representation of Users Experiences and UM

Hence, context-aware personalization can be represented as a two-stage process. First, context-aware UMs are being inferred from past experiences, and then personalized services are provided to the users. This separation improves the flexibility of the personalization process, as each stage can be implemented using a wide variety of techniques. The decision regarding the concrete technique to be used depends on various factors, such as availability of data, dynamicity of the application domain, data pre-processing capabilities and others. In this paper we sketch a number of illustrative techniques, focusing on the UMs inference, rather than on the personalization stage.

The contribution of our work is three-fold. First, we provide a high-level framework for semantic representation of context-aware user experiences. Second, we exploit this framework for defining extensible reasoning mechanisms for inferring the essential parts of context-aware UMs. This upgrades the capabilities of personalization systems and facilitates provision of accurate context-aware personalization. Third, we describe our ongoing efforts towards an evaluation of the proposed approach in different scenarios.

2 Related Work

Rich context models are of special value for supporting activities, in which the user is confronted with diverse and probably unexpected situations. For instance, when guiding a user during a sightseeing tour, an assistant may adapt its personalization with respect to contextual information such as available time, financial limitations, mobility constraints, and local weather conditions (see 3 and 2). Here, the user stays within a single coarse-grain tour context. Other scenarios combine rich context models adapted to diverse tasks. For instance, 8 describes a context-aware assistant for avoiding nursing accidents in hospitals. It distinguishes between diverse context models of various nursing tasks and predicts actions before their actual occurrence.

The above works exploit UMs, providing information about the user in diverse contexts. Such contextualized user modeling is a research area on its own. As pointed out in 4 and 6, context-based user modeling may be performed on the level of sensor data. Our work aims at a higher level of abstraction, in particular, at a UM built from semantic structures. Along this approach, 9 proposed the use of a common ontology-based user-context model as a basis for the exchange of UMs across applications. Here, the context is modeled as an extensible set of facets representing the characteristics of the user and his current context. Ubiquitous user modeling 5 extends this idea by continuously modeling the user by means of situational statements, which enable modeling the user in (ideally) any context. However, if the user is currently in a context not encountered before, which information from previous contexts could be exploited for user support? Therefore, we propose a reasoning mechanism allowing to assemble a UM for a given situation based on a collection of previous experiences.

3 Example Scenario

Everyday life is composed of various events where users request information suited to the individual characteristics of the users and the contexts. For example, let us consider two scenarios: one is defined for a work day and the other for a vacation day.

Let us imagine a traveler, who is married with two kids, likes music, nature, outdoor and water sports, and Italian cuisine. Contextual information implies a summer day with a nice weather and driving a private car. In the following scenarios we highlight the context-aware personalization services provided in form of recommendations.

1. Working day: Traveling for about an hour to a city nearby for a business meeting. The meeting is planned to start at 10:00 and end at 12:00, and the traveler is expected to return at 14:30 for another meeting.
2. Vacation day: Traveling to the same city for a daily vacation. During the day, the traveler will go to a lake, spend some time there, and return home afternoon after enjoying preferred water sports and lunch.

In the first scenario, the recommendation for traveling is to leave home at 08:30 (after rush hour), allowing some time for traffic congestions and planning to arrive early. In addition to the above contextual details, the context implies that the traveler travels alone, time is morning, season is summer, and travel goal is work. Recommendations are required about the road and parking place next to the meeting place. During the trip, another recommendation task arises: music selection. Our traveler drives on the highway, listening to a favorite singer, gets to the meeting place early, parks in walking distance from the meeting place. There are 15 minutes to wait, so the system recommends having coffee at a nearby bar.

In the second scenario, there are no time constraints, so the system suggests leaving at 09:00 to avoid traffic, taking a scenic road to the lake, parking in a free parking area far from the city, but where surfing equipment can be rented and some restaurants are available. During the trip to the lake, the system suggests a favored country CD.

These scenarios detail two possible flows of very similar activities for the same user in two different contexts: work and leisure. They clearly demonstrate the effect of context-awareness on the recommendations generated by the system.

4 Data Representation

The fundamental problem related to data representation is "how can this heterogeneous situational information be represented in a uniform and semantically-enriched fashion"? We addressed it by basing our approach on so-called s*ituational statements* 5 that serve as integrating data structures for user modeling and context-awareness.

The basic idea behind situational statements is to apply predefined meta-level information in an extended RDF representation with OWL ontologies. These ontologies provide a shared and common understanding of a domain allowing communication between heterogeneous widely spread systems. The recently defined general UM ontology GUMO 5 is collecting the user's dimensions modeled within user-adaptive systems, e.g., the user's age, and occupation. Furthermore, it also facilitates representing the user's interests and preferences.

In a similar manner, GUMO also facilitates modeling various dimensions of contexts (e.g., for the traveling scenario: day-time, season, companions, motivation and others) and items. Hence, we updated GUMO and inserted there several intuitive contextual dimensions. In general, we assume that the modeling of context is a one-time task that can be conducted by domain experts. Figure 2 illustrates partial representation of context in GUMO.

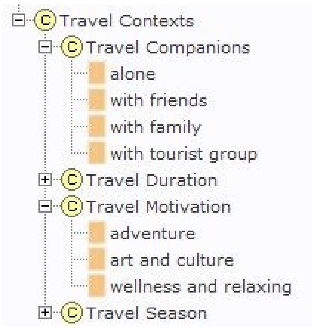

Fig.2. GUMO Context Representation

5 Reasoning Model

The above representation of UM data in RDF/OWL format facilitates provision of context-aware personalization. To explain the cross-context inference mechanism, we need to consider the UM data in more detail. User experience was previously defined as the feedback a user provided for a certain item in a certain context. For capturing this context, we use (in a simplified syntax) situational descriptions, such as:

```
context.time=afternoon
```

For example, consider experience *E* representing a situation, an item, and a rating:

```
context.motivation=work
context.time=afternoon
item.meal.price=moderate
rating=0.8
```

One possible approach for the inference task is defining reasoning rules (over the RDF representation) that reflect the relations between various contextual aspects. This approach is referred to as a *rule-based inference*. For example, rule-based inference can be done by abstraction rules deriving knowledge about more generic situations from more specific ones by discarding some contextual information. Consider a rule:

```
context.motivation AND context.time => context.time
```

It allows aggregating detailed knowledge referring to `context.motivation` and `context.time` into coarse-grained knowledge that refers to `context.time` only.

As such, abstraction rules define the factors that are more important for the context-awareness and help to deal with general situations, such as

```
context.time=afternoon
```

by inferring that for this situation, experience *E* can be used as a basis for recommendation. They allow inferring the missing parts of the UMs, even if no past experiences have been recorded for the very specific situation.

Another type of reasoning rules exploits knowledge about the relations between the values of certain contextual aspects. For example, consider a contextual situation *S*:

```
context.motivation=work
context.time=4PM
```

The experience *E* and the rule

```
4PM => afternoon
```

allow inferring the UM data also for the new situation *S*. So, the first type of rules is associated with the presence of certain semantical context aspects (the generality of the situations), whereas the second type is associated with the semantical structure of the domains and the domain knowledge. Needless to say that it is possible to define rules combining the above two types.

We would like to stress the fact that the above examples relate to the same user and item in a different context, i.e., new situation and previous experiences of the same user on the same item. Hence, these are examples of pure cross-context reasoning. Also, we can include the items in the rules, yielding cross-context cross-item reasoning from past experiences on other items in other contexts. Instead, including the users in the rules yields cross-context collaborative (cross-user) reasoning from past experiences of other users in other contexts. Currently, we only point out these possibilities, without exploring them in depth.

Another alternative approach for the inference task is adapting past experiences of similar users on similar items in similar contexts. This approach, usually applied in Case-Based personalization systems, is referred to as *similarity-based reasoning*. In this case, the reasoning process should determine the re-using (adaptation) mechanism of past experiences, which, in turn, can also be based on rules.

In this type of reasoning, to infer the missing part of a context-aware UM for a new experience, this experience is compared against previously recorded experiences. Then, a set of most similar experiences (the *K* most similar ones, or all those with a similarity over a threshold) are retrieved. Finally, the retrieved experiences and their ratings are re-used to infer UM data for the new experience.

Applying similarity-based reasoning requires defining a stable similarity metric for the experiences. We propose to base the similarity metric on the above RDF representation of the experiences, which guarantees that the experiences will be represented in a semi-structured form (despite a predefined structure, values of some slots may be unavailable). The similarity metric will facilitate retrieving similar experiences, and then aggregating them.

When we compare the above rule-based and similarity-based reasoning approaches, we see that on the one hand, rule-based reasoning may produce more accurate UM data, as in the typical scenario the rules are defined by domain experts. On the other hand, manual definition of inference rules may hamper scalability and flexibility. Conversely, the typical scenario for similarity-based reasoning is fully autonomous and gives, therefore, a more flexible process. However, to achieve this there is a need for a large number of past experiences to bootstrap the reasoning process. For this, other machine learning and reasoning approaches can be considered.

Once the required parts of the UM are inferred the reasoning stage is completed. The following stage of the context-aware personalization process deals with generating the recommendations. Any state-of-the-art recommendation technique may be exploited here. Separating reasoning and recommendation stages in the personalization allows higher flexibility, as higher computational effort may be put on either stage, independently of the other stage.

6 Evaluation Scenarios

We have conducted a number of initial valuations of the proposed approach in context-dependent scenarios within the *Passepartout* 1 project. The prototype illustrates typical search, browse and viewing activities with a personalized digital TV program guide. Users normally appear individually or in groups, and are switching between different daily, weekly, monthly or yearly contexts, with the same or similar preferences and interests. Hence, it is important to identify the right granularity for the cross-context reasoning. Performing comparative studies with predefined granularity settings for spatio-temporal aspects, context, and UM characteristics allows finding the most efficient setting for gathering experiences in different contexts.

We tested how our semantics-driven techniques improve the search for TV content in terms of increasing the recall by broadening the search, as well as the precision by finding the most relevant content. As test set we used real TV programs metadata collected dynamically daily over the Web. User profiles for six test users were constructed and given various interests and contexts. The results show that applying domain ontologies for finding concepts that (semantically) correspond to the keywords (i.e. synonyms, related terms, background, and context) dramatically improves the shortcomings described for using keywords only. We apply the user model concepts in the post-processing of results, as opposed to the ontology models which are used in the pre-processing of the search query before a call to the metadata service is made. As the employment of ontology models improves our search by making it broader, the user model is used to narrow the results and arrange the presented results in a personalized order. The more advanced effects occur when values in the user model are connected to a context. Without the notion of context, the interest values in the user models can only provide general recommendations, whereas in reality people may have very different TV interests, e.g., in the morning and in the evening, as we have discussed earlier.

Our test results illustrate two different aspects in which context can be used: (1) restricting an interest expression expressed by the user to be valid only in a particular context; and (2) automatic inclusion of the user's current context data, in addition to the context-based interest that may be valid, for the purpose of reducing the input that the user has to give (e.g., automatic inclusion of the user context intro the predicted experience). We also plan to use the geographical data in the filtering of results instead of adding it as a search term.

We have also planned further evaluations within the *SharedLife* 10 project. It is a multi-user shopping scenario, completed by other everyday activities. A positive experience observed in a certain situation is exploited for recommendations in other situations. However, this should be justified by a sufficient overlapping between the situations, since the positive feedback might actually relate to several context elements. Another rich field for experimenting is the configuration of the UM sharing behavior. For instance, a user might be more willing to deal with incoming sharing requests during a relaxed shopping trip than during cooking. Since such sharing can be treated as experiences, similarity between experiences and context reasoning rules provide a means to extracting information on when requests should be presented to the user, and when the system should try to handle them automatically.

In both projects, the evaluations assess the use of rich semantic information (e.g. GEO/TIME ontologies and contextualized UMs), to find proper settings to include in the reasoning, and the metadata and the semantic structures to present to the user. In both, users are collecting and exchanging experiences across various contexts, allowing evaluating of various cross-context reasoning approaches.

7 Conclusions and Future Work

This paper motivates the need for context-aware personalization and suggests an initial model for it. The model is based on cross-context reasoning, applied over semantically enhanced descriptions of user experiences. Note that the proposed cross-context reasoning model is extensible. It may be integrated with other personalization approaches, e.g., cross-user and cross-item reasoning. Hence, it integrates with the ideas adapted from the state-of-the art personalization techniques in order to provide a complete framework for provision of context-aware personalization services.

Future research will focus on formalizing the model, integrating it with known representation and reasoning techniques, demonstrating it in a real-life scenario as proof of concept and evaluating various reasoning mechanisms.

References

1. Aroyo, L., et al. (2007). SenSee: Personalized Ambient Media. In Multimedia Tools and Applications, to appear.
2. Cinotti T.S., et al. (2004). Evaluating Context-Aware Mobile Applications in Museums: Experiences from the MUSE Project. In proc. of Museums and the Web Conference.
3. Davies, N., et al. (2001). Using and Determining Location in a Context-Sensitive Tour Guide. In IEEE Computer Society Press.
4. Harvel, L., et al. (2004). Context Cube: Flexible and Effective Manipulation of Sensed Context Data. In proc. of the International Conference on Pervasive Computing.
5. D. Heckmann et al. (2005). GUMO – the General User Model Ontology. In proc. of the International Conference on User Modeling.
6. Kern, N., et al. (2006). Context Annotation for a Live Life Recording. In Journal of Personal and Ubiquitous Computing, Special Issue on Memory and Sharing of Experiences
7. Kobsa, A., et al. (2006). An LDAP-Based User Modeling Server and its Evaluation. In User Modeling and User-Adapted Interaction: The Journal of Personalization Research.
8. Kuwahara, N., et al. (2003). Wearable Auto-Event-Recording of Medical Nursing. In proc. of the Conference on Human-Computer Interaction.
9. Mehta, B., et al. (2005). Ontologically-Enriched Unified User Modeling for Cross-System Personalization. In proc. of the International Conference on User Modeling.
10. Wahlster, W., et al (2006). SharedLife: Towards Selective Sharing of Augmented Personal Memories. In 'Reasoning, Action and Interaction in AI Theories and Systems', Springer.

Policies for Distributed User Modeling in Online Communities

Tariq Muhammad and Julita Vassileva
Department of Computer Science
University of Saskatchewan
110 Science Place Saskatoon, SK S7N5A9
CANADA
{tam706, jiv}@cs.usask.ca

Abstract. Allowing collaboration between online communities can result in a fragmented user profile. Each community will maintain a profile of user based on local context and policies. To express, discover, interpret and update these '*localized*' fragments, we propose to use policies defined by the community owners. When users move across communities, this movement can be regulated through *transfer policies*. Policies can be used to enforce the access rights and implement adaptations to users' status and roles. This policy–based user modeling approach is a variant of the purpose-based decentralized user modeling approach which computes user models and carries out adaptation on demand from fragmented user model data available in the different collaborating communities.

Keywords: Purpose-based user modeling, User policies, Online Communities.

1 INTRODUCTION

The basic purpose of online communities is to support social interactions and exchange of digital resources among people (Kimberly et al., 2003), (DeSouza and Preece, 2004). In the physical world, we see the movement of people from one place to another due to economic or social reasons. Such movement results in depopulation at one place and overpopulation at another. Although also being susceptible to user migration, online communities should not fall victim to this phenomenon. In the virtual world, the availability of technological tools such as web services, make the "virtual merger" of online communities possible. Still, the current designs of online communities do not focus on allowing collaboration among large online groups. Most existing communities are independent from each other; allow no sharing and/or interaction across online community borders, thus losing the potential advantage that the virtual world has over the physical world in terms of sharing time and space. Inter-community collaboration can help resolve this issue of participation and sustainability. One of the main design problems to ensure inter-community collaboration is the transfer of the user data, including the user identity and user model, across online communities.

Most of the online communities manage user models for multiple reasons, varying from authentication to personalization. Users create individual accounts in different communities and they have to start from scratch their participation and building their reputation in each community. This results in a fragmented user models across the communities. Even if the designers / moderators of two communities agree on exchange of contents, it is hard to transfer a user model across two sites. The reason is that different online community applications typically use different database organizations or different ontologies and can therefore not transfer and understand the user model data received upon request from another application (community). The need arises for a mechanism to create a user model on request just in time according to the current context.

Collaborating online communities face user modeling challenges similar to those in open environments with ubiquitous, service-oriented or agent-based applications. User models in these environments are fragments developed for the adaptation purposes of each service, agent or ubiquitous computing applications, and stored locally by these virtually or physically distributed applications (nodes). Some of the emerging challenges are interoperability, updating and synchronization of user models across these nodes, while preserving the autonomy of each application, service or agent. Applications have been generating and updating user models with the help of procedures. The procedural approach focuses on the algorithms of locating, deducing, and using user data for adaptation, rather than on the user data representation. However the procedural approach results in a use–specific model (Anderson, 1988), where changes in the user characteristics are hard to implement and collaboration always requires changes. Therefore, the opposite, declarative approach is the currently popular solution, where facts can be added and removed freely in a uniformly represented user model (expressed according to agreed upon ontology and language, ensuring interoperability) and applications can use a standard reasoning mechanism to make conclusions based on the information in user models. The declarative approach focuses on the expression of the user model instead of the discovery, interpretation and integration of user data, since these processes are standardized. However, in the autonomous and diverse online communities existing currently it is impossible to ensure such standardization.

User models in online communities are based on policies describing the role, status and rights of each user to ensure security, adaptation and awareness. These policies are implemented in procedures. The purpose-based user modeling approach proposed by Niu et al. (2004) can be implemented through policies as an abstraction layer ensuring functionalities like a shared user data taxonomy/ontology, security, and discovery of user information fragments, interpretation and integration of user models. While not focusing on the standard representation of user model data, this approach does not preclude it, and it puts emphasis on the processing of the data in context. Therefore, we believe that this approach is suitable for the problem of sharing user data in collaborating online communities. The paper explains how policies can be used for managing, transferring user data in multi-community environment Comtella (http://umtella.usask.ca/um).

2 RELATED WORK

The Comtella system is a web-based online community framework, which was created in the MADMUC lab to support resource sharing and discussion by students (Cheng and Vassileva 2005). The users in Comtella are assigned different *status* to reward them for participation. The status is computed based on the number of desirable actions the users perform and is rewarded with certain privileges. The users of Comtella can take also different *roles*. They can create new communities and become *owners* of communities. In the same time the architecture of Comtella has evolved from a single web-based system for one community sharing URLs for different topics (each topic being a focus of the entire community for a given time), to a single system hosting many communities created by different users (each focused on a different topic), and finally to a multi-node system, consisting of many systems at different websites, hosted by different organizations and administered by users in the roles of *administrators*. The new design of Comtella allows communities to be hosted in different web sites (nodes). Communities can collaborate within and across nodes. Members of one community can join other communities and transfer there their old user profile from their previous community; they can maintain different roles and statuses (with their associated rights and privileges) in each community.

One of the basic purposes of user models in multi-user applications, apart from personalization, is to ensure security of computer systems. The Role-Based Access Control (RBAC) system was created for the first multi-user computer environments and has been used widely in web-enabled applications. In role-based access control systems users are associated with a roles defined according to the operational needs of groups and organizations. Rights of access are defined at the role level. Users can work in one or more roles and can perform actions associated with these roles (Mohammed and Dilts 1994, Sandhu and Park 1998, Park, Sandhu and Ahn 2001).

Reward-based communities like Comtella cannot model all users with RBAC, since these applications model not just roles, but also user goals, capabilities, user attitudes and knowledge (Kass and Finn 1988). Kagal et al. (2001) proposed an ontology-based RBAC approach for pervasive computing environments. This approach allows not only representing role hierarchies but also other user properties, which are expressed in XML language. Denaux et al. (2005) proposed an ontology-based user modeling to allow for interoperability and overcome the "cold start" problem.

Many applications keep their user model hidden from the user. However, applications such as learning environments often deploy an open learner (user) model, so that the learner can interact with the model to reflect on its content or to correct errors (Bull and Pain 1995, Bull 1997, Vassileva et al., 1999). A user model framework that is based on user policies can open the user model both for the user and for other systems. Policies will not only communicate the current status of the user but also explain why she has gained this status.

Agent-based software environments, mobile applications and online communities can not work with monolithic user models as each point maintains a local profile of the user according to context. Distributed user modeling or decentralized user modeling is an option for such environments. In this approach user information is scattered around in independent and autonomous agents as user model fragments. Each agent develops these fragments according to its context and preferences. The properties and issues of these '*fragmented, relativized, local and often quite shallow*' user models is described by Vassileva et al. (1999, 2003). The active modeling approach is a decentralized user modeling for learning environments (McCalla et al, 2000). Active learner modeling can be combined with open user models to create small fragmented models just in time when requested by the user (Hansen and McCalla, 2003). Purpose-based user modeling (Niu et al, 2004) is an approach that involves computing distributed and fragmented user models from various decentralized sources for a specific purpose. The purpose consists of a process and the user data types it requires as input and output. The process computes new user model data type and/or provides a certain application-dependent adaptation. Thus, a purpose is an independent processing unit, which can be applied to whatever fragmented user data is available at the moment from available sources. The purposes can work together in an anytime manner in a hierarchy based on abstraction. More specific purposes positioned towards the leaf nodes are executed when more data from fragmented sources is available while more general purposes near the root typically demand less or easier to access data. The purpose-based modeling approach has two advantages: speed and providing a local context for computing the model fragment and adaptation.

Purpose-based user modeling can be implemented using policies instead of purposes to compute user models on the fly in online communities. The policies define the rights and privileges of users in the new communities that they join, so they do not have to start from scratch as new users. The policy document, like a purpose (Niu et al, 2004), describes a procedure, but it is also declarative in some sense, since it is modular, human-readable and editable according to the wishes of the community owner or node administrator. A policy provides all the relevant information for computing a user model and adaptation of the functionality and interface to a given type of user in a given context, e.g. when visiting a community. Through appropriate policies online communities can collaborate and transition of users across communities can be made smoother. In the next section we explain how a policy driven framework can implement a purpose-based user modeling approach for collaborating online communities.

3 POLICY-DRIVEN ONLINE COMMUNITIES

Allowing users to move across communities results in a user profile fragments in all of the visited communities and requires interoperability of their user modeling components and a trust relationship among the collaborating communities. A typical user joins one community according to her primary interest. However the same user can visit other communities of marginal interest. In Comtella the user models are represented in a database which is updated according to user policies. By inspecting the community policies a user can understand the reason for the current state of the user model in a given context (provenance). Different policies command the transfer of user data along with the user's identity to any new community where a new user model can be established according to the context. Policies in Comtella determine the access rights, the status of users and user roles (and the privileges associated with roles and status) both in the home communities of the users and in new communities they are visiting. Policies can be created only by users in a particular role – the role of community owner (the user who created the community). The owner of a community creates and manages four types of policies: access control policies, status policies, role policies and transfer policies. Examples of these policies are shown in Fig. 1.

Status Policies

A. Policy to update user participation		Description
Policy Type:	Status	This is a policy defines how the measure of user participation (UP) is calculated in community "Pictures". UP is an important derived user data in the user model. Specifically, the policy defines weights for the different user activities (primary user data, e.g. *pn* – number of shared papers, *pq* – average rating of user's shared papers, *rn* – number of ratings given by user and *rq* – similarity of ratings given by user to the average rating of the paper) and the formula for the UP calculation.
Effective Date	Jan 10, 2007	
Node	http://kardam.usask.ca	
Community id:	1	
Community Title:	Pictures	
Weight for Paper Quantity (Wpn)	3	
Weight for Paper Quality (Wpq)	4	
Weight for Rating Quantity (Wrn)	3	
Weight for Rating Quality (Wrq)	4	
Action	UP:= *Wpn*pn+Wpq*pq+Wrn*rn+Wrq*rq*	

B. Policy to calculate user status level				Description
Level	Description	Start Value	End Value	This policy is used to classify the user into appropriate status level, i.e. to compute the user status (US) depending on his/her participation measure (UP). It sets the margins of each status level.
1	Gold	700	1000	
2	Silver	500	699	
3	Bronze	300	499	
4	Plastic	0	299	

Action	If (Plastic(StartValue)<= UP <= Plastic(Endvalue)) → US:=Plastic
	If(Bronze(StartValue)<=UP <=Bronze(Endvalue)) → US:=Bronze
	If (Silver(StartValue)<= UP <= Silver(Endvalue)) → US:=Silver
	If (Gold(StartValue)<= UP <= Gold(Endvalue)) → US:=Gold			

C. Policy for Status Permissions	√ Action allowed × Action not allowed					Description
Level	Description	Share link	Share File	Post	Rate	This policy defines user permissions based on the user status level. These permissions are used for interface adaptation and the effect is to disable certain options to the user.
1	Gold	√	√	√	√	
2	Silver	√	√	√	√	
3	Bronze	√	×	√	√	
4	Plastic	√	×	√	×	
Action	If US==Plastic or US==Bronze → disable "Share File" Interface widget If US==Plastic → disable "Rate" Interface widget					

Role Policies

D. Policy for Role Permissions	√ Action allowed × Action not allowed					**Description**
Level	Role	Delete link	Create Community	Edit Policy	Edit Role	This policy defines access rights and special permissions based on the role of the user. By editing this policy, the community owner can grant to users in different roles special rights and permissions for advanced actions such as "delete link", "create community", "edit policy" and "edit roles".
1	Owner	√	√	√	√	
2	Expert	√	√	√	×	
3	Operator	√	√	×	×	
4	Member	×	√	×	×	
Action	If UR==Member → disable "Delete Link", "Edit Policy", "Edit Role" widgets from the user interface If UR==Operator → disable "Edit Policy", "Edit Role" from interface If UR == Expert → disable "Edit Role" from interface					

Figure 1. Examples of different policies in Comtella.

Access control policies are rules representing conditions under which users can perform certain actions on a resource, such as reading, rating, replying, commenting, or deleting a posting. Usually access control policies are the basic policies that are used by higher level policies, such as status-, role- and transfer-policies to express specific decisions, e.g. allowing or disallowing a user request.

Status policies in Comtella implement the reward mechanism to stimulate desirable actions in the community (Bretzke and Vassileva, 2003). Status policies in Comtella (e.g. the one shown in Figure 1-A) define how the user participation metric is computed based on giving reward points for frequency and quality of certain desirable activities such as sharing and rating resources. Other status policies (E.g. Figure 1-B) describe the computation of the user status attribute. Community owners can manipulate the status policies for their communities and define their own user status levels (e.g. plastic, bronze, silver and gold or regular, prestige, elite) and their point thresholds. The two status policies described above result in changes in the user model. Other status policies define the access permissions that should be granted to users with a certain status. They result in adaptations of the interface that enable or disable certain functionality or look and feel (e.g. Figure 1-C). For example, the gold status users in Comtella have access to the gold-coloured interface frame, while plastic status users have access to the green-coloured interface.

Role policies define the conditions under which users with a given status can acquire a certain role and the accompanying rights and responsibilities. Like any organization online communities should manage a separation of duties. Role policies allow community owners to share the burden of community management with deserving community members. A community owner may designate a few members through either individual policy (by naming individuals) or through a selection-based policy (e.g. all gold-status members) to special roles, such as operators or experts. The moderator can assign special access rights to these roles, such as editing and deleting resources (see example policy in Figure 1-D). Role-based policies result in defining user groups based on their functional responsibilities such as expert, community moderator, and operator.

The three types of polices presented above define how to update the user model and what access rights to grant the user when she is working within her community. Each user in Comtella has a home community, which she can select from all communities hosted on the user's node when she starts using the system. It is expected that the user will contribute and participate mostly in her home community. The user accumulates a participation score which is represented in her main user model (the model related to her activity in her home community). The user's identity is also linked to her home community.

Users can search freely and find resources shared in other communities. In order to access and read these resources, they have to "visit" the other community. When a user moves from one community to another, for example, by requesting access to a resource in a new community, there is a question what rights and privileges, role and status this user should have in the new community. To govern movement of users across two communities, the communities must have a contract/agreement about the status, role and access rights of visiting users. These contracts are called transfer policies and can be unilateral (e.g. the owner of the receiving community defines the policy according to which to treat visitors from specific communities or in general) or bilateral (e.g. the two owners agree about mutual recognition of status, roles and rights). For example, the community owners may decide that visitors from the other community will be given automatically status with one level lower than the status they enjoy in their home community. In Comtella these policies are unilateral. If a user wants to visit a new community (e.g. to read an article posted in this community), she has to send a request to the owner of the new community. The community owner sets a transfer policy after reading the policy under which user was working in her home community. Comtella allows three options to the community owner: (I) enforce the current policy of the community; (II) allow the policy of the user's home community; and (III) define a new policy for visiting users from the user's home community. The definition of a new policy can be achieved by using different approaches. One may be to show the community owner the policies of both communities so that she can compare them and provide the owner with an editing tool allowing her to create a new policy, as shown in Figure 2. In this approach community owners can define new status levels and their respective thresholds. Another approach may be to declare one of the status slots of the community equal to one or more slots of the other community from where a user is coming. We have used the former approach as it provides finer grained control to community owners.

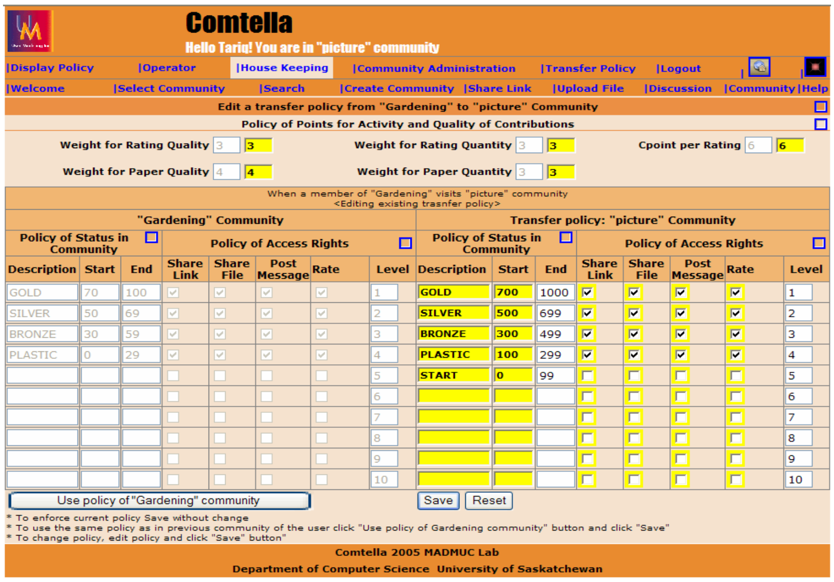

Figure 2: Editing a transfer policy in Comtella

One problem in many learning communities is the '*cold start*' (Denaux et al. 2004), (Sun and Vassileva, 2006) where the system fails to provide adaptation due to the lack of information about users when they first visit a community. With transfer policies the community does not have to wait for the accumulation of user information to offer customization and adaptation. Transfer policies can help in acquiring user model data about the previous experience of the user from other communities. These transfer policies provide a guideline whenever a user may visit a community for the first time. It will give the user a starting point to participate in the community instead of starting from scratch. The subsequent visits of the user will follow the same policy and update profile on every visit. Yet, through transferring back and forth across communities, users may find ways of increasing their status due to the inconsistencies between the policies in each community and too generous transfer policies. Therefore, the transfer policies for temporary visitors should be different from the transfer policies for users who want to make the community their new home community.

4 POLICIES IMPLEMENTING PURPOSES

Each community has a policy framework, which consists of:

- *shared view* used for all context and user data, both raw data as well as calculated user attributes;
- a set of user policies governing the community, each specifying the input data (about the user and context), the process and the output data (user model update or functionality / user interface adaptation);
- an execution mechanism running in a loop which selects an appropriate policy for the current user request and context and executes its process.

The policy framework ensures that the relevant policy is selected depending on the user request (which arrives on the *shared view*). The invoked policy in turn picks the required user data items (either raw data or user model data computed as output by other policies, in the same community or requested from other communities). There are many different policies in the set, which can be seen as managing different levels of decisions. For example, there are high-level policies that compute the role and status of the user in the community, using data received from other communities (which is either raw participation data or data computed by other policies). Lower-level policies control the user access rights using data about the user role or status computed by the higher-level policies. In this way, the framework provides both personalization and a simple security layer to protect against unauthorized users and actions.

Just like the purposes in purpose-based user modeling (Niu et al, 2004) a policy has three components: input, process and output. The input is either raw user data or user model data computed as output by other policies. For example, the input of a policy controlling user access to a community can be a user action attempting to access an item shared in the community. As another example, the input of a policy controlling user actions on community resources can be an action of a user attempting to rate a posting in the community. We call such raw data indicating user intentions a user request. A request consists of three parts: the subject, action and resource (Merrells, 2004) (OASIS, 2005) (Seth, 2004). Here "subject" is a primary identity key produced by a shared identity provider to which the collaborating communities have access or one of a federation of identify providers. This identity key can be used to fetch the user attributes hosted in the user database from both the current community (that is receiving request), from the home community of the user and from any other community which has data about this user. These user attributes can be inputs of another policy, for example, one that decides what status to grant the user in the community.

The process of a policy involves the algorithm that computes in context the output user model data or makes an adaptation decision. The process is executed by execution mechanism of the policy framework which retrieves the local and remote user profile, data required by the policy as input and places it in the *shared view*. The policy framework execution mechanism then computes the policy output data using the available input and current context data from the *shared view* and makes a decision, for example to allow / disallow the request or to adapt functionality or interface. For example, the process of an access control policy distinguishes between new users and local users (whose profiles are stored at the community). For a local user it retrieves the location of her user model, which becomes the output of the purpose and either grants or denies access depending on the role of the user. For new visitors it calls the appropriate transfer policy whose inputs match the user request and the current context and produces its output. The process of the transfer policy (using the user id as input) requests information from all other collaborating communities that have stored a model of this user and according to the mapping algorithm described by the community owner in the process of the transfer policy generates a local user model for the new user, which contains her status and role. This data will then be used as input by the community's access control, status and roles policies that decide about the user's rights and privileges.

5 DISCUSSION AND FUTURE WORK

Collaborating online communities have to deal with fragmented user models. These environments need interoperable, context- and purpose-sensitive user models. Online communities need a shared framework to express, discover, transfer and secure user models. User policies can be used for establishing the user's purpose and compute the required user model just in time. We propose a policy framework with the following advantages:

- Interactions between different communities will result in exchange of both users and contents, which otherwise would not be possible due to island nature of online communities (Harth et al., 2005), (Breslin et al., 2005). In this way there will be no necessity for each community to gain a critical mass of participation to be sustainable by itself.
- Transfer policies provide a starting point for customization for the user without '*cold start*'.

- Explicitly assigned roles for users lead to a more sophisticated user model, representing the context, purpose, trust and reputation of users within and across communities.
- People in other communities will feel more comfortable since 'strangers' will be allowed only after policy negotiation with their trusted domain/community.
- User policies are open and readable for the community members, so that they know the consequences of their actions and activity. Community owners can change the policies according to the changing needs of the communities.
- Availability of policy documents for a community system, owner and user work as tool to establish a trust between these entities.

The user policies implement a purpose-based user modeling approach. Yet there are some differences between the original purpose-based approach (Niu et al., 2004) and our approach. In Nui's approach the whole hierarchy of purposes has access to the raw data items denoted as R1, R2……Rn coming from distributed sources. These raw data items include information that comes with the user's request, information stored by application such as login information in database and data from peer assessment. The output of each purpose is transferred directly from one purpose to another depending on the hierarchical and sequence relationships between the purposes, which are pre-defined at design time. These pre-defined relationships limit the flexibility and reuse of purposes in Niu's approach. In our approach, the outputs of policies (purposes) are placed on the *shared view* and can be used as inputs for other purposes together with the other raw data and context data. This allows for more flexibility.

The policy-based user modeling approach is currently being implemented in Comtella. The next step is to carry out experiments to evaluate its reliability, efficiency, scalability and also the ease of editing policies that the interface provides for community owners. This implementation will also provide a platform to study the dynamics of online community. We envisage carrying out the following studies in the future:

Study of single community: Comtella has been used for the study of reward mechanisms and its effects on the participation in communities. The flexible reward mechanism of the current implementation provides an opportunity to observe the effects of different reward strategies. For example what should be the parameter values at the start of the community to attract users and how the reward mechanism may be adjusted to achieve the quality in contributions of users in the later stages of community's life? This information will be useful for defining the reward policies in the current and future deployments of Comtella and other reward-based communities.

Study of interaction between communities: Previous Comtella studies were focused on a single group and its dynamics. This implementation can be used to study both interactions within one community and interactions between communities. This study will capture the transfer of users between communities and will point out what factors trigger the transfer of users. The percentage of local versus visiting users in a community and statistics of the home-communities of visiting users will help to discover and develop relationships between communities. Knowledge about the movement of users between communities and factors contributing for the movement will be useful for both attracting and retaining users in future communities.

Study of user activity and sustainability: The study of contributions by local and visiting community members will help to appreciate the effects of community collaboration on its sustainability. This study will visualize activities such as *sharing* and *rating* by local and visiting community members. It would be interesting also to study the effects of policy-based user modeling on the *cold start* problem by comparing the time taken by local and visiting users to attain the top status in the community.

These studies will help us also understand better the strengths and limitations of policies as decentralized open user modeling approach and their benefits for enabling inter-community collaboration.

Acknowledgement: This work is supported by the NSERC Discovery Grant of the second author and by funding the NSERC/Cameco Chair for Women in Science and Engineering (Prairies).

REFERENCES

1. Anderson, J. R. (1988). The expert module. In M. Polson & J. Richardson (Eds.), Handbook of Intelligent Training Systems. Hillsdale, NJ: Erlbaum, 21-53.
2. Breslin, J. G. (2005). Towards Semantically-Interlinked Online Communities. Lecture Notes in Computer Science, vol. 3532, 500-515.
3. Bretzke H., Vassileva J. (2003) Motivating Cooperation in Peer to Peer Networks. Proceedings of User Modeling (UM'03), Johnstown, PA, June 22-26, Springer Verlag LNCS 2702, 2003, 218-227.
4. Bull, S., and Nghiem, T. (2002) Helping Learners to Understand Themselves with a Learner Model Open to Students, Peers, and Instructors. Proceedings of Workshop on Individual and Group Modeling Methods that Help Learners Understand Themselves, International Conference on Intelligent Tutoring Systems 2002, page 5-13.
5. Bull, S., and Pain, H. (1995) 'Did I say what I think I said and do you agree with me?': Inspecting and Questioning the Student Model. Proceedings of World Conference on Artificial Intelligence and Education (ACCE). Charlottesville, VA, pages 501-508.
6. Cheng R., Vassileva J. (2006) Design and Evaluation of an Adaptive Incentive Mechanism for Sustained Educational Online Communities. User Modeling and User-Adapted Interaction, special issue on User Modeling Supporting Collaboration and Online Communities, 16(3/4), 321-348.
7. DeSouza, C. S., Preece, J. (2004). A framework for analyzing and understanding online communities. Interacting with Computers, vol. 16, 579-610.
8. Denaux R., V. Dimitrova, and L. Aroyo. (2004) Interactive ontology-based user modeling for personalized learning content management. In AH 2004: Workshop Proceedings Part II, pages 338–347.
9. Denaux, R., Aroyo, L., and Dimitrova, V. (2005). An approach for ontology-based elicitation of user models to enable personalization on the semantic web. In Special interest Tracks and Posters of the 14th international Conference on World Wide Web (Chiba, Japan, May 10 - 14, 2005). WWW '05. ACM Press, New York, NY, 1170-1171. http://doi.acm.org/10.1145/1062745.1062923
10. Hansen c. and McCalla, G. (2003) Active Open Learner Modelling. Proceedings of AIED2003, July 20-24, 2003, Sydney, Australia.
11. Harth A., John G. Breslin, Ina O'Murchu, Stefan Decker (2005) Linking Semantically-Enabled Online Community The Semantic Web: Research and Applications. Second European Semantic Web Conference, ESWC 2005. Proceedings (Lecture Notes in Computer Science Vol. 3532), 2005, p 500-14
12. Kagal, L., Finin, T., and Joshi, A. (2001). Trust-Based Security in Pervasive Computing Environments. Computer 34, 12 (Dec. 2001), 154-157. http://dx.doi.org/10.1109/2.970591
13. Kass, R. and Finin, T. (1988) Modeling the user in natural language systems. Compute. Linguist. 14, 3 (Sep. 1988), 5-22.
14. Kimberly, R. Swinth, Farnham, D. Shelly and Davis, P. (2003). Sharing personal information in online community member profiles. 04-13-2005. http://research.microsoft.com/scg/#papers
15. McCalla G, Vassileva G., Greer J., Bull, S. (2000) Active Learner Modelling, in Gautier, Frasson & VanLehn (eds.) *Proceedings of ITS'2000*, Springer LNCS 1839, 53-62.
16. Merrells J. (2004) XACML: XML Access Control. http://www.idealliance.org/papers/ dx_xmle04/papers/04-01-04/04-01-04.pdf

17. Mohammed, I. and Dilts, D. M. 1994. Design for dynamic user-role-based security. Compute. Secure. 13, 9 (Oct. 1994), 661-671.

18. Niu X. , McCalla G. I., and J. Vassileva (2004) Purpose-based Expert Finding in a Portfolio Management System. *Computational Intelligence Journal*, Vol. 20, No. 4, 548-561.

19. OASIS (2005) eXtensible Access Control Markup Language 2 (XACML) Version 2.0 OASIS Standard, 1 Feb 2005 http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-core-spec-os.pdf visited: Oct 2006

20. Park, J. S., Sandhu, R., and Ahn, G. (2001) Role-based access control on the web. ACM Trans. Inf. Syst. Secur. 4, 1 (Feb. 2001), 37-71. http://doi.acm.org/10.1145/383775.383777

21. Sandhu, R. and Park, J. S. (1998) Decentralized user-role assignment for Web-based intranets. In Proceedings of the Third ACM Workshop on Role-Based Access Control Fairfax, Virginia, United States, October 1998. http://doi.acm.org/10.1145/286884.286887

22. Seth P. (2004). Sun's XACML Implementation. Programmer's guide for version 1.2. 2006 (Aril, 2006).

23. Sun L., Vassileva J. (2006) Social Visualization Encouraging Participation in Online Communities, In Groupware: Design, Implementation, and Use, Proceedings of CRIWG'2006, Springer LNCS 4154, 349-363.

24. Vassileva J., McCalla G., Greer J. (2003) Multi-Agent Multi-User Modeling, *User Modelling and User Adapted Interaction* **13**:(1), 2003, pp. 1-31.

25. Vassileva, J., Greer, J., McCalla, G. (1999) Openness and Disclosure in Multi-Agent Learner Models. Workshop on Open, Interactive, and Other Overt Approaches to Learner Modeling (Proceedings from 9th International Conference, AIED 1999), pages 43-49.

Intelligent Distributed User Modelling: from Semantics to Learning

Bhaskar Mehta and Wolfgang Nejdl

L3S Research Center/ University of Hannover,
Appelstrasse 4, 30167 Hannover, Germany

Abstract. Today, personalization in digital libraries and other information systems occurs separately within each system that one interacts with. However, there are several potential improvements w.r.t. such isolated approaches. Investments of users in personalizing a system either through explicit provision of information or through long and regular use are not transferable to other systems. Moreover, users have little or no control over the information that defines their *profile*, since user profiles are deeply buried in personalization engines. *Cross system personalization*, i.e. personalization that shares personalization information across different system in a user-centric way, overcomes the aforementioned problems. Information about users, which is originally scattered across multiple systems, is combined to obtain maximum leverage. Early approaches to Cross System Personalization were based on understanding user profiles semantically and transferring information between systems about the user. However, the heterogeneity in vocabulary and representation formats have made this approach unpractical; standardization efforts to create a fixed vocabulary for expressing user profiles are required to make this approach viable. Recent approaches to cross system personalization have relied on example profiles of a large number of people using multiple systems; machine learning techniques are then used to learning a mapping between profiling formats of these systems. Thus a new user is able to leverage his/her profiles from other systems to get an instant personalized experience of high quality. In this paper, we outline both of these approaches, pointing out the pros and cons of each of them, and how to use the concept of a decentralized unified user profile which acts as a *Passport* identifying users during their journey in information space.

1 Introduction

The World Wide Web provides access to a wealth of information and services to a huge and heterogeneous user population on a global scale. One important and successful design mechanism in dealing with this diversity of users is to *personalize* Web sites and services, i.e. to customize system contents, characteristics, or appearance with respect to a specific user. Since the last decade, commercial providers have identified personalization as a key driver of business growth and repeat customers, by enabling them to provide services tailored to individuals. This has led to a wide scale deployment of recommender systems which provide scalable personalization, a well known example being Amazon. However, benefiting from these personalized web sites requires both explicit and implicit involvement of the end-users. Each system independently builds up user profiles and may then use this information to personalize the system's content and service offering. Today, users often use multiple electronic system offering recommendations, which cannot learn from one another. The end result is that a user has to go through multiple training phases with different systems, often providing similar information and in some cases disjoint information to different isolated systems.

Such isolated approaches have two major drawbacks: firstly, investments of users in personalizing a system either through explicit provision of information or through long and regular use are not transferable to other systems. Secondly, users have little or no control over the information that defines their profile, since user data are deeply buried in personalization engines running on the server side. Intuitively, it seems that much can be improved with this situation: information learnt by one system could potentially be reused by another, and new insights learnt by a third system about the same user can be propagated to other systems, to offer an overall improved personalization experience. It can be expected that such a solution will greatly benefit new users(who use other personalized systems) and provide better recommendations than a non personalized '*most popular*' approach, a problem known as the '*cold start*' problem in literature.

Cross system personalization(CSP) [7] allows for sharing information across different information systems in a user-centric way and can overcome the aforementioned problems. Information about users, which is originally scattered across multiple systems, is combined to obtain maximum leverage and reuse of information. Previous approaches to cross system personalization[9] relied on each user having a *unified profile* which different systems can understand. The basis of 'understanding' in this approach is of a semantic nature, i.e. a user profile can be semantically interpreted by another system. The main challenge in this approach is to establish some common and globally accepted vocabulary and to create a standard every system will comply with.

Machine learning techniques provide a promising alternative by using example data to learn mappings between profile formats to enable cross system personalization without the need to rely on accepted semantic standards or ontologies. The key idea is that one can try to learn dependencies between profiles maintained within one system and profiles maintained within a second system based on data provided by users who use both systems and who are willing to share their profiles across systems – which we assume is in the interest of the user. Here, instead of requiring a common semantic framework, it is only required that a sufficient number of users cross between systems and that there is enough regularity among users that one can learn within a user population, a fact that is commonly exploited in social or *collaborative filtering*.

In this paper, we outline both of these approaches, pointing out the pros and cons of each of them, and how to use the concept of a decentralized unified user profile which acts as a *Passport* identifying users during their journey in information space. We also provide experimental results for learning based methods on collaborative filtering data, thus showing the measurable advantage that cross system personalization provides.

2 The UUCM user model for Unified User Modeling

The UUCM is a user context model that is structured along different dimensions and captures the fact that the user interacts with systems in different working contexts by structuring the model accordingly. In order to support cross-system personalization, the model has to be flexible and extensible enough to deal with the variations in personalization approaches and to incorporate the various aspects relevant for capturing the users' characteristics and his current situation. The Context Passport is an aggregated user profile in the UUCM format which can be used with multiple systems

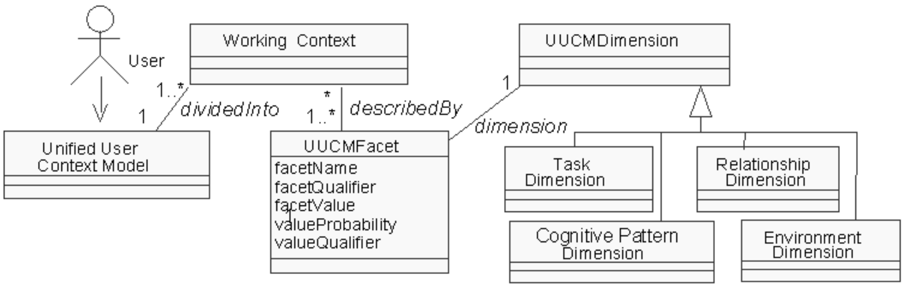

Fig. 1. Building Blocks of the UUCM

which models various facets of a user. As shown in Fig. 2, information systems can still maintain their own user profiles, but these profiles are synchronized with the Context Passport. This section describes the UUCM model.

2.1 UUCM as a Meta-Model

The main building blocks for the UUCM is an extensible set of facets(see Figure 1). The facets represent the different characteristics of the user and his current context. We use the term facets instead of properties, because we do not only capture attribute value pairs but also probabilities and qualifiers for facet values, thus giving a richer description as it is typical for frame-based languages. An extensible set of UUCM dimensions enables the structuring of the facets into groups of user characteristics (e.g. facets related to cognitive pattern).

In the context of UUCM, qualification of names as well as values of the facets is a crucial aspect. Both names and values may refer to vocabularies or ontologies, giving the possibility to connect to shared vocabularies, thus simplifying interpretation in a global (cross-system personalization) context. In summary, each UUCM facet is described by the following properties, part of which are optional:

- *Facet name* is the Name of the UUCM facet to be described;
- *Facet qualifier* is used to bind the facet itself to a defining facet vocabulary or ontology;
- *Facet value* is the value of the facet, which can be a simple literal as well a reference to a structured resource depending on the domain.
- *Value qualifier* is a qualifier for the value(s) of the facet, i.e. it points to the vocabulary or ontology the value is taken from;
- *Value probability*: A weight reflecting the probability of the facet value.
- *Facet dimension*: Each facet is assigned to one of the UUCM dimensions

The UUCM defines a sort of a meta-model for the concrete dimensions and facets used in the description of a 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. In support of the UUCM, other ontologies are required: a *Facet ontology* that defines the different facets, a *dimension ontology* that defines facet dimensions and, optionally, also ontologies or vocabularies for the facet values. The UUCM meta-model thus, can be combined with UUCM facet and dimension ontologies to form concrete user context models that provide the schema for the construction of user context profiles.

The UUCM is encoded as an RDF Schema augmented with OWL expressions. This technology, enables simple exchange within the (Semantic) Web context, reasoning over user characteristics for value adding services and URIs provide a systematic support for the qualification of facets and facet values. 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.

2.2 Example of UUCM Facets and Dimensions

Although the set of the UUCM facets and UUCM dimensions is not predefined and extensible, we identified a set of four central dimensions together with important and representative facets within these dimensions to illustrate our approach and to give a starting point for the definition of concrete user context models:

Cognitive Pattern dimension : It describes cognitive characteristics of the user, as they are traditionally used in personalization approaches. Example facets in this area are 1) *areas-of-interest*, describing the interests of a user typically based on a controlled vocabulary or ontology or subjects; 2) *competence*, which can be divided into two facet subclasses *skill* and *expertise*, and 3) *preference*, that can be used to model user preferences.

Task dimension : The task dimension of the UUCM may contain facets like 1) *current task*, which describes the task the user is currently involved in, and 2) *task role* describing the role of the user in the current task. Furthermore, the user context model may also contain a facet *task history.*

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 facet names are 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.

Environment dimension : It refers to those parameters, which are typically used for context-awareness approaches. Facets like current *time*, *location*(physical location of user), *device* (e.g. PC, PDA, etc.), *language* chosen by the user, etc. are parameters, which influence the interaction between the user and the computer in a specific setting or situation. Many other facets describing the environment might be important depending upon the specific application.

2.3 UUCM and Cross System Personalization

There are three objectives which cross-system personalization needs to address. These are a) broader user models that can cope with the variety in user modeling, b) handling heterogeneous personalization systems and approaches, and c) giving more control to the user. In line with these objectives, we claim that user profiles should be stored

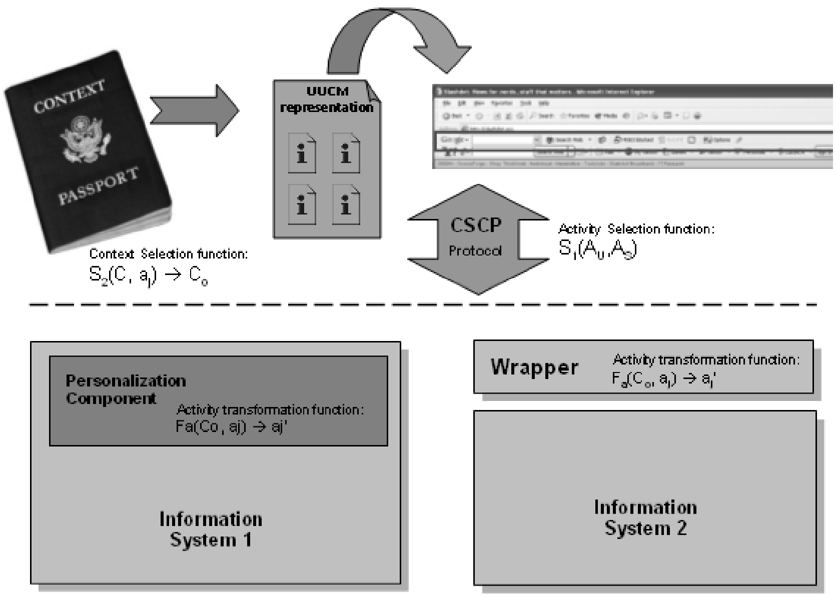

Fig. 2. High Level Architecture using Context Passport and CSCP

closer to the user. In [10], we introduced the concept of a *Context Passport* which accompanies the user in his/her travel through information space.

However, maintaining user profiles on the user's side presents some challenges. Interacting with multiple information systems may lead to a large amount of interaction data. The first step of reducing this amount is interpretation of this data to extract user characteristics from it. Since the individual system best understands the local interactions this should be done within the individual personalization engine and only higher level descriptions of users should be exchanged between the information system and the unified user profile, which we call the *Context Passport*.

In a given situation, not all of the user context information is relevant [10]. For e.g., in a user's private context where she is looking to travel to Vienna, the user context related to official work, or shopping plans for next week may not be relevant. Therefore the Context Passport has to be an active component, which not only stores user context, but also extracts the relevant portion of the user context, called *Context-of-Use*. This relevant past is influenced by the current activity, as well as by what part of the user context the interacting system can understand.

As described in our Box model [10] underlying the cross-system personalization approach, there are three steps to use user context for personalization: Activity Selection, Context Selection and Activity Transformation (Figure 2). We define a personalized activity based on this formalization: *Given an activity a_i, and including user context information C', a_i gets transformed into a personalized activity a_i'.* The nature of the transformation depends on the system for which the personalization is done, i.e. different information systems could use the same information about the user to give different kinds of personalized access to information. If a user's current goal is to travel to Vienna, a travel system could offer recommendations on flight tickets, while a hotel site could offer hotel bookings. Clearly the exchange of such information requires a negotiation between activities that an information system(IS) can perform and those activities that the user context outlines. The *Cross System Communication*

Protocol (CSCP) provides a platform for such negotiations. In our current testbed, the IS and the Context Passport (which is a browser toolbar) communicate via the CSCP. This communication is client driven and the IS exposes a set of stateful Web Services which allows the negotiation to run in a stateful manner (like SMTP).

2.4 Drawbacks of Semantic Cross System Personalization

Clearly, profile formats in different systems tends to be different in how they model a user. For multiple systems to be able to interoperate, a common vocabulary has to be used which requires a standardization process. This process might define a controlled vocabulary e.g. default facet names, or the range of values a facet can take. While individual systems might still define their own facets, they would be required to make mapping available to existing facets. Another drawback of this method is misinterpretation of a facet to represent data about the user which is different from other systems. (e.g. a movie provider using the facet "comedy" for storing all movie preferences. Clearly, this could lead to false information becoming a part of the user profile. Using an aggregate of user profiles can be used to sort out such inconsistencies automatically: this is the key idea in the next section which introduces CSP based on machine learning.

3 Automatic Cross System Personalization

A drawback of traditional electronic systems is their inability to cope sensibly with new or unexpected situations, leading to sudden crashes or unexpected outcome. Anticipating every possible scenario and defensively programming computer systems is possible only in restricted scenarios. Clearly, to operate autonomously in a real world setting, electronic systems have a key requirement to learn from new situations and adapt accordingly. The field of *Machine Learning* has evolved from this requirement in the Artificial Intelligence community. In this field, one considers the important question of how to make machines able to learn. Learning in this context can be of different types, one of which is *inductive inference*, where one observes examples that represent samples of some statistical phenomenon. In *unsupervised* learning one typically tries to discover inconsistencies, anomalies in observed data, similar conceptually to data mining. In *supervised learning*, one typically has input and output data of a given sample of observations, where one tries to infer functions which map the input to output with minimum error. We model the problem of Cross System personalization in the framework of supervised (and later semi-supervised) learning to learn mappings between two different systems.

3.1 Learning Mapping between profiles

For simplicity, we consider a two system scenario in which there are only two sites or systems denoted by A and B that perform some sort of personalization and maintain separate profiles of their users; generalization to an arbitrary number of systems is relatively straightforward and is discussed later. We assume that there is a certain number of N_c common users that are known to both systems. For simplification, we assume that the user profiles for a user u_i are stored as vectors $\mathbf{x}_i \in \mathcal{X} \subseteq \mathbb{R}^n$ and $\mathbf{y}_i \in \mathcal{Y} \subseteq \mathbb{R}^m$ for systems A and B, respectively. Given the profile x_i of a user in system A, the objective is to find the profile y_i of the same user in system B, so formally we are looking to find a mapping

$$F_{AB} : \mathbb{R}^n \rightarrow \mathbb{R}^m, \qquad \text{s.t.} \quad F_{AB}(\mathbf{x}_i) \approx \mathbf{y}_i \tag{1}$$

for users u_i. Notice that if users exist for which profiles in both system are known, i.e. a training set $\{(\mathbf{x}_i, \mathbf{y}_i) : i = 1, \ldots, l\}$, then this amounts to a standard supervised learning problem. However, regression problems typically only involve a single (real-valued) response variable, whereas here the function F_{AB} that needs to be learned is *vector-valued*. In fact, if profiles store say rating information about products or items at a site, then the dimensionality of the output can be significant (e.g. in the tens of thousands). Moreover, notice that we expect the outputs to be highly correlated in such a case, a crucial fact that is exploited by recommender systems. For computational reasons it is inefficient and often impractical to learn independent regression functions for each profile component. Moreover, ignoring inter-dependencies can seriously deteriorate the prediction accuracy that is possible when taking such correlations into account. Lastly, one also has to expect that a large fraction of users are only known to one system (either A or B). This brings up the question of how to exploit data without known correspondence in a principled manner, a problem generally referred to as *semi-supervised learning*. Notice that the situation is symmetric and that unlabeled data may be available for both systems, i.e. sets of vectors $\mathbf{x}_i$ without corresponding $\mathbf{y}_i$ and vice versa. In summary, we have three conceptual requirements roe a machine learning method:

- Perform vector-valued regression *en bloc* and not independently
- Exploit correlations between different output dimensions (or response variables)
- Utilize data without known correspondences

In addition, the nature of the envisioned application requires:

- Scalability of the method to large user populations and many systems/sites
- Capability to deal with missing and incomplete data

There are some recent learning methods that can be utilized for vector-valued regression problem, but some of them do not fulfill the above requirements. Kernel dependency estimation [12] (KDE) is a technique that performs kernel PCA [11] on the output side and then learns independent regression functions from inputs to the PCA-space. However, KDE can only deal with unlabeled data on the output side and requires to solve computationally demanding pre-image problems for prediction [1]. Another option is Gaussian process regression with coupled outputs [6]. Here it is again difficult to take unlabeled data into account while preserving the computational efficiency of the procedure. The same is true for more traditional approaches like Multi-Layer-Perceptrons with multiple outputs. Instead of using regression methods, we used *manifold learning* in this context[7]. The key idea is to embed user profiles from different systems in low-dimensional manifolds such that profiles known to be in correspondence (i.e. profiles of the same user) are mapped to the same point. This means the manifolds will be aligned at correspondence points. A more general version of CLLE has been derived in [5], which takes the Laplacian eigenmap approach [2] as the starting point.

In addition, we have also proposed the use of linear dimensionality reduction methods for the CSP task, which have shown better performance. [8] describes how sparse factor analysis was used to learn a mapping between data points representing user profiles in two systems. Other alternative methods include PLSA, which we found to perform better as a predictive model. In this work, we will provide a comparative analysis between the relative performance of these methods.

3.2 Cross System Personalization as a matrix completion problem

Two basic assumptions help us in casting the CSP task as a missing value problem: *first*, that users have their profiles for multiple systems available to them, and *second*, that users are willing to provide their multiple profiles for computing a mapping between the profile formats of these systems. In this section, we use these basic assumptions to cast CSP as a missing value problem.

In a two system scenario, we have two sites A and B, containing user profiles for their users represented as vectors. A user u_i has a profile $\mathbf{x}_i^A \in \mathbb{R}^m$ at site A, and a profile $\mathbf{x}_i^B \in \mathbb{R}^p$ at site B. Let matrices $\mathbf{X}^A$ and $\mathbf{X}^B$ represent all user profiles (with each user representing a column) at systems A and B and further assume that c users are common to both sites and that the data matrices can be partitioned as:

$$\mathbf{X}^A = \left[\mathbf{X}_c^A \; \mathbf{X}_s^A\right], \qquad \mathbf{X}^B = \left[\mathbf{X}_c^B \; \mathbf{X}_s^B\right], \tag{2}$$

where $\mathbf{X}_c^A$ and $\mathbf{X}_c^B$ represent the sub-matrices of $\mathbf{X}^A$ and $\mathbf{X}^B$ corresponding to the common users and $\mathbf{X}_s^A$ and $\mathbf{X}_s^B$ the sub-matrices for users that are unique to A and B.

One way of looking at the CSP problem in the context of latent semantics is to relate the profiles in both (or multiple) systems by assuming that the user profiles are likely to be consistent in terms of the basic factors, i.e. that they can be explained by latent factors common to both systems.

A simple manner of enforcing this constraint is to construct a new combined vector $\mathbf{x} = [\mathbf{x}^A \; \mathbf{x}^B]$ and to perform a joint analysis over the combined profile space of system A and B. This means we effectively generate a data matrix

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}_c^A & \mathbf{X}_s^A & ? \\ \mathbf{X}_c^B & ? & \mathbf{X}_s^B \end{bmatrix}, \tag{3}$$

where '?' denotes matrices of appropriate size with unobserved values. Note that the other submatrices of $\mathbf{X}$ may also contain (many) missing entries. Also note that a column of this matrix $\mathbf{X}$ effectively contains a unified user profile of the user across two systems, adding more systems can be done in a similar fashion. It is interesting to make a further simplification by restricting the data matrix to users that are known to both systems

$$\mathbf{X}_c = \left[\mathbf{X}_c^A \;\; \mathbf{X}_c^B\right], \tag{4}$$

and to ignore the data concerning users only known to one system. Obviously, this will accelerate the model fitting compared to working with the full matrix $\mathbf{X}$. This situation corresponds to a *supervised learning* setting where labeled output data is available for all training samples. However, it is likely to be less accurate than the *semi-supervised learning* setting where $\mathbf{X}$ is used, since the unlabeled samples will potentially improve the estimation of the parameters.

Related Work Recently, a few techniques have been suggested for the purposes for Cross System Personalization which deal with the vector valued learning problem that CSP entails. The earliest technique for CSP is Manifold Alignment [7], which performs satisfactorily in the test scenarios evaluated, but does not deal well with incomplete data.Manifold alignment uses non linear dimensionality reduction techniques like Locally Linear Embedding(LLE) and Laplacian Eigenmaps, which have previously not been applied to Collaborative filtering, and do not scale very well. The next technique to emerge is using Sparse Factor Analysis(SFA)[4, 8], which performs very well even on large datasets. SFA was originally proposed by [4] in the context of privacy preserving collaborative filtering. The advantage of PLSA over the above methods is better performance for the collaborative filtering domain, and ease of updation in case new items are added. While SFA is a fast method for collaborative filtering, it does not offer any easy mechanism to add new items without a complete re-computation of the model.

4 Evaluation

Our evaluation is aimed at testing the following hypotheses:

1. CSP offers an advantage over mean item voting for a large number of first time users,
2. CSP using *PLSA* offers an advantage over existing methods like SparseFA.

We have implemented SFA and PLSA in Matlab and Java, and have run them on the same data to provide a comparison. Clearly, Semantic CSP cannot be evaluated in this environment since semantic user profiles are not readily available, and have to be extracted. The error introduced in the extraction process makes isolation of the effectiveness of semantic methods very hard to evaluate. For the rest of the evaluation, we focus on SFA and PLSA as solutions to the CSP task. We choose the EachMovie[1] data with ratings from 72,916 users for 1,682 movies. Ratings are given on a numeric six point scale (0.0, 0.2, 0.4, 0.6, 0.8, 1.0). The entire dataset consists of around 2.8 million votes, however around 1.8 million of these votes are by the first 21835 users. We chose this dense subset of 21835 users and 1682 movies and we scale these ratings on a scale of 5, which is also the scale in other datasets like *MovieLens* and lately *Netflix*.

To simulate two systems A and B, we divide this data set into two parts by splitting the item set of the entire data. This way we get two datasets with the same number of users, but with ratings over different items. In our experiments, we have used 15,000 users for both A and B, with 8,000 users being common between the two systems. To mimic a real life setting, we allow a random 5% of items to overlap between the datasets. The overlap is not explicitly maintained. In our test runs, we build a PLSA model using the matrix $\mathbf{X}$ (see eq. (2)) varying c from 1000 users to 8000 users. For the users not in correspondance, we randomly rearrange their order. We refer to this case as the *full* data case. In our setting, it is vital that we can build an effective predictive model with as few users crossing over from one system to another which works effectively for a large number of new users. We use 5000 users as test (randomly from the 7000 users not common to the systems). In addition, we also performed the model building step using only the users common to both systems using $\mathbf{X}_c$ (see eq. (3)). We refer to this case as the *common* data case.

Metrics used

1. Mean Average Error $= \frac{1}{m} \sum |p_v - a_v|$, where p_v is the predicted vote and a_v is the actual vote. The average is taken only over known values (assume m votes in test set).

[1] http://research.compaq.com/SRC/eachmovie

2. Ranking score of top-20 items. $R_{score} = \frac{100 * \sum R}{\sum R_{max}}$. This metric gives a values between 0 and 100. Higher values indicate a ranking with top items as the most highly rated ones. One big advantage of this metric is that it gives the same score for permutations of items with the same score. This removes the problem of breaking ties.(see [3] for details)

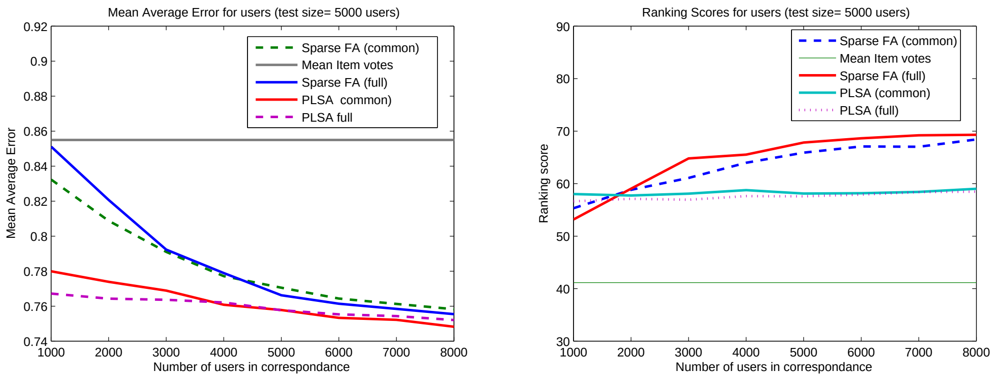

Fig. 3. MAE and Ranking scores for 5000 test users (with 5 fold validation). ”‘*common*”′ refers to the use of only common users (eq. 3) for training the model.

Results

The experimental bench described above sets the scene: PLSA models and SparseFA models are trained over identical datasets, and MAE and Ranking Scores are measured. Results are then averaged over 5 runs and plotted in Figure 3. For the SparseFA model training, we use an improved implementation (w.r.t [8]) which is optimized w.r.t. model parameters and reports better results than previously. SparseFA remains a fast and effective model; however, we expect PLSA to outperform SparseFA.

Figure 3 provides experimental proof: PLSA has a distinct advantage with smaller training data and provides highly accurate recommendations for 5000 test users *even* when only 1000 users have crossed over. While SparseFA also outperforms the baseline *most popular*[2] method, it catches up with PLSA only after more than 7000 users have crossed over: even then PLSA maintains a slight lead. The results in the ranking score experiment show an advantage for Sparse FA over PLSA: this means that while PLSA is an overall more accurate method, Sparse FA is able to pick the top 20 relevant items and rank them better than PLSA. A lower *Mean Average Error* for PLSA shows that the complete profile predicted by PLSA is closer to the original profile than the one predicted by SparseFA. One more important observation is that the models trained with only common data(supervised) outperform the models trained with full data (semi-supervised). However, this trend is observable only when a small number of users are common to both systems. Once around 4000 users have crossed over, the semi supervised methods have a small lead. In a practical situation, we might use only the common users, since the overhead of training this model is much smaller than the full data.

[2] The most popular strategy recommends the most highly rated(on an average) items .

5 Conclusion

This paper outlines a novel approach to leverage user data distributed across various electronic systems to provide a better personalization experience. One major benefit of this approach is dealing with the new user problem: a new user of a collaborative filtering system can usually be provided only the non-personalized recommendation based on popular items. Our approach allows systems to make better recommendations using the user's profile in other systems. We provide an overview of all methods used to solve this problem so far, and also show experimental results for the best performing methods. Other articles provide more details about the exact methods focusing on more intricate details; this paper focuses on the journey so far in modeling users for decentralized recommendation and the challenges experienced so far. Future work involves creating practical frameworks around the solutions involving machine learning, and also establishing data sets where multi domain user profiles belonging to the same set of people is available.

Acknowledgments: The authors would like to thank Thomas Hofmann and Peter Fankhauser for various discussions fundamental to this work.

References

1. G. Bakir, J. Weston, and B. Scholkopf. Learning to find pre-images. In L. S. Thrun, S. and B. Scholkopf, editors, *Advances in Neural Information Processing Systems*, volume 16, pages 449–456, Cambridge, MA, USA, 2004. MIT Press.
2. M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003.
3. J. Breese, D. Heckerman, and C. M. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *UAI*, pages 43–52, 1998.
4. J. F. Canny. Collaborative filtering with privacy via factor analysis. In *SIGIR*, pages 238–245, 2002.
5. J. Ham, D. Lee, and L. Saul. Semisupervised alignment of manifolds. In R. G. Cowell and Z. Ghahramani, editors, *AISTATS 2005*, pages 120–127. Society for Artificial Intelligence and Statistics, 2005.
6. S. Keerthi and W. Chu. A matching pursuit approach to sparse gaussian process regression. In Y. Weiss, B. Scholkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems 18*. MIT Press, Cambridge, MA, 2006.
7. B. Mehta and T. Hofmann. Cross system personalization by learning manifold alignment (under submission), 2006.
8. B. Mehta, T. Hofmann, and P. Fankhauser. Cross system personalization by sparse factor analysis. *ITWP Workshop at AAAI.*, 2006.
9. B. Mehta, C. Niederée, A. Stewart, M. Degemmis, P. Lops, and G. Semeraro. Ontologically-enriched unified user modeling for cross-system personalization. In *User Modeling*, pages 119–123, 2005.
10. C. J. Niederée, A. Stewart, B. Mehta, and M. Hemmje. A multi-dimensional, unified user model for cross-system personalization. In *Proceedings of Advanced Visual Interfaces International Working Conference (AVI 2004) - Workshop on Environments for Personalized Information Access, Gallipoli, Italy*, 2004.
11. B. Scholkopf, A. J. Smola, and K.-R. Muller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319, 1998.
12. J. Weston, O. Chapelle, A. Elisseeff, B. Scholkopf, and V. Vapnik. Kernel dependency estimation. In *NIPS*, pages 873–880, 2002.

A preliminary step toward user model interoperability in the adaptive social web*

Francesca Carmagnola, Federica Cena, Omar Cortassa, Cristina Gena, and Andrea Toso

Dipartimento di Informatica, Università di Torino
Corso Svizzera 185, Torino, Italy
{carmagnola, cena, cortassa, cgena}@di.unito.it; toso@csp.it

Abstract. This paper presents an adaptive social web site, iCITY, and its tag-based user model which can be filled with the tags the user has already exploited in other social web sites in order to learn the user's interests and to avoid the cold start problem. Moreover, the iCITY tag-based user model can be exported and shared with other social applications in a semantic enhanced way. Finally, we propose that systems publish also the user profile together with the list of tags (with the elaboration over them) in a shared common syntax (like RDF(S), OWL, RSS).

1 Introduction

Nowadays, we are assisting to a big transformation of the Web as we are used to conceived. We can refer to it as a new paradigm, the Web 2.0 [7], a label coined by O'Reilly Media in 2004, to address a new generation of Web-based services such as social networking sites, social software [6], wikis, communication tools, weblogs, postcasts, RSS feeds (and other forms of many-to-many publishing) and folksonomies [4]- that emphasize online collaboration among users. This new paradigm offers users several ways to easily participate in the creation of web contents: it makes easy and stimulates the process of tagging (labelling resources by means of keywords), inserting new contents, sharing objects, providing comments and so on. In this sense, the most popular Web tools are Del.icio.us[1], which allows to tag and share bookmarks, Flickr[2] which allows to store, share and retrieve photos, Digg[3] which allows to share news, and Youtube[4] which lets to partake videos.

The "one-size-fit-all" approach is completely useless to satisfy user needs in this novel social environment. Users are no more passive: they are active and demanding, and they require they needs to be personally satisfied. Thus, personalization, already considered

* This work has been funded by the the Municipality of Torino and the CSP research center. We want to thank Franco Carcillo, Luca Console, Fabiana Vernero, Anna Goy and Ilaria Torre for their support.

1 http://del.icio.us/
2 http://www.flickr.com/
3 http://www.digg.com/
4 http://www.youtube.com/

crucial in many areas (from e-commerce to e-learning, from tourism and cultural heritage to digital libraries and so on) is required in the social web as well.

With the proliferation of web-based user-adaptive systems [5], a user can interact with different systems that collect data about her.
The result is that user data are fragmented over many applications on the Web and the user profile is inherently distributed and often not consistent. User-adaptive applications build a model of the user starting from personal information (e.g. characteristics, preferences, knowledge, interests, goals, activities). Such information can be explicitly provided by the user herself, or implicitly inferred by her interactions with the system. With the coming of the Web 2.0, systems have the chance to collect also the tags the user uses to label items, and learn from them. In this way, systems may exploit tag annotation in order to enrich and extend the user model [8], [1] “Annotations can become part of his user profile as an indication of his perspective on the content collection and interest in the annotated object” [8]. Moreover systems may use this ”tag-enriched” profile for recommendations, using techniques as collaborative filtering or case based reasoning. It is also interesting to semantically analyze tags and reason on them in order to infer knowledge about a specific user. Thus, we propose to map all these tags to some ontologies; in this way, systems may reason on the semantics on the tags, especially in relation to the user who has inserted them [2].
In this direction, what we present in this paper is the idea to give systems the possibility to exchange the tags (and the related information) together with the profile of the user who has inserted them. The advantages of this solution are mainly related to the possibility for social systems who want to offer personalized service to overcame the “cold start problem” which may occur at the beginning of the interaction with a new user, when no information about her is available. If systems exchange the user profile and/or the list of tags the specific user has inserted, they do not need to wait for a big amount of tagging activities from the user, and at the same time they avoid bothering the user asking for some personal information during the registration phase. Some systems already make the list of tags available in some xml-based syntax (like RDF, OWL).

The paper is structured as follows: Sec. 2 presents a possibile scenario in the social adaptive web, Sec. 3 illustrates our proposed architecture for interoperability of user models and tags ,while Sec. 4 concludes the paper.

2 Scenario

iCITY[5] is a social web-based and multi-device recommender system. It provides suggestions on cultural events in the city of Turin, and allows users to insert new events, to add information about events, to insert comments and tags. Recommendations are based on the user model enriched with tags, and on the user location, and the presentation interface is adapted to the device being used. A general description of iCITY can be found in [1], while for a description of the tag-based user model see [2].

[5] A work in progress prototype can be found at http://icity.di.unito.it/dsa/

Now imagine a scenario where the user Cristina - which regularly uses social softwares such as Del.icio.us (to collect and share bookmarks) and Flickr (to store, share and retrieve photos) - decides to register in iCITY in order to be always aware of the events in Turin. Furthermore she can store and tag the events in the iCITY agenda, and retrieve them on her GPRS equipped mobile phone when she is around. During the registration phase she inserts, besides some other personal information, her accounts (express trough URLs)[6] of the web communities she belongs to. Such an operation is due to the fact that she discovers that iCITY can learn her interests and preferences by reading the tags she has inserted in other web community tools. Moreover, the iCITY also monitors and reasons about her direct interactions with the system itself and about the tags she inserts. She believes very useful to be supported by iCITY in finding more interesting events. She especially appreciate the fact that she can always check the assumptions about her interests and preferences in her scrutable user profile and eventually correct them.

3 An Architecture for user model interoperability in the adaptive social web

According to our approach, what a web community application may wish to do with users' tags is threefold: i) to extract them regardless of the format used to represent them; ii) to reason upon them in order to enrich the user's knowledge (especially regarding interests and preferences); iii) to make such tags available to other web applications in order to achieve an effective exchange of knowledge. In order to perform the three tasks, we are developing a modular architecture. The main components of such architecture are the Importer Module and the Exporter Module, which have respectively the assignments of extracting the tags and making them available to other applications. Let's more deeply analyze the tasks of each component, with the aid of Figure 1.
Once the user provides her web community accounts, the Importer Module retrieves the corresponding files containing the set of tags exploited by the user in the web community tools she has pointed out. Then the Importer Module extracts the tags, independently by the format used to represent them. For instance, Imp1 has been set up to retrieve tags from Del.icio.us which uses the following RDF markup:

```
 <item rdf:about="http://www.kiki.it"/>
 ...
 <taxo:topics>
    <rdf:Bag>
    <rdf:li resource="http://del.icio.us/tag/japanese" />
    </rdf:Bag>
</taxo:topics>
```

While Imp2 has been set up to retrieve tags from Flickr uses the following XML markup:

[6] http://del.icio.us/cristinagena; http://www.flickr.com/photos/alfia1973

```
<media:category scheme="urn:flickr:tags" >
    naples ravello church
</media:category >
```

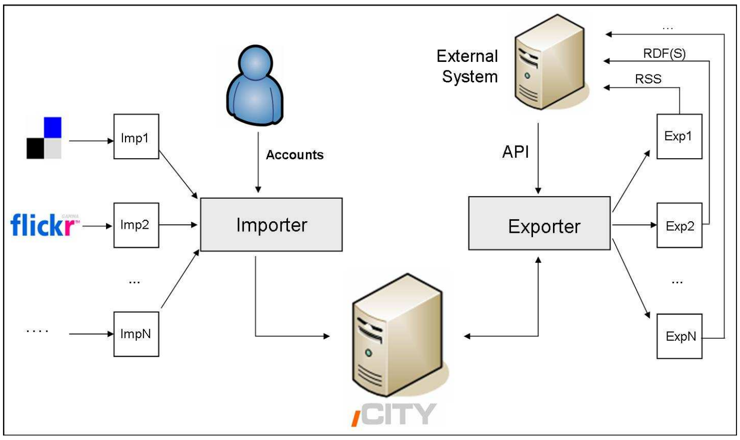

Fig. 1. Architecture of Importer and Exporter Modules

Once all the user's tags have been extracted, the Importer Module looks for correspondence in the Event Ontology of iCITY, which is a RDFS ontology defined starting from the classification of events in *TorinoCultura*[7], a web portal managed by the municipality of Torino for informing citizens about cultural ongoing events in the city, and enriched with WordNet[8]. In particular, classes and subclasses of the ontology are mapped on WordNet concepts by means of the WordNetTab extension for Protege[9], in order to automatically map the tag inserted by the user into the event ontology. Thus iCITY reasons over the tag to enhance the knowledge about user's interests and preferences (for a more detailed description of this mechanism see [2]).

The Importer Module is in fact able to find a correspondence between the extracted user's tags and the WordNet terms mapped on the ontology. Thus the system increases the level of inferred user's interests related to the class the tag belongs to (notice that the iCITY user model is designed as an overlay model of the Event Ontology). For instance if Cristina has often used the tag "music" to annotate her bookmarks in Del.icio.us, iCITY infers that she likes music and propose her musical events. However, Cristina could also either uses personal tags that don't have any correspondence with the ontology (e.g., "researchLinks", "personalLinks"), or terms related to a different concepts. For instance, she could use the tag "Japanese" to annotate the web sites of her preferred Japanese restaurant, but she could not like Japanese music.

[7] http://www.torinocultura.it/
[8] http://wordnet.princeton.edu/
[9] http://protege.stanford.edu/

To better understand the semantics of the tags, the Importer Module is designed to analyze the HTML keyword meta-tags of web site the tagged bookmark is related to (as done by De.li.cious). However, either the module could not understand the meta-tags for some reason or there is the risk that no HTML meta-tags are available (i.e. in Flickr). To overcome such problems and let other systems better understand the iCITY public tag-based user profile, we are defining in iCITY an Exporter Module which generates a RSS file containing RDF statements about the tags the user inserts into the system. In such file each tag refers to the class of the public ontology it belongs to (the tag is referred to an event which is automatically mapped on the ontology).
Thus, the other web-based systems that would explore the tags a user used in iCITY system can find a semantic hints for the understanding and the disambiguation of the concepts.

4 Conclusion and future works

Several web community tools provide users with the chance of inserting tags. In our opinion, tags are really useful in revealing users' interests and preferences; thus user-adaptive systems may benefit from them to enrich their user models. This paper has presented a preliminary proposal which shows how users' tag could be exchanged over the web to share and re-use the user interests that are fragmented in the distributed user models of social web applications.
We are investigating the possibility of sharing and exporting - using the same architecture and a shared format (e.g., RSS, RDF(S), OWL, UserML) - the complete user models (or a portion of it) together with the users' tags. In this way the user could save her user model and voluntary submit her profile to adaptive social applications able to parse and understand such format. Thus the user is no more a passive entity, but she has the power to control the information systems has stored about her. In this direction, the need of scrutable user model became crucial [3].

References

1. Francesca Carmagnola, Federica Cena, Luca Console, Omar Cortassa, Marta Ferri, Cristina Gena, Anna Goy, Mario Parena, Ilaria Torre, Andrea Toso, Fabiana Vernero, and Agnese Vellar. icity an adaptive social mobile guide for cultural events. *Proceedings of Mobile Guide 2006, http://mobileguide06.di.unito.it/*, 2006.
2. Francesca Carmagnola, Federica Cena, Omar Cortassa, Cristina Gena, and Ilaria Torre. Towards a tag-based user model: how can user model benefit from tags? *To appear in the Proceedings of UM2007*, 2007.
3. Kay J, Kummerfeld R J, and Lauder P. Foundations for personalised documents: a scrutable user model server. In *Proceedings of ADCS'2001, Australian Document Computing Symposium*, pages 43–50.
4. Guy M and Tonkin E. Folksonomie. tidying up with tags. *D-Lib Magazine*, 12(1):2, January 2006.
5. Maybury M. and Brusilovsky P. *The Adaptive Web*, volume 45. Communications of the ACM, 2002.
6. Tepper M. The rise of social software, networker. *ACM Press*, 7(3), 2003.

7. O’Reilly T. What is web 2.0. design patterns and business models for the next generation of software. 30/09/05.
8. van Setten M, Brussee R, van Vliet H, Gazendam L, van H uten Y, and Veenstra M. On the importance of ”who tagged what”. Dublin, Irland, June 20th 2006. Workshop on Social Navigation and Community-Based Adaptation Technologies (AH 06).

An Agent-Based Architecture for Museum Visitors' Guide Systems

Tsvi Kuflik [1,2],Adriano Albertini[1], Paolo Busetta[1], Cesare Rocchi[1], Oliviero Stock[1], Massimo Zancanaro[1]

[1]ITC-irst, Istituto per la Ricerca Scientifica e Tecnologica,
Povo, Trento, Italy
[2]Department of Management Information Systems
University of Haifa, Haifa, Israel
[1]{kuflik, albertini, busetta, rocchi, stock, zancana}@itc.it
[2]tsvikak@is.haifa.ac.il

Abstract. Recent developments in museum visitors' guides focus on context awareness, personalization and multimodal and multimedia information presentation to individuals and groups of visitors. However, the modern museum is becoming an "Active Museum", which is a special example of an active environment that interacts with its inhabitants. Since recent museum visitors' guides have focused more on the application and less on the system architecture and infrastructure, much effort is now being invested in the preparation of infrastructure that will support the specific research application. This work focuses on the architecture of the "Active Museum" as demonstrated by two research projects on museum visitors' guides, and suggests a generic, layered architecture for such systems. Such architecture would facilitate research cooperation and increase its effectiveness, and also serve later as a basis for the development of museum visitors' guides.

1. Introduction

Interactive multimedia is a promising medium for use in heritage attractions and museums as a complement to existing interpretative techniques. Interactive multimedia exhibits are excellent interpreters in that they can communicate large amounts of often complex information in a user-friendly and interesting way, whilst allowing visitors to access the information they require at their own pace (Alison and Gwaltney, 1991). Interactive multimedia presentations are not a replacement for traditional interpretation methods, but rather complement them, and can assist the visitor by placing exhibits and artifacts in their historical and cultural context. They appear to be universally popular with visitors (Economou, 1997).

Audio Guides nowadays are very popular as they free visitors from reading. A recent and sophisticated one is the Discovery Point, which is a small remote control-like device that allows users to hear short stories related to the work of art; it is in use at the Carnegie Museum of Art in Pittsburgh (Berkovich et al., 2003). The Discovery Point prototype is a headset-less audio system consisting of the physical device that the visitor holds and special speakers which deliver pinpointed audio that can only be heard near the work of art. The portability of the guidebook and audio guides could be effectively combined with the level of interaction and media richness of CD-ROM and interactive multimedia kiosks. For instance in "GEMISIS 2000", developed at the University of Salford, the objects (textile machines) exhibited in the museum gallery are illustrated and navigation is being effected via a touch screen. Each element on the screen acts as a hot-spot, which when touched, links to supplementary information on the chosen topic. The main navigation into each room of the museum is represented by buttons' icons at the bottom of the screen. (Evans and Sterry, 1999).

Recently, portable computers and Personal Digital Assistants (PDA's) have been introduced into museums. These devices are also being used to support visitors to cities much as does the GUIDE system, which supports visitors to the city of Lancaster (Cheverst 2000). Another example of PDA application can be seen at Genoa's Costa Aquarium (Bellotti, 2002). The interface's basic element is a multimedia card that corresponds to each presentation subject, such as a particular fish or a fish tank containing several fish species. Each multimedia card provides users with content and touch-screen buttons that enable them to control content presentation and navigate between tanks.

The PEACH (Personal Experience with Active Cultural Heritage - http://peach.itc.it) project located in the Castle of Buonconsiglio in Trento (Italy), shows that the application of the rules of cinematography to multimedia presentations delivered on a mobile museum guide helps decrease the cognitive overload (Alfaro et al., 2004). Using the PEACH system, museum visitors can request personalized 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 or desktops, and more).

In the context of the PEACH project, a number of prototypes aimed at providing the visitor with a personalized experience were built and evaluated. One prototype was composed of a primary multimedia guide (Alfaro et al., 2004), the mobile system of which combined the dynamically adapted language-based output (an improvement on well established techniques) with a dynamically produced visual documentary (based on cinematic techniques). Inputs to the system came from the locations of the visitors and from observations the system itself made about their behavior and presumed interests and what they had been exposed to and presented so far. Experiments with visitors showed that many subjects would have liked to access follow-up information during the presentation. At the same time, this option should not cause awkward interruptions to the flow of the presentation. In a later guide (Rocchi et al., 2004), the focus of the prototype was on automatically produced video clips to be played on the small screen of the mobile device, using a life-like character either as an anchorman or a presenter.

PEACH focuses especially on presentations personalization. A number of projects have introduced technology that presents objects exhibited in museums to individuals. The technology typically takes advantage of a positioning system (a system based on devices that generate an infrared signal from fixed positions, on radio frequency emitters, on triangulation through emitter/receivers of wireless digital signals, or on very sensitive GPS systems which work inside buildings). The visitor has a small portable device (for example, a PDA) and can receive information relevant to the particular object. The system may know the profile of each visitor, where the visitor has been, what his physical path has been, and what has been presented to him. More generally, a dynamic user model can be built during the course of the visit. This opens up the possibility of offering presentations and information specifically tailored to the individual visitor.

In order to increase the personalization of the museum guide, the concept of situation-aware content must be incorporated. Information is most effective if presented in a cohesive way, building on previously delivered information, and in accordance with the setting in which the users finds themselves. In other words, the presented information must be appropriate to the physical location of the visitor as well as to the location of the works of art within the environment. Cohesion, on the other hand, requires that the information to be found in an exhibit be presented in a flexible yet coherent manner with respect to the user’s physical location and to the overall flow of information. The overall experience and the absorption of new information is thus maximized, while also stimulating the visitor’s interest along with the desire to inquire, analyze and teach (Stock et al., 2005).

Within PEACH, we are investigating the concept of the “Active Museum”. An active museum is a form of “active environment” (McCarthy, 2001), and can be seen as a large scale multi-user, multi-media, multi-modal system. According to Bordegoni et al. (1997), a medium is a physical space in which perceptible entities are realized. Indeed, in a museum (as well as in a city of cultural interest, an archaeological site, and so on) the most prominent medium is the environment itself. The main goal which the information presentation system must achieve is the integration of the ‘physical’ experience, without competing with the original exhibit items for the visitor’s attention.

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

- 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 is a single, monolithic system.
- Services are provided by a variable set of components that can join and leave the environment, and can be operating anywhere.
- Services provided by components may (partially) overlap; these components therefore need to coordinate with each other in order to decide, for instance, which one should provide a specific service, and how.

Complex distributed applications emerging in areas such as e-Business, e-Government, and the so-called ambient intelligence (i.e., “intelligent” pervasive computing (Ducatel et al., 2001), such as our "active museum" environment) need to adopt forms of group communication that are very different from classical client-server and Web-based models (see, for instance, Penserini et al., 2003). The so-called service-oriented computing (SOC) is the paradigm that accommodates the more dynamic and adaptive interaction schemata mentioned above. Service-oriented computing is applicable to ambient intelligence as a way to access environmental services, e.g., sensors or actuators close to a user. Multi Agent Systems (MAS) naturally accommodate the SOC paradigm. In fact, each service can be seen as an autonomous agent (or an aggregation of autonomous agents), possibly without global visibility and control over the global system, and characterized by unpredictable/intermittent connections with other agents of the system.

Adopting techniques based on the concept and use of autonomous software agents, at both the design and the implementation level, is central to providing the functionalities and qualities we expect from such a system. The PEACH project was developed as a multi-agent system, separating the application from the agents' communication infrastructure, allowing developers to focus on application-related aspects by taking advantage of all the services provided by the infrastructure (Kuflik et al., 2004).

To summarize, very few attempts have been made to investigate the architectural issues emerging from the embedding of a mobile guide into an overall active museum. In an active museum, different devices, mobile or fixed, have to interact with a variety of services. Not only cannot the number of devices in the museum be easily predicted, but also the available services may change over time. Moreover, it is desirable that services are provided in a user- and context-adapted way. A common, generic architecture may facilitate research by allowing researchers to share system components, relieving them of the necessity to build a new infrastructure for every new project.

This paper reports on the experience of the PEACH project and its successor in the Hecht museum and suggests a generic architecture for "Active Museum".

2. PEACH Case Study

2.1 "PEACH" Museum Visit Scenario and Architectural Components

Let us assume that a visitor to the museum is using hand-held device for requesting and receiving presentations about exhibits. Let us consider the following scenario: while walking around the museum and watching the exhibits the visitor requests a presentation of (one item of/part of) an exhibition. As said, the primary interaction device used by the visitor is a Personal Digital Assistant (a PDA that corresponds to what we will call a ***User Assistant***). Nevertheless, while walking around (and before the requested presentation is delivered), the visitor may approach some presentation devices that are more comfortable to use and handle the presentation better than her/his PDA, e.g., in terms of pixel resolution or audio quality. How can the presentation be prepared and delivered to the visitor in the best possible way? Moreover, presentations can be given using variety of media options – audio, video, or text and they should be adapted to the specific visitor, based on his location, interest, visit history etc'

We assume that there are different ***Presentation Composers*** capable of building presentations for the visitor, and that each ***Presentation Composer*** relies on ***Information Mediator***(s) to provide the information required for the generation of the presentation. Moreover, we may also assume that each ***Presentation Composer*** is able to proactively propose its 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. Moreover, we expect that all the services are “dynamically validated”, that is, due to the fact that the environment and the user location are quickly changing, only appropriate services are considered.

Such an environment is highly dynamic: users enter and leave the environment, and availability of services cannot be guaranteed (service providers may be busy serving requests or simply off-line). Hence the various components should exhibit autonomous behavior and proactiveness. All the above makes agent-oriented software engineering (Jennings, 2000) the most appropriate approach for developing such systems.

2.2 Infrastructure

We used a type of group communication, called channeled multicast (Busetta et al., 2002), as the main technique for agent coordination in ambient intelligence scenarios. Messages sent on a channel are received by all agents tuned into it. Channelled multicast often reduces the amount of communication needed when more than two agents are involved in a task, and allows overhearing (i.e., the ability to listen to messages addressed to others). Overhearing, in turn, enables the collection of contextual information, pro-active assistance (Busetta et al., 2001), monitoring (Kaminka et al., 2002), and even partial recovery of message losses.

Our experimental communication infrastructure, called LoudVoice, was designed to support channeled multicast and implicit organizations, a group of unknown number of agents performing a specific role (Busetta et al., 2003). LoudVoice uses the fast but inherently unreliable IP multicast and it allows senders to address messages either to some specific agent or to a role, i.e. to all agents that, at the time of delivery, offer a certain service on a channel.

2.3 Applications

Currently two prototypes are under experimentation: one in Italy and one in Israel.

Castello del Buonconsiglio – "Sala Grande". The first version of the agent’s system was developed for the Buonconsiglio Castle in Trento. In a group of adjacent rooms containing pictures and artistic objects dated to about the early sixteenth century. The system is composed of two main (interdependent) components: a three-tier "classical" application that consists of a presentation layer (User assistant), application layer (Web server) and data layer (Multimedia database), and a Multi Agent System that consists of a collection of functional agents communicating to provide the required services, as illustrated by Figure 1. These agents communicate in order to provide relevant presentations to the museum visitor based on his/her location (later on, this will be extended to provide personalized presentations).

The initial system is composed of five agents[1]:

---

[1] Please note that the user modeler agent (dashed) is not part of that system

1. ***User Assistant*** runs on the user's PDA and provides the user with interface to the system (information presentation/user actions);
2. ***Presentation Composer***, which receives explicit and implicit user requests and replies with appropriate presentations which are small Flash presentations;
3. ***Information Broker*** (currently a Multimedia Data Base with an interface) containing all available presentations, which responds to a presentation request by providing lists of those available;
4. ***Spatial Information Broker***, which receives user position information, manages it, and sends the trusted position to the Presentation Composer;
5. ***PositionAgent*** that tracks users in the environment (by tracking their PDAs) and reports their position to the “positionAgentsCollector” channel.

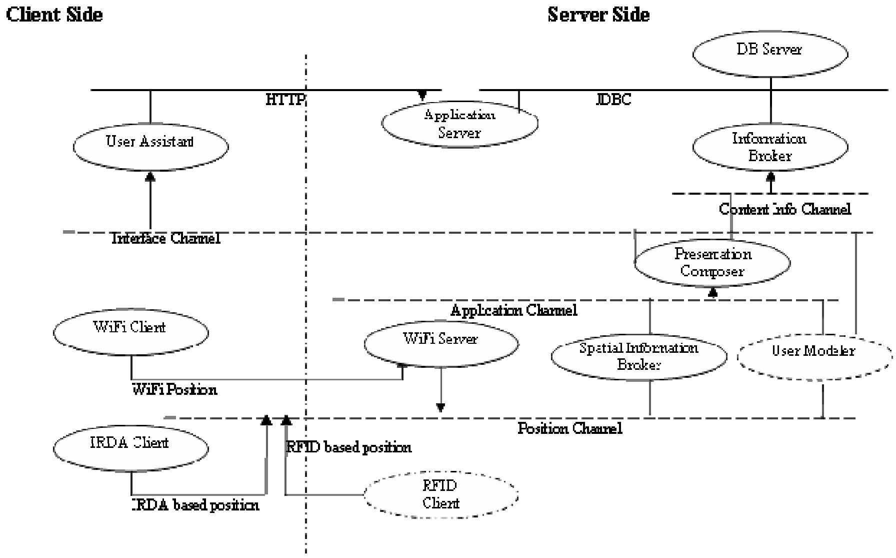

Fig. 1. Bounconsiglio/Hecht museums system architecture

Three different types of localization systems were used in the experiment: WiFi, Irda, and Rfid. The various positioning agents can work simultaneously or one at a time. The WiFi-based positioning agent (WiFi server) periodically provides WiFi-based positioning information about the visitors (i.e., their PDAs) present in the environment. The RFID and IRDA agents provide positioning information whenever available (whenever an IR beacon or RFID signal are detected). This information is used by the other agents to provide presentations relevant to the current user position.

The Spatial Information broker collects and evaluates positioning information and selects the most trusted one to report. This is required due to the fact that, WiFi-based positioning information is continuously available, but was found to be inaccurate, while IR-based positioning and RFID-based positioning are accurate but not continuously available. It should be noted that in practice, the accuracy of the WiFi-based positioning system was found to be inadequate (accuracy of +/- 3 meters) but it provided continuous positioning data. Hence the combination with RFID and IRDA technologies proved to provide complimenting solutions.

The system works as a client-server system, where the client is the ***User Assistant***. The client may request presentations from the server – the ***Presentation Composer*** that in turn uses information available in the ***Information Broker***.

Currently four LoudVoice communication channels are defined: "Position Channel" for collecting positioning information; “Application Channel” that is used for positioning information reporting (between the ***Spatial information Broker*** and ***Presentation Composer***); "Interface Channel" for user interface communication; and “Content Info Channel” which is used for presentation communication (between the ***Information Broker*** and the ***Presentation Composer***).

A web application protocol is used to complement the communication between the mobile device (***User Assistant***) and the ***Information Broker,*** for requesting and transferring the actual presentations selected for presentation to the selected presentation device. The ***Information Broker*** server provides JDBC interface to the Multimedia DataBase, so that a selected presentation can be requested by the Client and sent to him/her to be played. The database server subsystem includes a MySQL database.

This server also provides the web interface for receiving service requests and serving them through the relevant application. It is composed of a Tomcat web server which receives requests in HTTP protocol from the mobile device and serves them by activating the application that accesses the ***Information Broker*** (Multimedia Database). A specific message protocol for the entire system was defined, based principally on the position information, since position is the most relevant information in our system.

Hecht Museum. The first system described above is currently being deployed in the Hecht museum, at the University of Haifa in Israel. In two adjacent exhibitions, one presenting a 2,400 years old ship recovered near the coast of Ma'agan Michael (northern Mediterranean coast of Israel) and the second presenting the Phoenician culture. Both are archeological exhibitions. The same basic system architecture is used, with minor modifications which extend it to support more complicated scenarios. In the Hecht scenario, the positioning technologies used are based on WiFI and IR (RFID is not used). In addition there are several user modeling agents: collaborative, stereotypic and feature-based supporting presentation personalization. Figure 1 illustrates also the Hecht system architecture. The basic architecture is identical for the two exhibits, with the addition of the ***User Modelling*** agent, and without the application of the RFID positioning technology. It is easy to see that system architecture can be easily modified / extended to include new services, without the need to re-design the whole system from scratch.

3. Generic Architecture for Museum Visitors Guide

The work done at the Buonconsiglio Castle and later extended to the Hecht museum demonstrates the usefulness of a generic agents' communication infrastructure (described in section 2.2) as a basis for the development of "active museum". Moreover, it demonstrates the generality and scalability of the proposed initial system architecture. Based on the experience gained so far, we would like to propose the application of generic agent architecture to active museums, applying the idea of implicit organization. In order to illustrate the need for and the idea of Implicit Organization, let us consider the following simplified scenario presented in the sequence diagram of Figure 3. 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 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. Each member suggests the service it can provide. The “best” service should be selected (negotiated among the members of the Implicit Organization members) 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 of screening the presentation being planned. When all the information has been received, the Implicit Organization of ***Presentation Composers*** can reason and decide on the best presentation to prepare. This presentation will then be sent from the composers as a group response to the selected ***User Assistant***.

According to the museum scenario briefly described above, each component represents a role or an Implicit Organization of agents. The following implicit organizations are identified:

Spatial information mediator. This implicit organization is responsible for reporting user current position periodically and/or on request. It provides positioning relations between visitors and museum exhibits.

User modelers that may include different agents implementing user modeling approaches, such as the collaborative approach, a stereotypic approach or 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.

Information logger/analyzer is a set of development and research support tools (ignored for the sake of simplifying the presentation).

Every implicit organization in this group may adopt a relevant negotiation policy, either a competition where agents are competing to provide information relevant to a presentation, or a collaboration, where agents may collaborate to provide a complete solution, for instance an accurate estimation of the user’s position based on complementary sensors input (WiFi and IRDA in our case).

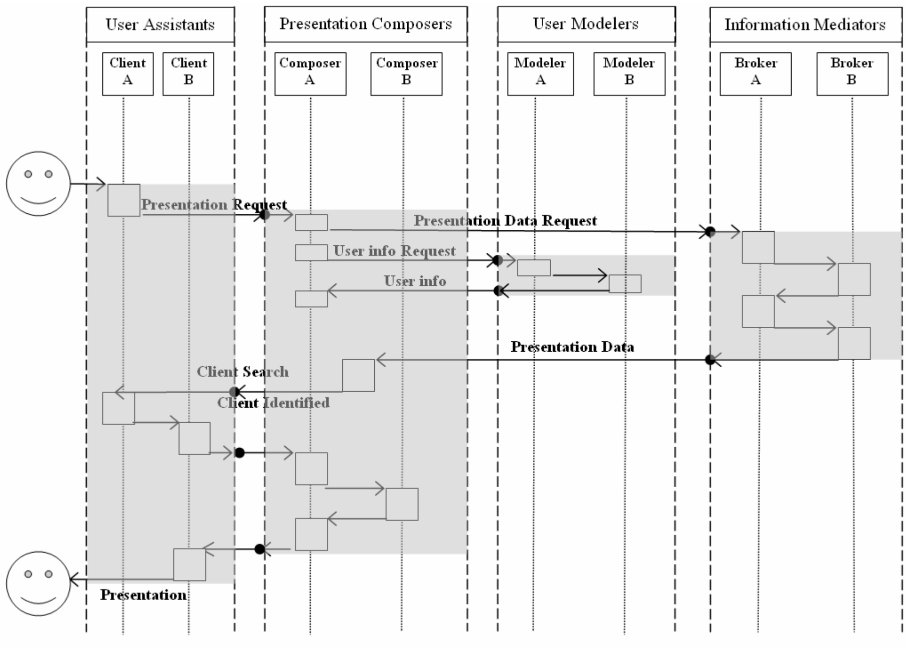

Fig. 2. Implicit Organizations of Agents Communication

The above mentioned architecture is highly dynamic: any number of agents can perform any role, this addresses the previously presented challenges of active environment where there may be a variable number of users at a any place and time, users (and services) may enter and leave the environment at will and services can be provided by more than one agent at any given time. Moreover, new roles can be identified and new implicit organizations added to the system without a need to change the whole system, by simply adapting the relevant agents to the new services offered. For example, a new positioning technology will not require any modification of the system, while adding a museum-shop agent(s) will require only the adaptation of user assistants, to take advantage of the new services that are offered.

4. Conclusions

Museum visitors' guides have recently been attracting a growing interest and firing research efforts. However, the architectural aspects of these systems had not yet been coherently addressed. This work demonstrates how the complex requirements of a highly distributed active museum system are supported by applications developed using a generic agents' communication infrastructure. It also shows how the idea of implicit organization enables the definition of dynamic and extensible system architecture.

As a result, we were able to propose a generic agent-based architecture that may facilitate the development of Active Museum. Future work should focus on supporting the generic architecture using an appropriate communication protocol, and on guidelines for extending such systems when new roles are defined. It is worth noting that current bandwidth of WiFi technology limits the practical application of the presented architecture, but we assume that with time, communication technology development will resolve this limitation.

References

1. Alfaro I., Nardon M., Pianesi P., Stock O., Zancanaro M. (2004) Using Cinematic Techniques on Mobile Devices for Cultural Tourism, In *Information Technology & Tourism*, Vol. 7, Issue 2..
2. Alison D. and Gwaltney T., (1991) How people use electronic interactives. In Proceeding of *Hypermedia and Interactivity in Museums* Pittsburgh: Archives & Museums Informatics Technical Report.
3. Bellotti F., Berta R., de Gloria A. and Margarone M. (2002) User Testing a Hypermedia Tour Guide, in *Pervasive Computing* IEEE, April – June.
4. Berkovich M., Date J., Keeler R., Louw M. and O'Toole M. (2003) Discovery Point: Enhancing the Museum Experience with Technology. In Proceedings of CHI 2003, April 5-10, 2003, Ft. Lauderdale, Florida, USA.
5. Busetta P., Serafini L., Singh D., and Zini F. (2001) Extending Multi-Agent Cooperation by Overhearing. In Proceedings of the *Sixth Int. Conf. on Cooperative Information Systems* (CoopIS 2001), Trento, Italy.
6. Busetta P., Don`a A., and Nori M. (2002) Channeled multicast for group communications. In Proceedings of the *First International Joint Conference on Autonomous agents and Multiagent Systems*. pages 1280–1287. ACM Press.
7. Busetta P., Merzi M., Rossi S., and Legras F. (2000) Intra-role coordination using group communication: A preliminary report. In F. Dignum, editor, Advances in Agent Communication, LNAI. Springer Verlag 2003.Cheverst K., Davies N., Mitchell K., Friday A., and C. Efstratiou. Developing a context-aware electronic tourist guide: Some issues and experiences. In *Proceedings of CHI 2000*, Amsterdam.
8. Danin A. K. (2001) Understanding and Using Context. In *Personal and Ubiquitous Computing 5(1), pp. 4-7.*
9. Ducatel K., Bogdanowicz M., Scapolo F., Leijten J., and Burgelman J.-C. (2001) Scenarios for ambient intelligence in 2010. *Technical report, Information Society Technologies Programme of the European Union Commission (IST),* Feb.. http://www.cordis.lu/ist/.
10. Economou M. (1997) The Evaluation of Multimedia Application for Gallery Interpretation: The Euesperides Project in Oxford. In Bearman, D. and Trant, J., (Eds.), *Museum Interactive Multimedia: Cultural Heritage Systems, Design and Interfaces*. Pittsburgh, p. 218 – 226.
11. Evans J. A. and Sterry P. (1999) Portable Computers and Interactive Multimedia: A New Paradigm for Interpreting Museum Collections", In: Bearman, D. and Trant, J., (Eds.) Cultural Heritage Informatics 1999: Selected Papers from ICHIM99. Pittsburgh, p.113 – 126.

12. Goren-Bar D., Graziola I., Kuflik T., Pianesi F., Rocchi C., Stock O. and Zancanaro M. (2005) I Like It – An Affective Interface for a Multimodal Museum Guide. In *Proceedings of Intelligent User Interfaces IUI'05*, San Diego, CA., January.
13. Graziola I., Pianesi P., Zancanaro M. and Goren-Bar D. (2005) Dimensions of Adaptivity in Mobile Systems: Personality and People's Attitudes. In *Proceedings of the 10th International Conference on Intelligent User Interfaces*, ACM, New-York, p. 223 – 230.
14. Jennings, N.R., (2000) On Agent-Based Software Engineering. *Artificial Intelligence*, 117 (2), 277-296.
15. Kaminka, G., Pynadath, D. and Tambe. M. (2002) Monitoring Teams by Overhearing: A Multi-Agent Plan-Recognition Approach. *Journal of Artificial Intelligence Research*, 17: 83–135
16. Kuflik, T., Busetta P., Penserini, L., Bresciani, P. and Zancanaro, M. (2004) Personalized Information Delivery in Dynamic Museum Environment by Implicit Organizations of Agents. In *Workshop on Environments for Personalized Information Access*, May 25, 2004, Gallipoli, Italy, pp. 22-33.
17. Penserini L., Liu L., Mylopoulos J., Panti M., and Spalazzi, L. (2003) Cooperation strategies for agent-based p2p systems. In *WIAS: Web Intelligence and Agent Systems: An International Journal, IOS Press*, 1(1):3–21, 2003.
18. McCarthy, J. (2001) *Active environments: Sensing and responding to groups of people. Personal and Ubiquitous Computing*, 5(1)
19. Rocchi C., Stock O., Zancanaro M., Kruppa M., and Krüger A. (2004) The Museum Visit: Generating Seamless Personalized Presentations on Multiple Devices. In *Proceedings of Intelligent User Interfaces IUI 2004*, Madeira, January.
20. Stock O., Zancanaro M., Not E. Intelligent Interactive Information Presentation for Cultural Tourism (2005) In O. Stock, M. Zancanaro (eds.) Multimodal Intelligent Information Presentation. Springer, Dordrecht.

Acknowledgements

This work has been developed in the context of the PEACH Project and of the ITC-University of Haifa Collaboration. The authors would like to thank Paolo Bresciani and Loris Penserini for their contribution to the system's architecture definition of the "Sala Grande" system.

User Profiling for Generating Bids in Digital Signage Advertising Auctions

Jörg Müller and Antonio Krüger

Institute for Geoinformatics, University of Münster, Münster, Germany,
{joerg.mueller, antonio.krueger}@uni-muenster.de

Abstract. In this paper we present a simple approach to adapt advertisements on digital signage to the interests of the audience. We use auctions to sell the advertising space to the highest bidding advertisement and decision theory to determine for each advertisement how much to bid. Within this framework, we use methods from content based recommender systems to adapt the bid to the interests of the audience. Each advertisement has a set of keywords, and the history of all advertisements a user was interested in is kept. We propose to use a naive Bayes classifier to estimate the probability that a user is interested in a certain advertisement given the keywords and users history. We propose interaction mechanisms to generate feedback to train the estimation functions and use the m-estimate to deal with the cold start problem.

1 Introduction

Imagine you walk around with two friends in your favorite shopping mall. You just pass a digital sign where two strangers are standing in front of it. As you notice that the display changes its content to adapt to your interests, you stop to have a glimpse on it. You see an advertisement presenting a discount for a DVD you are interested in, and take out your mobile phone to establish a connection to the display and copy it to your phone. While your mobile phone then navigates you to the place where you can buy the DVD, you wonder how the digital signs can always present you advertisements you are really interested in. In this paper we focus on how to select those advertisements you are interested in based on your interaction history. Main contributions are

- The application of the naive Bayes classifier to this problem.
- The presentation of feedback mechanisms to train the estimation functions.
- The application of the m-estimate to solve the cold start problem.

2 Related Work

The concept of using auctions to select advertisements shown on digital signage was introduced in [5]. The audience is sensed with a Bluetooth sensor, and advertisements are preferentially shown to those users that have not seen them yet. A bidding strategy for digital signage advertising auctions that uses the expected utility given a certain context to determine the optimal bid is presented in [3]. The application of the naive Bayesian classifier for recommender systems was proposed in [6]. In that work, websites are categorized into "interesting" and "not interesting" categories. The system is trained by manual feedback from the user and employs the naive Bayes classifier to automatically classify webpages based on features extracted from the content. An overview of recommender systems is provided in [1]. A general overview of work on situated public displays can be found in [4].

3 Auctions for Digital Signage Advertising

In this paper we concentrate on advertisements that entice the user to take a specific action, eg. buy a certain product. This action has a certain utility $U(act)$ for the advertiser, which

Fig. 1. The test environment for this study. The displays are installed in an university setting and show advertisements for talks, lectures and seminars.

is why the advertiser advertises at all. Because advertising space on displays is limited, for each situation the advertisements that are shown must be selected. We sell advertising space using a generalized second prize auction [2]. Each advertisement is represented by an agent, which is provided context data to generate its bid. For this paper, the context taken into account is the number of faces in front of the display, as detected with a camera, and the Bluetooth IDs of the audiences mobile phones, as detected with a Bluetooth sensor. Each advertisement should then bid its individual utility of being shown given the context data.

4 Applying User Profiles to Estimate the Optimal Bid

A simple model of the expected utility of showing an advertisement $EU(ad)$ is the utility that the user takes the advertised action $U(act)$ times the probability $P(act)$ that he does so. For multiple users in the audience, the individual utilities can simply be summed. Thus

$$EU(ad) = \sum_{i \in viewers} P(act_i | Context_i, ad) * U(act_i)$$

To be able to estimate whether a certain user is interested in an advertisement, we follow the approach proposed in [6]. Each advertisement is represented by a number of features f_i, in our case we use the submitting institution and a number of manually entered keywords. It would be helpful, although not necessary, if keywords are semantically unambiguous, that is, different advertisers use the same keywords for the same purposes. Then we take the naive Bayes assumption that the features are conditionally independent given act and ad, and apply Bayes' rule to achieve

$$P(act_i | f_1 \dots f_n, ad) = P(act_i | ad) \frac{\prod_{j \in feat} P(f_j | ad, act_i)}{\prod_{j \in feat} P(f_j | ad)}$$

$P(f_j|ad)$ is the probability that a new advertisement has the feature f_j and can simply be estimated from the history of advertisements. $P(act_i|ad)$ is the probability that the user acts upon any given advertisement. It can be estimated from the percentage of advertisements the user has seen up to now and those he acted upon. $P(f_j|ad, act_i)$ is the probability

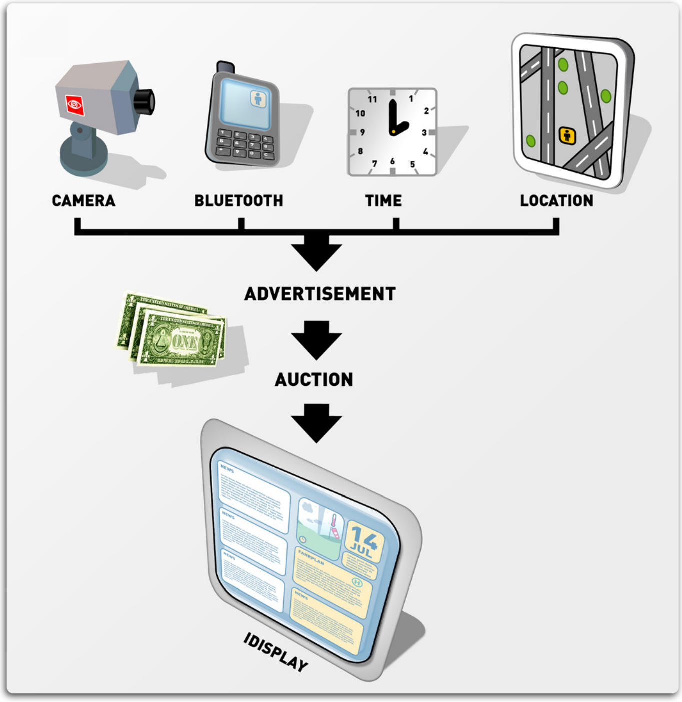

Fig. 2. The auction process used in our approach. At the start of each advertising cycle, context information is gathered. The number of users looking at the display, their identity, and the current time and location are provided to the advertisement. Advertisements take this context information and estimate the probability that the users will act. From this estimation they compute the total expected utility of being shown and generate their bids for the upcoming advertising cycle. In a generalized second price auction, the available advertising slots are then sold to the highest bidders. For the duration of the advertising cycle, the selected advertisements are shown on the display.

that an advertisement that the user acted upon has the feature f_j. This can be estimated from the past advertisements the user acted upon. To be able to estimate both of the last probabilities, a feedback mechanism is necessary to determine which advertisements the user really acted upon. We propose two approaches to implementing such a feedback mechanism. First, interaction mechanisms, for example via the user's mobile phone, can be used to enable the user to copy an advertisement to his todo list or calendar, or be directly navigated to the place where he can buy the product. Second, a bonus coupon program could be used to trace back which advertisements led to which buys. The user could pick up a coded coupon at the advertisement, for example by taking a photo with his mobile phone, to get a discount at the shop. This coupon would encode the time and location of the advertisement. In addition to the feedback problem, we would have to deal with the cold start problem. For a new user, no information is available to estimate the mentioned probabilities. Therefore, we propose to use the m-estimate to start for each user with a global estimate of a mean user and gradually refine it as individual data is gathered. To estimate for example $P(f_j|ad, act_i)$, let act_{i,f_j} be the number of actions user i took with the feature f_j, act_i the total number of actions for user i, act_{f_j} the total number of actions with feature f_j for all users and act the total number of actions for all users. Let $P(f_j|ad, act) = \frac{act_{f_j}}{act}$. Then the m-estimate for $P(f_j|ad, act_i)$ is $\frac{act_{i,f_j} + mP(f_j|ad,act)}{act_i + m}$. M is also called equivalent sample size and can be interpreted as how often we assume to have observed user i to behave like the average user. A reasonable number for m would be 10, for example.

5 Conclusion and Future Work

We presented a simple approach to adapt advertisements on digital signage to the interests of the audience. We used auctions to sell the advertising space to the highest bidding advertisement and decision theory to determine for each advertisement how much to bid. Within this framework, we used methods from content based recommender systems to adapt the bid to the interests of the audience. Each advertisement has a set of keywords, and the history of all advertisements a user was interested in is kept. We proposed to use a naive Bayes classifier to estimate the probability that a user is interested in a certain advertisement given the keywords and the users history. We proposed interaction mechanisms to deal with the feedback problem and using the m-estimate to deal with the new user problem.
We plan to refine our method and evaluate other approaches from recommender systems for applicability to generate the bid as well. We plan to automate the feature identification, using expected information gain or the TF-IDF measure. In addition, we plan to evaluate our approach in a simulation as well as in a real-world experiment.

References

1. Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6):734–749, June 2005.
2. B. Edelman, M. Ostrovsky, and M. Schwarz. Internet advertising and the generalized second price auction: Selling billions of dollars worth of keywords. Working Paper, http://rwj.berkeley.edu/schwarz/, 2005.
3. Jörg Müller and Antonio Krüger. How much to bid in digital signage advertising auctions? In *Adjunct proceedings of Pervasive 2007*, 2007.
4. Kenton O'Hara, Mark Perry, and Elizabeth Churchill. *Public and Situated Displays: Social and Interactional Aspects of Shared Display Technologies*. Springer, 2003.
5. Terry Payne, Ester David, Nicholas R. Jennings, and Matthew Sharifi. Auction mechanisms for efficient advertisement selection on public displays. In *Proceedings of European Conference on Artificial Intelligence*, pages 285–289, 2006.
6. Michael J. Pazzani and Daniel Billsus. Learning and revising user profiles: The identification of interesting web sites. *Machine Learning*, 27(3):313–331, 1997.

Here and Now:
A User-Adaptive and Location-Aware Task Planner

Christoph Stahl, Dominik Heckmann, Tim Schwartz, Oliver Fickert

Saarland University
66123 Saarbrücken, Germany
[stahl, dominik, schwartz]@cs.uni-sb.de

Abstract. In this paper, we present a ubiquitous location-based task planner that integrates a to-do list and a schedule/calendar with route knowledge and adapts both its view and alarms to the user's current situation. The task planner is hosted on a web server and can be accessed from everywhere via a mobile Web terminal, such as a mobile phone or notebook, or a public display infrastructure which recognizes the user by their Bluetooth-device. Tasks can be localized by specifying a certain location where the task can be accomplished, such as an office or a store. Alternatively, a category can be chosen from an ontology that includes activities like shopping, sports or traveling by airplane. Since the task planner is likely to include dozens of tasks for the near future, it is too large to be browsed on the go. Therefore the planner implements a 'here-and-now' view, which adapts to the current time/date and location of the user. Based on knowledge about the purpose, address and opening hours of locations and routes, the task planner is able to filter for tasks that can be accomplished nearby, considering the time to reach the location and other deadlines. A second feature is an adaptive reminder, which considers the time that is needed to travel to the specified location of a task.

1 Introduction

With the advent of mobile computing devices, personal information management (PIM) tools have been implemented on PDAs and are now a common feature of mobile phones. Usually, the devices offer a calendar that allows to add events and to set an alarm time for a reminder. The calendar is typically supplemented by a to-do list or notepad, where the user can add tasks that have no certain date. Whereas the conceptual model behind these PIM tools is well understood by both the designers and the users, it has obvious drawbacks that we want to address in this paper.

Consider the following scenario, where a user C who is working in Saarbrücken creates a new entry in his calendar that reminds him to take a flight that starts in two weeks from Zweibrücken at 6:35 pm. C has to consider the time that is required to drive to the airport, lets say 30 minutes, and to prepare himself for the trip, another 30 minutes. The check-in procedure requires that C arrives at the airport one hour earlier, so he will set his alarm to 4:05 pm. However, two weeks later C is on a business trip in Munich, and the low-cost flight could not be changed. Now he is more than 4 hours from Zweibrücken. The alert can't recognize the new situation and would remind C much too late. In other situations, when the user is already on site, the problem is reversed and the alert will be activated too early. This is not too much of a problem since the user can regularly check their calendar for upcoming events. However, on mobile devices the display size is a limiting factor and the events have to be checked day-by-day in a cumbersome procedure. Also, the to-do list has to be additionally browsed for unfinished tasks. Again, the display is usually limited to a few tasks, so the user always has to navigate step by step through the complete list of items.

In the scenario, C also requires driving instructions from Munich to Zweibrücken. He has to look up the street address of the airport from the website and to manually enter it into a route planner or onboard navigation system. In case of an unfamiliar rental car, C might even need to consult the manual of the car in order to do so.

Further shortcomings arise from the P in the PIM concept: the tools are strictly limited to personal use. Let's assume another scenario: As C's girlfriend D recognizes that there is no milk in the kitchen, she adds it to her mobile phone's to-do list. C, who is in a meeting with a customer close to a shopping center, could quickly buy it on his way home - if he would know about both the milk and the store nearby.

Our goal is to ease the use of reminder tools by reducing the users' cognitive load and to simplify the workflow between PIM and navigation systems through the principles of user-adaption and location-awareness. We adopt a Web 2.0-based approach similar to Google's calendar application that offers ubiquitous access via mobile internet terminals and allows for collaboration between multiple users through the sharing of tasks and events. The necessary ubiquitous user- and location modelling infrastructure is contributed by the GUMO and UbisWorld ontology. Furthermore, we utilize a commercial Web service for route planning.

In the following sections, we will introduce relevant concepts and related work, before we present our own task planner and describe the modelling of tasks, users and locations. We explain how the reminder function and the task view adapt to the current location of the user and close the paper with an outlook and conclusions.

2 Basic Concepts and Related Work

In order to ensure that the whole variety of the distributed systems use the same semantics for the communicated concepts we base this work on ontologies and especially on semantic web technologies. The initial idea behind the semantic web is to annotate documents in the World Wide Web with semantic information by the use of ontologies, see [Fensel et al., 2003]. Languages like RDF and OWL have been designed to allow the author to declare the page contents as resources. Statements can be included in order to describe the semantics and relationships between resources using ontologies. By doing so, the web may be used in the sense of a very large distributed knowledge base. The goal in mind is the development of a machine-processable and human readable language, see [Studer et al., 2003], for representing and exchanging knowledge about a specialized domain. For example in [Stahl and Heckmann, 2004] we show how to use the semantic web technology for ubiquitous hybrid location modeling.

The concept of web services provides an interesting solution to solve the integration problem among heterogeneous application systems. It can be seen as a kind of standardized software technology to integrate and share various systems. It has the advantage of flexibility by perfectly defining standard specifications for mutually sharable data among distributed systems. So the web services provide the advantage that they can transparently access any web servers in any place with any device and at any time. The web services architecture combines three essential roles: service provider, service registry and service requestor.

During the course of our work, a reactive calendar on a mobile phone has been implemented by Glaubitt [Glaubitt 2006] that overcomes the limitations of a static calendar in a mobile environment through three types of situation-dependent events. First, the system dynamically adapts the reminder for a task according to the distance between the user and the location where the event is going to take place. Secondly, events can be triggered as the user enters a certain place. The third type of event is defined by its participants. For example, a meeting event begins if all the necessary persons come together. As an event begins, the phone adapts its ringtone settings according to the type of event. For example, the phone is muted during a conference. As the event ends, the phone's normal mode of operation is restored. Since Glaubitt's work is not based on spatial ontologies, locations can only be specified in geographic coordinates, which narrows the usability of the system. It has also not yet implemented street map data for route planning, so that the distances are only rough approximations.

For our planner, we use commercial Web-services for routing and geocoding provided by PTV[1]. The *eLocate* [PTV AG, 2004a] server converts postal address data into geographic coordinates so that the addresses can then be invoked for route calculations. The postal address can be incomplete and can also contain spelling mistakes. The *eRoute* [PTV AG, 2004b] server is based on an algorithm calculating the shortest routes on the basis of cost, using a digital road network. The costs per trace consist of the costs per time unit and the costs per route. The speed for a trace is determined by the type of road and the vehicle profile for this type. The algorithm can choose between the shortest route, the quickest route, or a linear combination of the two.

For the exact calculation of walking distances, we use the Yamamoto toolkit [Stahl and Haupert, 2005] that allows us to model multi-level indoor environments and supports route finding for pedestrians. In [Stahl and Heckmann, 2004] we describe a hybrid location model that consists of a semantically supported symbolic location model as part of the UbisWorld spatial ontology (see www.ubisworld.org) and the geometric location model of the Yamamoto toolkit. Aside from its core functionality for navigation, it also provides a common notion of location to mobile applications. This requires a standardized and structured vocabulary referring to real-world locations that allows for spatially indexed information and the exchange of positions between applications. Our model represents real-world places at different granularities as resources in the World Wide Web. The geometrical model is joined with UbisWorld by uniform resource identifiers (URIs) as location identifiers.

In order to realize the location-adaptation of our system, we need knowledge about the current whereabouts of the user. If the user is staying outside, this can be done

[1] PTV Planung Transport Verkehr AG, Karlsruhe. Website: http://www.ptv.de

with the use of GPS. Unfortunately, GPS is not working in indoor environments, e.g. a shopping mall or an airport. Currently there are several systems under development to overcome this restriction. All of these systems use some sort of senders (ultrasound beacons, infrared beacons, WiFi-hotspots, RFID tags, Bluetooth beacons, to name but a few) and corresponding sensors to detect or read these senders. Basically there are two options to set up such a system: Installing sensors in the building and letting the user wear the sender or installing the senders in the environment and letting the user wear the sensors. In the first case, the so-called exocentric localization, the user is sending information to the environment and some centralized server uses this data to calculate their position. In other words, the user is tracked. In the latter case, the egocentric localization, the user receives information from the environment and their personal device uses the data to calculate the current position. In [Brandherm and Schwartz, 2005] and [Schwartz et al., 2005] we describe the basics of an egocentric localization system that uses geo referenced dynamic Bayesian Networks to determine the position of a user. This system, which is now called *LORIOT* (Location and ORientation in Indoor and Outdoor environmenTs), uses infrared beacons (IR beacons) and active RFID tags to determine the user's position. LORIOT runs on a PDA and uses the built-in infrared receiver and an attached active RFID reader as sensors. Each IR beacon sends out a 16 bit wide identification code. Receiving such a beacon gives a high probability that the user is standing near that particular beacon. Furthermore, if we know the direction of the infrared light beam, we can determine the user's direction. However, the disadvantage of IR beacons is that the PDA's IR sensor must be in the line of sight of the beacon and can thus very easily be blocked. The RFID tags on the other hand send their information via radio waves, which can be received even when the PDA resides in the user's pocket. But due reflections and damping of radio waves, receiving an RFID tag gives only little evidence that the user is standing in its vicinity and the signal is missing a direction-information. By combining these two sensor readings with the help of dynamic Bayesian Networks, we achieve what we call an Always Best Positioned system (ABP system): As long as there are either IR beacons or RFID tags detectable, we will be able to estimate a position whose precision depends on the type of the sender. If we can receive both, we will get an even higher precision. In Section "Localization" we describe a system that can be used to estimate positions in an uninstrumented environment.

3 A Web-based Task Planner

Our ubiquitous task planner *Ubidoo* integrates the conceptual models of a diary and a to-do list into a hierarchical task tree, where each task can have multiple subtasks. In our context, each task acts a reminder to do a certain activity or action. The subtasks can either represent alternative choices, multiple goals, or fixed sequences of actions.
Similar to a to-do list, the status of each task is graphically represented by a checkbox. Initially, the task is not assigned and the checkbox is greyed (disabled). Once the task is assigned to someone, the checkbox becomes active (enabled). Finally, the task can be checked by the responsible user to indicate completion or marked with a warning sign to indicate failure. In analogy to a calendar, each task can also represent an event.

The user can choose its type between a fixed date and an open interval. The prior begins and ends exactly at the given dates, whereas the latter type describes a task more loosely by its earliest start and/or latest finish (deadline). In order to support coordination, multiple users can be invited and associated with tasks. Collaborative tasks are also supported as documents and references to websites can be attached to shared tasks.

The *Ubidoo* task planner is running on a Web-server[2] and can be accessed everywhere via the internet. The user interface for desktop PCs is split in three frames, as shown in Figure 1. On the left hand side, the users' tasks and subtasks are shown in a foldable tree structure. By selecting a task from the tree on the left, a summary of its details (dates, location and description) is shown and icons allow moving or deleting this task. In the frame to the right, various properties of the selected task can be edited which are arranged into six tabs. The bottom frame provides links to the user profile and a traditional calendar view. The user can also manually choose a location for an adapted task view ("here and now") that will be described later in more detail. For mobile devices, a reduced interface without frames and tabs is also available.

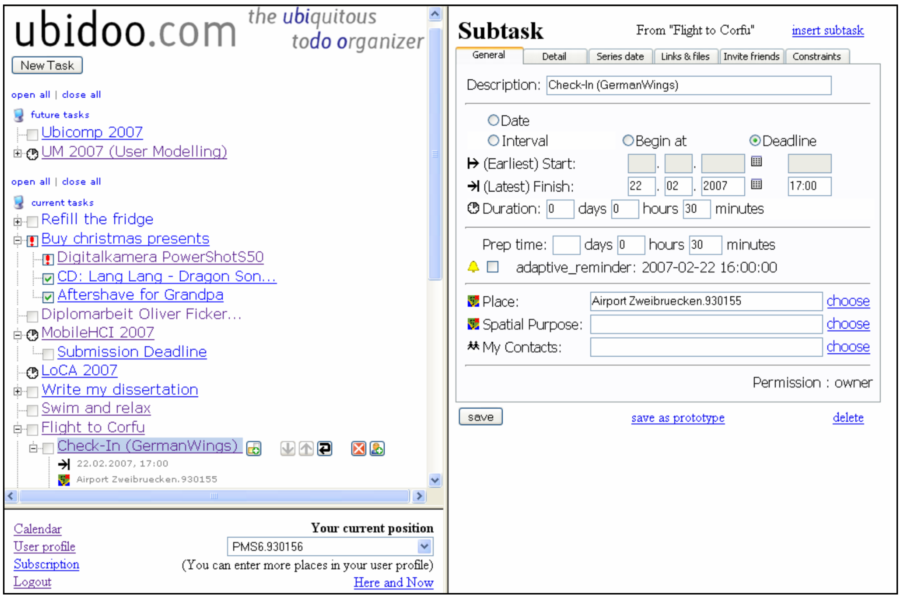

Figure 1: The Web-based user interface for the ubiquitous task planner *Ubidoo*.

Adaptive task reminder

Usually, calendars in mobile devices offer an alarm function that reminds the user on events at a fixed time and date. Setting the proper time for an alarm requires the user

[2] The **ubi**quitous to-**do o**rganizer. Website: http://www.ubidoo.com/

to consider everything that needs to be done and prepared before the event, most importantly how to go there. Some events might require the user to go home first and to dress accordingly or to pickup some things, which takes additional time. The user has to plan and foresee the whole situation under uncertainty. However, often the situation changes in an unpredictable manner and we will not be where we have planned to be. Thus the alarm will remind us too early, unnecessarily distracting us from our work, or worse, remind us too late and we can't reach the event timely.

Our ubiquitous task planner addresses this issue through an adaptive reminder, which continuously recalculates the best time for the alarm based on the current geographic distance between the user and the event. In addition, a general preparation time can be specified that will be considered by the reminder. Tasks can be localized by specifying a location from the spatial ontology in *UbisWorld* (see the "Place" attribute of a task in the "General" tab in Figure 1). As the location of the user (according to the user profile) changes the task planner updates the distance between the user and all events using route planning web services. The PTV *eRoute* service returns the estimated driving time between two locations. In addition, we consider the walking distances to the next parking lot or bus stop. The concept of web services makes it relatively easy to include other services in our planner. Semanic web technologies like WSDL (Web service description language) and SOAP (Simple object access protocol) let us send the same requests for routes to a web service for public transport systems, such as busses and trains. The planner would than choose between car and bus based on the user's preferences.

For the sake of simplicity, the actual reminder functionality has been implemented by sending an email to the user that includes the description of the event and a link to the task planner. This method requires a mobile device with an email client, but one could basically also use an SMS gateway.

Here-and-now: an adaptive task view

We have seen how a small improvement like the location-aware reminder can ease the use of calendars and alarms in the context of mobile devices, but even such a reactive calendar is still far from the goal of a 'personal assistant'.

By adding more general knowledge about the relationship between tasks, activities and locations through the use of ontologies, we can make further progress towards this goal. The 'here-and-now' view filters the user's tasks according to the current situation, which depends on time and location of the user, so that it suggests only those tasks which can actually be worked on. The time horizon of this view is limited by the next binding event to occur (with a fixed beginning or deadline), and to be more precise, by the time of this event's adaptive reminder.

Regarding location, the adaptive view does not simply match the location of the task with the user; similar to the adaptive reminder, the system performs a route calculation to estimate the actual distance and time to get there. Furthermore, the system considers the opening hours of a place, so that the planner can filter those locations which are currently closed. In combination with route knowledge, it can even find out if the location will close before the user can actually get there.

Likewise, the system considers the purpose of locations in terms of activities on a semantic level. If the user associates a task in the planner with activities from the ontology such as *shopping for electronics* instead of a certain location instance or store, the adaptive view automatically performs a search within UbisWorld's spatial elements and suggests the closest suitable location nearby for this task. Depending on the current time and location of the user, the system might suggest different places for the same task.

Figure 2 shows the same tasks as seen from two different locations. On the left image, we see the adapted view for Saarbrücken. The first task, swim and relax, is associated with the spatial purpose of *waterworld* and the planner suggests a place called *Calypso* in Saarbrücken that is located 7 kilometers (or 11 minutes by car) away from the current location of the user (in his office). A click on the *R* icon opens a route planner that provides a map and driving directions. The second task reminds the user to buy olives and milk. For the first item, a store called Saarbasar is set by the user as location, whereas for the second item the general activity *shopping.food* has been specified so that the planner automatically suggests the Globus supermarket. The last task is a reminder to catch the flight from Zweibrücken, it takes 31 minutes to go there from the office. Now compare the times and distances with the adaptive view for Munich on the right. The olives do not appear, since the store is too far away. For the milk, a different store in munich is suggested and also the waterworld has been substituted by a local place. The adaptive reminder for the airport will happen earlier, since the estimated travelling time is now more than 4 hours.

Figure 2: The location-adaptive task view as seen in Saarbrücken (left) and Munich (right).

Altogether, the adaptive view has several benefits for the user. It generally reduces the user's cognitive load that is required to browse through the complete task tree (equivalent to a calendar and to-do list). On mobile devices, it helps to save valuable screen space on the limited displays. Likewise, it helps to proactively present only relevant information to the user on public displays, which often do not allow for interaction (no touchscreen or mounted out of reach to prevent vandalism). We have developed a public display infrastructure called *IPlay Blue* [Schöttle, 2006] for the task planner that uses non-interactive, wall-mounted displays. The users are automatically recognized by scanning for their mobile Bluetooth devices.

4 Ubiquitous Domain Modelling

Our ubiquitous task planner is based on a centralized SQL database that includes the task descriptions of all its users. The user-adaptive and location-aware features are realized by linking user and task descriptions to the UbisWorld knowledge base. In the following we introduce UbisWorld and describe our extensions to the ontology.

User modelling

The UbisWorld knowledge-base has been designed to model and query the characteristics of a user, including their activity, as well as the environmental context. It also provides a symbolic spatial model to express location as described above. One goal of UbisWorld was to provide a flexible web-based model that can be inspected by the user through a convenient user interface and accessed by applications using semantic web technologies. In order to represent the user's current location, tasks or preferences, we use the general user model ontology GUMO (see www.gumo.org).
An important information for a navigation planer is for example the users's preference between car or public transport in certain situations. However since we do not want all systems to ask the user the same question several times, we use the new concept of ubiquitous user modeling which describes ongoing modeling and exploitation of user behavior with a variety of systems that share their user models. In our case, the user model is shared between the task planner and the public display infrastructure [Schöttle, 2006] which comprises several applications like a situated shopping list and personalized advertisements. The public displays' user interfaces are implemented through PHP scripts and Web-browser components. The ubiquitous user model provides the users' Bluetooth ID for recognition and a personal graphical icon which allows to address the user in an anonymized manner.

Location modelling

The UbisWorld ontology includes spatial concepts to represent locations in different granularities like country, city, building and room which can be seen in Figure 3. Location instances have a profile that is similar to a user profile and consists of situational statements. The spatial relation of *inclusion* is used to express the connection between rooms and buildings, so that the general attributes of a room can be inferred or inherited from the building's profile.
The profile of a building contains information about its postal address and geographic coordinates, the opening hours if applicable and the general purpose of the building in

terms of activities that can be done there. Some example purposes are depicted in the upper right of Figure 3, and an instance of a location profile is depicted in Figure 4. Further information is given about the nearest bus stop and parking lot so that additional walking distances can be estimated which are not considered by a route planner based on street maps. For fine-grained geometric modelling of pedestrian routes and buildings, we use the Yamamoto [Stahl 2006] toolkit. It allows for representing indoor environments in multiple levels and includes a routing component that provides shortest paths for pedestrians. Using the spatial *map reference* relation, location instances in UbisWorld can be linked to their geometric representation in the Yamamoto models. Likewise, the geometric building and room models include symbolic references to their counterparts in UbisWorld. In combination, UbisWorld and Yamamoto form a so-called hybrid location model as described in [Stahl and Heckmann, 2004].

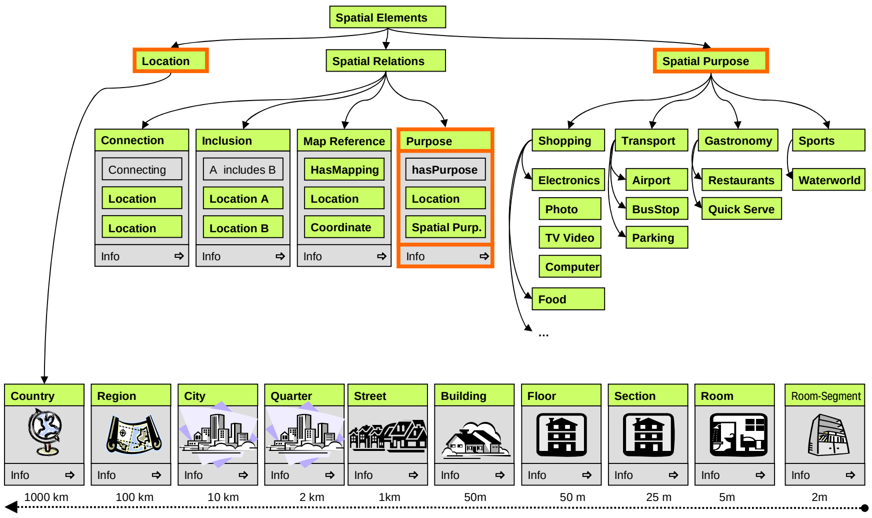

Figure 3: Spatial concepts in UbisOntology: locations, spatial relations and spatial purposes.

Task modelling

For task modelling, we use a separate *Mysql* database which uses *UbisIdentifiers* as references to elements in the UbisWorld knowledge base. UbisIdentifiers are strings that consist of a descriptive, human-readable part and a six-digit unique identifier, separated by a dot. Each task can be associated with a parent task and one or many users of the UbisWorld. Tasks can be localized either directly by referencing spatial elements within UbisWorld (e.g. `Building E11.400040`), or indirectly by choosing them in terms of activities from the spatial purposes provided by the ontology (e.g. `Education.300060`). Through these references, the task planner can retrieve the address details and opening times for locations from UbisWorld's location model.

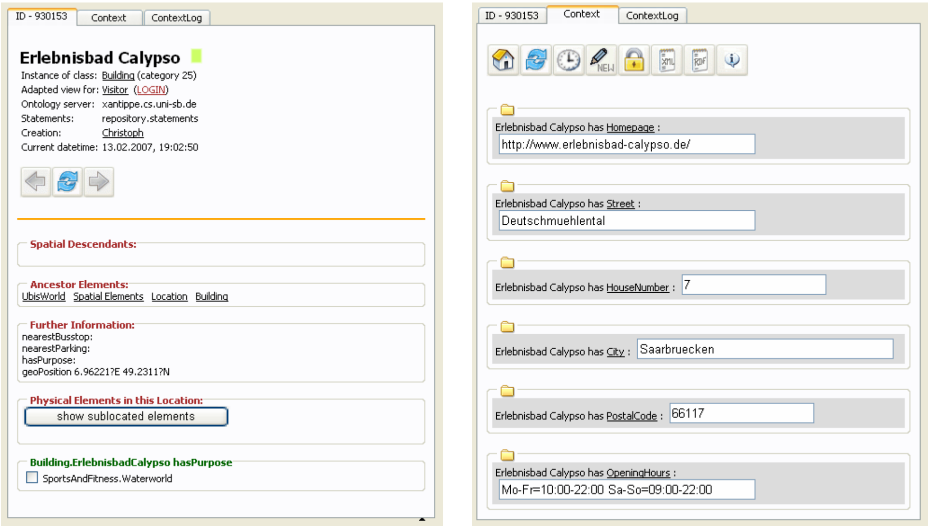

Figure 4: The location profile for a waterworld called Calypso in UbisWorld.

5 Localization

As stated in the Related Work section, we can use LORIOT to determine the user's position. Since LORIOT is an egocentric positioning system, the user can give permission to the system to send the current position to UbisWorld, where his user model will be updated accordingly. The disadvantage of LORIOT is that the user has to wear an additional device (a PDA) that is furthermore extended by non-standard hardware (the active RFID reader). To overcome this disadvantage we thought of another way to determine the user's position, which can be used in conjunction with public displays and the mobile phone of the user: We equipped all of our public displays with standard Bluetooth dongles. A small program is running in the background of each display that constantly scans its vicinity for active Bluetooth devices, e.g. mobile phones. Every Bluetooth device has its own unique address and a user can choose to store this address in his personal user model in UbisWorld. Each display knows about its own location and by a lookup of the received Bluetooth addresses in UbisWorld, it can infer which users are standing in front of it. According to the definition in the Related Work section, this is an exocentric localization method. Its precision is very coarse and depends on the range of the used Bluetooth dongles.
To extend the idea of the Always Best Positioned paradigm, we are currently researching on a combination of standard sensors that can be found in state-of-the-art mobile phones: GSM radio cells, Bluetooth and WiFi. The information to which radio cell a mobile phone is currently connected to, can give a very rough estimation of the location of a user. For example, the campus of Saarland University is completely covered by one cell. Detecting that cell gives evidence that the user is currently on the

campus. Also there are a lot of different WiFi networks set up on the campus, so that detecting our laboratory's own WiFi network can refine the estimated position to "somewhere near or in the lab". With all the Bluetooth enabled displays and computers in the lab, detecting certain stationary Bluetooth devices can give evidence on a room level. The number of the detected Bluetooth devices can also be used to infer the situational context of the user. A very high number of detected mobile phones can give evidence that the user is currently having lunch in the refectory, a low number that he is in a meeting and an increased number that he is visiting or giving a lecture.
We plan to use machine-learning methods to accomplish this, so that every user can train the system to their own needs. An application on the mobile phone will constantly read all its sensors and guess the current location. If the user wants to use the location-based task planner, the application will propose the estimated location (or no location at all, if the system is completely untrained). The user can then refine this proposal, where the possible semantic descriptions of the location are taken from UbisWorld. The application will then try to attach this refined location to its current sensor readings, where it can use the hierarchical spatial ontology of UbisWorld to assign the different layers of the ontology to the different precisions of the sensors.

6 Outlook and Conclusions

We have presented a web-based ubiquitous task planner that combines a calendar with a to-do list. Each task can be associated with a location or a general activity using the UbisWorld ontology and knowledge base. We employ a route planning web service to estimate geographical distances based on street maps so that the planner can adapt its reminder functionality and task view to the current location of the user. Semantic knowledge allows the planner to automatically suggest alternative places. We have adopted a pure reactive approach for our system, since we believe there is too much uncertainty involved in everyday life for a reasonable planning approach.
The mobile internet is still expensive to use today, and it might seem like a disadvantage to store ones calendar on a server instead of the mobile device itself. Also privacy issues might scare people off from using such a service. On the other hand, web-based email services are very popular and preferred by many users against local mail clients for the sake of ubiquitous access and less configuration. We believe that the mobile internet will become affordable soon through flat rates and that server-based PIM solutions will gain popularity, since they are device independent and avoid synchronization problems, which are quite common today. Another major advantage is the easy sharing of tasks and events between friends and colleagues.
In the context of user modelling, our task planner has demonstrated how applications can gain intelligent behaviour through adaptation based on a ubiquitous user- and location model. The UbisWorld web interface provides the necessary building blocks to choose locations and activities from the ontology by a few clicks. However, the current implementation of the selection tree does not scale with the size of the ontology and needs to be improved through *AJAX* Web technology.
For the future, we will use the ubiquitous task planer as a platform for more specialized user assistance applications in ubiquitous computing scenarios. We are

going to integrate the task planner into an intelligent kitchen environment to serve as a shopping aid. As a user drops something in the augmented waste basket, the item will be recognized through *RFID* technology and will be automatically added to the shopping list so that it appears in the supermarket-adapted viewing context.
Further research is needed to shed light on the lifecycle and dependencies between tasks in the to-do list and events in the calendar. It is typical that collaborative tasks become events, for example if we make an agreement to go shopping after work. On the other hand, calendar events like a birthday party might spawn new tasks in the to-do list like deciding and shopping for a present. Such transitions are currently not well supported by the task planner's user interface and need to be studied in more detail.

References

1. [Brandherm and Schwartz, 2005] Brandherm, B. and Schwartz, T. (2005). Geo Referenced Dynamic Bayesian Networks for User Positioning on Mobile Systems. First International Workshop on Location- and Context-Awareness. In: T. Strang and C. Linnhoff-Popien (Eds.): LoCA 2005, LNCS 3479, pp. 223 – 234, 2005. Springer-Verlag Berlin Heidelberg.
2. [Fensel et al., 2003] Fensel, D., Liebermann, J., and Wahlster, W., editors (2003). Spinning the Semantic Web. Bringing the World Wide Web to Its Full Potential. MIT Press, Boston.
3. [Glaubitt, 2006] Glaubitt, U. 2006. Realisierung eines situationsabhängigen und reaktiven Kalenders für mobile Benutzer. Bachelorarbeit an der Technischen Universität Darmstadt, Fachbereich Informatik, Fachgebiet Telekooperation, Prof. Dr. Max Mühlhäuser.
4. [PTV AG, 2004a] PTV Planung Transport Verkehr AG (2004). eLocateServer 3.0 Documentation. A server for geocoding address data and its request object. PTV AG, Karlsruhe.
5. [PTV AG, 2004b] PTV Planung Transport Verkehr AG, (2004). eRouteServer 3.0 Documentation. A server for calculating routes and its request object. PTV AG, Karlsruhe.
6. [Schöttle, 2006] Schöttle, F. (2006). IPlay Blue: Interaktive Displays mit Bluetoothbasierter Benutzererkennung. Fortgeschrittenenpraktikum am Lehrstuhl für Künstliche Intelligenz, Prof. Wolfgang Wahlster, Universität des Saarlandes.
7. [Schwartz et al., 2005] Schwartz, T., Brandherm, B. and Heckmann, D. (2005). Calculation of the User-Direction in an Always Best Positioned Mobile Localization System. Proceedings of the International Workshop on Artificial Intelligence in Mobile Systems (AIMS 2005), Salzburg, Austria, 2005
8. [Stahl and Haupert, 2006] Stahl, C. and Haupert, J. 2006. Taking Location Modelling to new Levels: A Map Modelling Toolkit for Intelligent Environments. 2nd International Workshop on Location- and Context-Awareness. In: M. Hazas, J. Krumm, and T. Strang (Eds.): LoCA 2006, LNCS 3987, pp. 74 – 85, 2006. Springer-Verlag Berlin Heidelberg.
9. [Stahl and Heckmann, 2004] Stahl, C. and Heckmann, D. (2004). Using semantic web technology for ubiquitous location and situation modeling. The Journal of Geographic Information Sciences CPGIS: Berkeley, 10(2):157–165.
10. [Studer et al., 2003] Studer, R., Hotho, A., Stumme, G., and Volz, R. (2003). Semantic web - state of the art and future directions. Künstliche Intelligenz, 3(3):5–9.

Author Index