International Workshop

on

Semantic Web Personalization

Budva, Montenegro,

June 12, 2006

Co-located with the

3rd European

Semantic Web Conference

Preface

The ultimate goal of the Semantic Web is to enable applications which offer enhanced and efficient possibilities for end users to benefit from electronically stored information. The vision of a Semantic Web, *in which information is given a well-defined meaning, better enabling computers and people to work in cooperation* already stresses the importance of efficient end user support for accessing and working with Web information. However, current development in the Semantic Web focuses on formalisms, languages, reasoning and the development of according technological frameworks, so to speak the first part of the vision. These technologies shall constitute an environment capable of enabling enhanced, efficient and user-centered applications, thus enabling the second part of the vision, and the goal of the Semantic Web.

This workshop brought together researchers and practitioners in the fields of Semantic Web technologies and personalization in order to discuss the emerging possibilities of realizing personalization in a Semantic Web. As personalization is not a new topic at all, the workshop's goal was especially to identify needs for personalization in the Semantic Web, but also experiences on personalized systems and how personalization in the Semantic Web can benefit and learn from these experiences. Furthermore, of course, first applications and prototypes offering the users personalized experiences were proposed and discussed.

Accepted papers for this workshop focused on three thematic issues:

- Reasoning and rules for user modeling and personalization in the Semantic Web
- Acquisition and application of user profiles
- Architectures enabling Personalization in the Semantic Web

We would like to thank all authors of the workshop for their contributions and inspiring discussions, and the organizing and program committee of this workshop, which made our job as organizers very enjoyable and smooth.

Budva, June 12, 2006

Makram Bouzid, Motorola, France
Nicola Henze, University of Hannover

Organization

Organizing Commitee

Lora Aroyo, Technical University of Eindhoven, Netherlands
Makram Bouzid, Motorola, France
Bruno Defude - Institut National des Télécommunications, France
Vania Dimitrova, University of Leeds, UK
Olaf Droegehorn, University of Kassel, Germany
Nicola Henze, University of Hannover, Germany
Massimo Marchiori, University of Venice, Italy

Program Commitee

Pragya Agarwal, UCL, UK
Hamid R. Arabnia, The University of Georgia, USA
Lora Aroyo, Technical University of Eindhoven, Netherlands
Matteo Baldoni, University of Torino, Italy
Cristina Baroglio, University of Torino, Italy
Christian Becker, University of Stuttgart, Germany
Claudia Buzzi, Institute of Informatics and Telematics, Italy
Frannçois Carrez, Alcatel, France
Monica Crubezy, Stanford Medical Informatics, USA
Rose Dieng-Kuntz, INRIA Sophia Antipolis, France
Vania Dimitrova, University of Leeds, UK
Serge Garlatti, ENST Bretagne, France
Dominik Heckman, DFKI, Germany
Nicola Henze, University of Hannover, Germany
Pascal Hitzler, University of Karlsruhe, Germany
Paola Hobson, Motorola, UK
Geert-Jan Houben, Vrije Universiteit Brussel, Belgium
Anthony Jameson, International University in Germany
Judy Kay, University of Sidney, Australia
Yiannis Kompatsiaris, CERTH/ITI, Greece
Alexander Kroener, DFKI, Germany
Massimo Marchiori, University of Venice, Italy
Bamshad Mobasher, DePaul University, USA
Klaus Moessner, University of Surrey, UK
Viviana Patti, University of Torino, Italy
Myriam Ribière, Motorola, France
Alfons Salden, Telematica Instituut, Netherlands
Fabrizio Silvestri, Information Science and Technology Institute, Italy
Samir Tata, Institut National des Télécommunications, France
Julita Vassileva, University of Saskatchewan, Canada

Table of Contents

PreDiCtS: A Personalised Service Discovery and Composition Framework

Charlie Abela, Matthew Montebello

Department of Computer Science and AI
University of Malta
{charlie.abela, matthew.montebello}@um.edu.mt

Abstract. The proliferation of Web Services is fostering the need for applications to provide more personalisation during the service discovery and composition phases. An application has to cater for different types of users and seamlessly provide suitably understandable and refined replies. In this paper, we describe the motivating details behind PreDiCtS[1], a framework for personalised service discovery and composition. The underlying concept behind PreDiCtS is that, similar service composition problems could be tackled in a similar manner by reusing past composition best practices. These have to be useful and at the same time flexible enough to allow for adaptations to new problems. For this reason we are opting to use template-based composition information. PreDiCtS's retrieval and refinement technique is based on conversational case-based reasoning (CCBR) and makes use of a core OWL ontology called CCBROnto for case representations.

Keywords: CCBR, Ontologies, Semantic Web, Web services

1. Introduction

Reusability and interoperability are at the core of the Web Services paradigm. This technology promises seamlessly interoperable and reusable Web components that facilitate rapid application development and integration. When referring to composition, this is usually interpreted as the integration of a number of services into a new workflow or process. A number of compositional techniques have been researched ranging from both, manual and semi-automatic solutions through the use of graphical authoring tools [18], [19], to automated solutions based on techniques such as AI planning [17] [20] and others.

The problem with most of the composition techniques mentioned above is three fold (i) such approaches attempt to address service composition by composing web services from scratch, ignoring reuse or adaptation of existing compositions or parts of compositions, (ii) it is assumed that the requester knows exactly what he wants and

1 This research has partially been funded by the European Commission and by the Swiss Federal Office for Education and Science within the 6th Framework Programme project REWERSE number 506779

how to obtain it and (iii) composing web services by means of *concrete* service interfaces leads to tightly-coupled compositions in which each service involved in the chain is tied to a web service instance. Using this approach for service reuse, may lead to changes in the underlying workflow which range from slight modifications of the bindings to whole re-designing of parts of the workflow description. Therefore in our opinion, services should be interpreted at an abstract level to facilitate their independent composition. [10] adds, "*abstract workflows capture a layer of process description that abstracts away from the task and behaviour of concrete workflows*", and this allows for more generalisation and a higher level of reusability. A system can start by considering such abstractly defined workflow knowledge and work towards a concrete binding with actual services that satisfy the workflow.

To make effective reuse of such abstract workflow definitions one could consider CBR, that is amenable for storing, reusing and adapting past experience for current problems. Nevertheless CBR restricts the user to define a complete problem definition at the start of the case-retrieval process. Therefore a mixed-initiative technique such as CCBR [3] is more appropriate since it allows for a partial definition of the problem by the user, and makes use of a refinement process to identify more clearly the user's problem state.

In summary we have identified the following motivating points:

1. Reusability of compositions has the advantage of not starting from scratch whenever a new functionality is required.
2. For effective reusability a higher level of abstraction has to be considered, which generalises service concepts and is not bound to specific service instances.
3. Personalisation of compositions can be achieved by first identifying more clearly the user's needs and then allowing for reuse and adaptation of past compositions based on these needs prior to binding with actual services.

The goal of this work is to present, the motivation behind, and prototype of PreDiCtS, a framework which allows for personalisation of service discovery and composition through the reuse of past composition knowledge. One could say that we are trying to encode and store common practices of compositions which could then be retrieved, reused and adapted through a personalisation technique. The solution we propose in PreDiCtS has two phases.

For the first phase, which we call the *Similarity Phase*, we have adopted a mixed-initiative technique based on CCBR. This provides for the personalisation process. Given a new problem or service composition request, this approach allows first to retrieve a ranked list of past, similar situations which are then ranked and suggested to the requester. Through a dialogue process the requester can decide when to stop this iterative-filtering phase, and whether to reuse or adapt a chosen case. Case definition is through an OWL-based ontology which we call CCBROnto [2] and which provides for the description of context, problem and solution knowledge At present PreDiCtS allows for case creation and retrieval (adaptation is in the pipeline) and once a case (or set of cases) is retrieved, it can be presented to the next phase, which we call the *Integration Phase* where a mapping is attempted, from the features found in the chosen solution, to actual services found in a service registry. Due to space restrictions this is dealt with in a future paper.

The rest of this paper is organized as follows. In Section 2 we will give some brief background information on CCBR. Then in Section 3 we will give an overview of the OWL case ontology, CCBROnto. In Section 4 we will present the architecture of PreDiCtS and some implementation details mainly focusing on the case-creator and case-retriever components. After which we present the last section with future work and concluding remarks.

2. Conversational Case-Based Reasoning

Case-Based Reasoning is an artificial intelligence technique that allows for the reuse of past experience to solve new problems. The CBR process requires the user to provide a well-defined problem description from the onset of the process. But users usually cannot define their problem clearly and accurately at this stage. On the other hand, CCBR allows for the problem state to be only partially defined at the start of the retrieval process. Eventually the process allows more detail about the user's needs to be captured by presenting a set of discriminative and ranked questions automatically. Depending on the user's supplied answers, cases are filtered out and incrementally the problem state is refined. With each stage of this problem refinement process, the system presents the most relevant solutions associated to the problem. In this way the user is kept in control of the direction that this problem analysis process is taking while at the same time she is presented with solutions that could solve the initial problem. If no exact solution exists, the most suitable one is presented and the user is allowed to adapt this to fit her new requirements. Nevertheless, this adaptation process necessitates considerable domain knowledge as explained in [4], and is best left for experts.

One issue with CCBR is the number of questions that the system presents to the user at every stage of the case retrieval process. This issue was tackled by [11] which defined question-answer pairs in a taxonomy and by [1] through the use of knowledge-intensive similarity metrics. In PreDiCtS we have adapted the former method[2] since a QA pairs taxonomy is defined to be an acyclic directed graph in which nodes are related to other nodes through parent-child relations and it is assumed that a node subsumes all its descendent nodes. This is very similar to how classes in OWL are related via the *subClassOf* relation and this fits well with the underlying case structure that we use in PreDiCtS.

3. CCBROnto

CCBROnto is an important component of PreDiCtS since it provides for (i) case and question-answer pair definitions, and (ii) the association of domain and case-specific knowledge. In CCBROnto the topmost concept is a *Case*. Its basic components are defined by the *CaseContext*, *Problem* and *Solution* classes. In [8] context is defined as "*any information that can be used to characterize the situation of an entity. An entity*

[2] Whenever we refer to this taxonomic theory we will be referring the work done by Gupta

is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves". We fully agree with this definition and in the *CaseContext*, we have included knowledge related to the case creator, case history, ranking and case provenance. We have considered ideas presented in [6], [7] and [15] which discuss the importance of context in relation to Web Services and stresses on the importance of the use of context in CBR, especially when cases require adaptation. Such context knowledge makes it possible to differentiate between users and thus the system could adapt cases accordingly. For example in the travelling domain, both going to a conference and going for a holiday may require similar services, such as hotel booking and flight reservation, though the use of a conference booking service is only required in the former. Thus, based on the contexts or roles of the users (a researcher the former and a tourist the latter) the CBR system can adapt the case knowledge to present cases that satisfy the requirements of both. A researcher can adapt the case for the tourist by including a suitable conference booking service.

In PreDiCtS we consider highly important such context knowledge since it helps to identify, why a case was created and by whom, together with certain aspects of case usage and its relevance to solving a particular problem. The *CaseCreator* provides a *Role* description that the creator associates himself with, together with a *foaf:Person* instance definition that describes who this person is. The motivation behind using foaf is to eventually be able to embed some level of reputation relevant to the person who created the case. The importance of this feature will become more visible and important when cases are shared.

The *CaseContext* also provides a place holder for *CaseHistory*. The knowledge associated with this feature is important when it comes to case ranking and usage, since it allows users to identify the relevance and usefulness of a case in solving a particular problem. It is also important for the case administrator when case maintenance is performed. Cases whose history indicates negative feedback may be removed from the case base. Case *Provenance* is also used in conjunction with reputation since it indicates a URL from where the case originated. Encapsulating such information in each case will help in maintaining a reliable case base.

The *Problem* state description in a PreDiCtS case is based on the taxonomic theory. Every problem is described by a list of QA pairs rather then a bag. This is required since QA pairs have to be ranked when they are presented to the user. Each *QAPair* is associated with a *CategoryName*, a *Question* and an *Answer* (see Fig.1). Each question has a textual description and is associated with a concept from the domain ontology through the *isRelatedTo* relation. We further assume that Answers could be either binary or nominal-valued. For this reason we have created two types of answer classes, *YesNoAnswer* and *ConceptAnswer*. The former is associated with a literal represented by either a *Yes* or a *No*. While the latter, requires an association with a concept in some domain ontology, through the previously mentioned *isRelatedTo* property. The motivation behind the use of this property is related to the taxonomic theory, which requires that QA pairs are defined in a taxonomy so that during case retrieval, the number of redundant questions presented to the requester is reduced. Thus during the case creation stage, each question and answer description is associated with an ontological concept defined in the domain of discourse. This is similar to how [1] associates ontology concepts with pre-defined questions. In

PreDiCtS we want to make use of such <concept-question> association so that questions and answers are implicitly defined in a taxonomy. This association is also important when similarities between QAPairs and between cases are calculated.

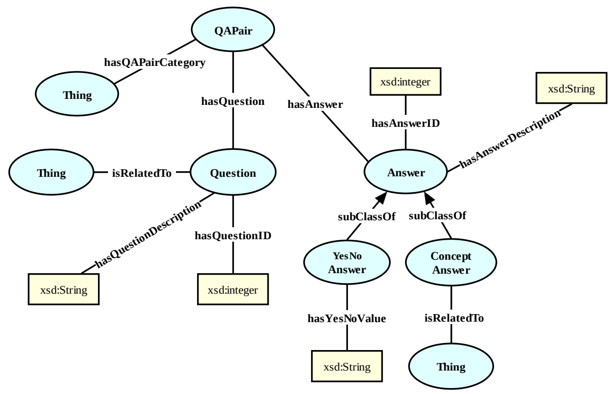

Fig.1: CCBROnto Problem structure

The Solution in PreDiCtS provides a hook where composition templates can be inserted. The main goal behind such a structure is to be able to present abstract composition knowledge as solutions to the user's request and at the same time allow for more flexibility when searching for actual services. In fact each *Solution* is defined to have an *Action* which has a description and is*DefinedBy* an *AbstractTemplate*. A template can be sub-classed by any service composition description, such as that defined by OWL-S. An OWL-S template in this case is an intersection between a service, profile and process definitions.

4. PreDiCtS: implementation issues

As explained in other sections, the PreDiCtS framework allows for the creation and retrieval of cases in its *Similarity phase* (see Fig. 2). The respective components that perform these two tasks are the *CaseCreator* and the *CaseRetrieval*. PreDiCtS is written in Java and is developed in Eclipse. It uses a MySQL database to store the cases and makes use of both Jena and the OWL-S APIs.

The *Similarity phase* is triggered by the user whenever she requires knowledge related to past compositions. In PreDiCtS the user is not expected to know exactly which type of services or service composition are required but she is required to answer a set of questions such that the system identifies more clearly what is required. Given information related to the domain, the retrieval process is initiated whereby all questions in a taxonomy relevant to that particular domain are presented to the user. Given the set of questions to choose from, the user can then decide to answer some of these questions. Depending on the answers provided, the system will try to find cases

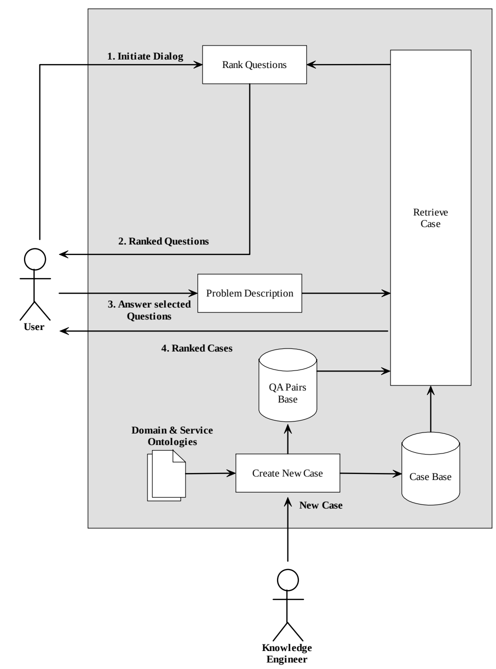

Fig.2: Taxonomic CCBR in PreDiCtS (adapted from Weber03)

in which questions where answered in a similar manner. A similarity measure is used to rank cases. The questions which are present in the retrieved cases but which are still unanswered, yet are related to the problem, are then presented in a ranked order to the user. The process continues until the user either chooses a case which includes a suitable solution or else, in absence of such a case, decides to adapt one of the most similar cases, thus further personalising the solution to her needs. The user can also opt to create a case from scratch to meet her requirements.

In the next sections we will describe the above mentioned PreDiCtS components by referring to an example from the health domain which deals with the combination of services that are used when a patient is admitted to hospital.

3.2 Case Creation

The CaseCreator component allows the expert user to add a new case to the case base. A case *c* can be defined as $c = (dsc, cxt, \{q_1a_3 \ldots q_ia_j\}, act, frq)$ where;
dsc is a textual description of the case.
cxt represents a set of context related features, such as *Role* and *CaseCreator* information based on foaf.
$\{q_1a_3 \ldots q_ia_j\}$ is a representation of the problem state by a set of question-answer pairs

act denotes the solution which is represented by service composition knowledge stored in an abstract template.
frq, is the frequency with which a case is reused.

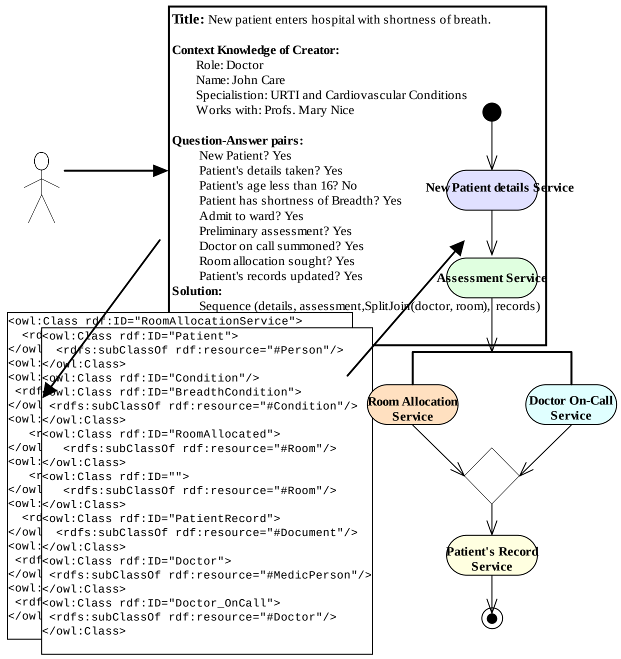

Fig. 3: Adding a new case

The example presented in Fig. 3 represents the combination of knowledge that is required to build a new case. PreDiCtS takes into consideration both domain and composition knowledge and combines them, based on the knowledge of the creator. In the example, the case creator is a Doctor (John) who specialises in URTI (Upper Respiratory Tract Infections) and cardiovascular conditions. The case in question represents the situation where a new patient, who is more then 16 years old, has entered hospital with shortness of breath. The creator enters context information about himself and any relations that he has with other persons. In this scenario, John has work relations with Professor Mary Nice. This information provides for a level of reputation in the expertise of the creator. The composition knowledge in this case represents a number of services that the hospital system wants to use to efficiently cater for patients entering hospital. This particular functionality is required to monitor the patient from the moment that he enters the hospital until he is comfortably stationed in a room.

To add service information to a case, the creator can use a visual component which is based on UML activity diagrams, though other representations, which are more user-friendly, are being considered. Each visual representation is mapped into a process model representation. In this work we use OWL-S as the underlying language for this representation.

A *service* definition in OWL-S is just a place holder for information relating the profile, process and grounding. We are not considering any grounding knowledge at this stage, since this will be tackled later on in the *Integration phase* when actual service bindings are sought. As regards the profile, we only consider that knowledge which is relevant and which is not tide to specific providers. The profile part of the template includes the definitions of *inputs* and *outputs*, *profilehierarchy* and references to the process and service components. The profile hierarchy is considered to be of particular importance since it represents a reference to the service domain knowledge, that is, it identifies the taxonomic location of a particular set of service profiles. We think that such ontologies will become increasingly more important in relation to best practice knowledge. The template also provides information related to how a number of service components are combined together. What is most important here, are the control constructs such as *Sequence*, *If-Then-Else*, and *Split* that determine the order of execution of the service components. These service components are defined through the OWL-S *Perform* construct which associates a particular service component with another by binding its outputs to another service component's inputs.

An important aspect of case-creation in CCBR is the addition of question-answer pairs since they are fundamental for the case retrieval process. Through PreDiCtS we allow the creator to either reuse existing QA pairs or create new ones. Textual questions are associated with concepts defined in ontologies and this provides an implicit taxonomic structure for QA pairs. Such association provides the possibility to reason about these concepts, and also to limit the number of questions to present to the user during the retrieval process. The taxonomic theory requires that each case includes the most specific QA pair from a particular taxonomy. Given the open-world assumed by ontologies on the Web, we assume that the knowledge (triples) associated with a set of QA pairs is closed by adapting the idea of a local-closed world defined by [12].

Adding a new case to the case base is mainly the job of the knowledge expert, nevertheless we envision that even the not so expert user may be able to add cases when required. For this reason we have used the same technique as that used by recommending systems and also adopted by [21], which allows case-users to give feedback on the utility of a particular case to solve a specific problem.

3.3 Case Retrieval

Similarity is based on an adaptation of the taxonomic theory, and is divided into two steps, similarity between question-answer pairs and an aggregate similarity to retrieve the most suitable cases. The prior, involves the similarity between the QA pairs chosen by the user and those found in a case. In the taxonomic theory two pairs are defined to be more similar if the one found in the case is a descendant (therefore more specific) of the other, rather then its parent (therefore more generic). Though we have adopted this similarity assessment metric, we take into consideration that each QA pair is a set of triples or rather an acyclic directed graph. Thus similarity between QA pairs is based on the similarity between two such graphs. The taxonomic similarity is calculated as follows:

$$sim\ (C_{Q1}, C_{Q2}) = \begin{cases} 1 & \text{if } C_{Q2} \subseteq C_{Q1} \\ (n+1-m)/(n+1+m) & \text{if } C_{Q1} \subseteq C_{Q2} \\ 0 & \text{otherwise} \end{cases}$$

where, C_{Q1} and C_{Q2} are concepts

n= number of edges between C_{Q1} and the root i.e. the concept *Thing*

m= number of edges between C_{Q1} and C_{Q2}

Having calculated such similarity between QA pairs then an aggregate similarity metric is used to calculate the overall similarity between the user query Q_U and a case problem description, P_C. This aggregate similarity is calculated as follows:

$$sim\ (Q_U, P_C) = \frac{\sum_{i \in Q_U, j \in P_C} sim(C_{Q\,i}, C_{Q\,j})}{T}$$

where, T in the original taxonomic theory represents the number of taxonomies, here it represents the number of different ontologies that are used to define the concepts found in the QA pairs.

We are also looking at other research work which provides for similar measures, in particular work related to ontology-based similarity measures [13], [16] and semantic distance [5], [14]. Such work is important since it does not only consider the taxonomic similarity between concepts but also similarity based on the number of relations and attributes associated with the concepts.

4. Conclusion

In this paper we presented the main concepts behind PreDiCtS. The use of CCBR as a pre-process to the service discovery and composition is promising since it provides for inherent personalisation of the service request and thus as a consequence also more personalised compositions. We also presented CCBROnto as a case definition language which allows for seamless integration between CCBR and the Semantic Web, by providing reasoning capabilities about concepts within the case definitions. Nevertheless, there is still a lot to be done, especially where it comes to case generation and evaluation. A case base can only be evaluated effectively if the number of cases is large. We are infact considering the possibility of generating cases, for experimental purposes, by extracting the required template knowledge from already available service descriptions and then adding context information and QA pairs. Other issues for future consideration include the design of the questions and the way in which they are associated with ontology concepts, the effective evaluation of the similarity metrics used with an eye on work being done on semantic similarity and also the inclusion of an adaptation component. The latter will provide for more personalisation of the solutions presented by PreDiCtS and thus also of the services that will be presented to the user.

References

1. A.Aamodt, M. Gu, A Knowledge-Intensive Method for Conversational CBR, Proc. ICCBR'05, Chicago, August 2005
2. C. Abela, CCBROnto, http://www.semantech.org/ontologies/CCBROnto.owl
3. D.W Aha, L.A. Breslow, H. Muñoz-Avila, Conversational case-based reasoning. Applied Intelligence, 14, 9-32. (2001).
4. M.S. Aktas, D.B. Leake. et al, A Web based CCBR Recommender System for Ontology aided Metadata Discovery, GRID'04
5. A. Bernstein, et al, Simpack: a Generic Java library for Similarity Measures in Ontologies, University of Zurich, August 2005.
6. F. Bry et al, Context Modeling in OWL for Smart Buildings, Proc. of GvB2005.
7. M. d'Aquin, et al, Decentralized Case-Based Reasoning for the Semantic Web, Proc. ISWC 2005
8. A. Dey, Understanding and Using Context, in proceeding of Personal and Ubiquitous Computing, issue on Situated Interaction and Ubiquitous Computing, Feb 2001.
9. B. Diaz-Agudo et al, On Developing a Distributed CBR Framework through Semantic Web Services, Workshop on OWL: Experiences and Directions, Galway'05
10. A. Goderis et al. Seven bottlenecks to workflow reuse and repurposing. 4th Int. Semantic Web Conference, Galway, Ireland, 6-10 Nov. 2005
11. K. Gupta, Taxonomic Conversational Case-Based Reasoning, Proceedings of the 4th International Conference on Case-Based Reasoning, 2001
12. J. Heflin, H. Muñoz-Avila, LCW-Based Agent Planning for the Semantic Web, AAAI Workshop WS-02-112002
13. M. Hefke, A Framework for the successful Introduction of KM using CBR and the Semantic Web Technologies, I-Know 2004
14. N.Henze, M. Herrlich, The Personal Reader: A Framework for Enabling Personalization Services on the Semantic Web, Proc. of ABIS 04, Berlin, Germany.
15. Z. Maamar et al, Context for Personalised Web Services, 38th Hawaii International Conference on system Science, 2005
16. A. Maedche, V. Zacharias, Clustering Ontology-based Metadata in the Semantic Web, Joint Conferences (ECML'02) and (PKDD'02), Finland, Helsinki, 2002
17. J. Peer, A POP-based Replanning Agent for Automatic Web Service Composition ESWC'05
18. J. Scicluna et al, Visual Modelling of OWL-S Services, IADIS International Conference WWW/Internet, Madrid Spain, October 2004
19. E. Sirin et al, Semi-automatic composition of web services using semantic descriptions, in ICEIS 2003, Angers, France, April 2003
20. E. Sirin et al. HTN planning for web service composition using SHOP2. Journal of Web Semantics, 1(4):377-396, 2004
21. B.Weber, S. Rinderle, W. Wild, M. Reichert, CCBR–Driven Business Process Evolution, Proc. ICCBR'05, Chicago, August 2005

Personalizing web surfing with semantically enriched personal profiles

Anupriya Ankolekar, Denny Vrandečić

Institute AIFB, University of Karlsruhe, Germany
{anupriya,denny}@aifb.uni-karlsruhe.de

Abstract. Personalization mechanisms on the web today are clumsy and obtrusive, because users need to log in to multiple websites and enter their personal information and preferences separately for each. In addition, the user profile is different for each website and cannot be combined with other information on the web. Using Semantic Web technologies, in particular FOAF, we can identify the person browsing to the website. In this paper, we propose an extension of the HTTP `GET` method to include a new parameter that points to the URL of the user's FOAF file. This simple, but powerful extension enables the web server to use information contained in the person's FOAF file to personalize the web pages returned. We also present a proof-of-concept implementation by customizing our institute welcome page using a visitor's FOAF file.

1 Introduction

No one knows you are a dog on the Internet, or so the saying goes. While this may have its advantages, letting websites know who you are enables them to personalize your web surfing experience, serving you information customized for your needs and preferences. Knowing the interests, activities, acquaintances or accessibility problems of a user, the web site can adapt the web pages it serves and offer a personalized, familiar and welcoming experience for the user. The personalizations may include content personalization (show me the things I am interested in), or link personalization (show me only those links that I am likely to click on) and structure/navigation personalization (reorder the website as I am likely to understand and search through it). Several studies have explored the use of personalization and knowledge of people's activities on a website to support web browsing and have found it to considerably enhance the browsing experience for users [15].

A key requirement for enabling such personalization is that the web server must be able to identify the person visiting the website and know her characteristics and preferences. Currently, the vast majority of sites attempt to personalize the user's browsing experience by requiring the user to create an account on the website and login every time a personalized service is used. *Amazon* and *Yahoo!*, for example, follow this model. However, this is quite painful as the user needs to remember multiple logins and usually go through a log-in procedure at every site, which disturbs the smooth flow of web browsing. What we really want is a seamless experience, where the person decides which profile she wants to adopt and then websites automatically recognize her and offer her customized content. Ideally, the user should not even be aware that she needs to identify herself to a website, but is naturally offered information relevant to her.

What we need therefore is a user-side personal profile that describes a person and her interests, affiliations and acquaintances in an open standard. Semantic Web [2] technologies provide exactly this in the form of a FOAF (Friend of a Friend) [3] file. In this paper, we propose a personalization mechanism that allows a web server to identify a user's FOAF file with every page request. The web server may then use the information in the FOAF file to personalize the page it returns, potentially retrieving additional information on the web in the process. This is where the advantages of using Semantic Web technologies become apparent. Since the FOAF file is in RDF [1], the information it contains can easily be combined with other RDF information on the web.

In the following, we first discuss the concrete details of our approach (section 2) and then related work on personalization mechanisms (section 3). We then demonstrate the feasibility of our approach through a proof-of-concept implementation in section 4, showing how the AIFB website personalizes its welcome page with information from a visitor's FOAF file. Of course, this is still a fairly simple use of the information. More sophisticated usage of the data in the visitor's FOAF file can enable a range of scenarios as discussed in section 5. Finally, we discuss some of the implications of our approach, in particular with respect to privacy in section 6, before summarizing the contributions of the paper in the concluding section 7.

2 Approach

We rely on a user's Friend Of A Friend (FOAF) [3] profile to personalize her browsing experience. The FOAF vocabulary is one of the most popular RDF vocabularies on the web and describes a person, in terms of several attributes of the person, such as homepages, affiliations, photographs and contact details, as well as specifying the acquaintances and friends of the person. Millions of FOAF files already exist, and are particularly popular within the blogging community.

The FOAF vocabulary definitions are written as RDF [1] statements. This allows software to process the FOAF information and follow links to the FOAF files of friends and acquaintances to gather as much information as it needs. Using RDF as a description format has another advantage: we can easily combine FOAF statements with statements from other RDF vocabularies, such as RSS[1] for describing blog feeds, and the Web of Trust (WOT) Schema, to describe signatures on RDF documents, or geographical location vocabularies. Thus, a FOAF file can contain all kinds of information about the user and point to files with more specific information about the user. The web server can use as much of the information as it understands, ignoring the rest. So the system allows

[1] http://en.wikipedia.org/wiki/RSS_(protocol)

for a very graceful performance degradation, and still enables it to express any information that could be useful for personalization.

To give web servers access to a user's FOAF profile, we extend HTTP's `GET` method to include a new parameter that points to the URL of the user's FOAF file. When the web server receives a `GET` request, it examines its parameters in order to determine which information to send back. At the most obvious level, this includes the web page to return. However, there is also an element of browser-personalization involved. For example, depending on which browser is requesting the page, Internet Explorer or a Mozilla browser, the web server can present different versions of the web page best suited to the browser. By extending the `GET` parameter to point to a user's FOAF file, the web server can fetch information about the user and customize the returned page for the user.

This is a fairly simple idea, but it is attractive for several reasons:

1. By relying on the HTTP protocol, we are using the lowest common denominator when it comes to web access, so this method can be used with minimal modifications by most websites.
2. This eliminates the need for the user to login to a site explicitly, with all its associated problems of remembering the login/password details, the time and effort required for the login procedure as well as the break in the smooth navigation through the web.
3. Personal information is within the control of the user rather than multiple websites, so the user can decide what information to expose. Unlike previous user passport methods [12], the FOAF files are simple to understand, straightforward to create, and based on an open standard format.
4. The FOAF file is passed with every HTTP `GET` call, enabling the user to change profiles within a given session.

3 Related work

There are several mechanisms used currently for achieving personalization on the web [9]. These include *server-side accounts*, which require the user to create an account on the website and log in to it when making use of personalized services, *cookies*, used for storing identification and user preferences on the user's machine, and *identity profiles*, such as *Microsoft Passport* [12], *AOL Screen Name*[2] or *OASIS Open Identity*, which provide a single sign-on for multiple services.

Server-side accounts are provided by most major websites and portals such as *Amazon* and *Google*, and cookies are used (often indiscriminately) by an even larger number of websites. However, there are distinct disadvantages to all of these approaches. To begin with, none of these mechanisms give fine-grained control to the user of the information he or she is presenting to a website. Server-side accounts usually require a standard list of information regardless of whether and how it is used subsequently. Furthermore, there is no single point of control for the user. With server-side accounts, user information is scattered all over

[2] `https://my.screenname.aol.com/`

the web, tied up in hundreds of websites for the typical web surfer. Cookies, simple site-defined key-value pairs, do store information on the user end, but since the format is defined by the web site, more often than not the cookies are meaningless to the user. Since cookies are site-specific user information, often the same information, such as the geographical location of the user, is stored in multiple cookies for different websites. Identity profiles do present an unified store of user information, but these are typically stored in proprietary formats at the vendor end, such as *Microsoft* or *AOL*, thus again compromising user control of her information. With our approach based on FOAF files, information is stored in open standards on the user end, allowing for user control of a centralized store of her information. In addition, since FOAF uses Semantic Web standards, FOAF information can easily be combined with other Semantic Web information, further enhancing its value.

Secondly, the majority of these personalization mechanisms do not store social information about friends, acquaintances and colleagues. Assuming the user has made extensive use of social community sites, such information is locked in multiple websites, such as *LinkedIn*[3], *OpenBC*[4] and *Yahoo!*[5]. This means that the user has to tediously enter the same people in multiple websites and the communities often have different kinds of contacts (for example, *OpenBC* is used extensively and and almost exclusively in Germany), which are not connected to each other. The FOAF files allow us to not only present a single-point for describing all contacts, but also enable combining this information with the web browsing behavior, so, for instance, a web site can tell you which of your friends visited the site recently.

Despite the single point of control, our approach does not eliminate the need for site-specific user profiles and preferences. Despite generally preferring large text sizes, a user may still want to specify that she prefers the standard text size on *Yahoo!*. Such information may still need to be stored at the web site end. However, our approach does mean that the user always has a set of default preferences and the site only needs to save information about site-specific preferences that differ from the default preferences. The primary criticism of user-centric identity management has been that it is not portable [11]. Personal information stored on one computer cannot be easily transferred to my mobile phone. This is a non-issue in our case, since the identity as FOAF file is always accessible over the web. Of course, this can lead to security and access control issues, which we refer to in the discussion (section 6).

4 Implementation

Extending the HTTP `GET` request requires modifications on both the client and the server end. At the client end, the web browser must be extended to include a link to the user's FOAF file when sending a `GET` request to a web server. At

[3] `https://www.linkedin.com/`
[4] `https://www.openbc.com`
[5] `http://www.yahoo.com`

the server end, the web server must be able to understand the HTTP protocol extension and read the FOAF file pointed to. In the following, we discuss our proposed HTTP protocol extension, the implementation of the extension for the Mozilla-based Firefox web browser[6] and the modification of the AIFB Semantic Portal[7] [7] to make use of the extension.

We extended the HTTP protocol in conformance with the HTTP specification RFC2616 [6]. We added a new line, a property `XFOAF`, that specifies the URL of the user's FOAF file [3]. A web server that is aware of this extension may then fetch the FOAF file, require additional resources if needed, and then process it appropriately. All other web servers can simply ignore the line, since according to RFC2616, unrecognized header fields should be ignored and must be forwarded transparently. Thus the user experience does not change in any way when visiting pages unaware of this extension. Figure 1 shows an example of an extended `GET` call.

```
http://www.aifb.uni-karlsruhe.de/english

GET /english HTTP/1.1
Host: www.aifb.uni-karlsruhe.de
Accept: text/xml,application/xhtml+xml,text/html
XFOAF: http://nodix.de/foaf.rdf
```

Fig. 1. An example of an extended `GET` request.

On the browser side, we used the HTTP extension with the Mozilla based Firefox browser. We used the Firefox plugin *ModifyHeaders*[8] to add an `XFOAF` header and an appropriate value. We tested the extended `GET` with several websites, and as expected we did not encounter any problems. Although *ModifyHeaders* already provides us with all required functionality (deactivating the extension, changing the `XFOAF` value on the fly, etc.), the user interface is too technical by far for the casual user.

On the server side, we enhanced the AIFB Semantic Portal, the official website of the AIFB research institute, to make use of the extended HTTP requests. The portal is a ZOPE application[9] and offers several semantic features, like a full export of its data in OWL, using several vocabularies like SWRC [14], FOAF [3], and vCard [5, 8]. The portal also provides a SPARQL endpoint to query the data, and an RSS feed to syndicate news.

Now that the AIFB web server is able to access the user's FOAF file, it checks for persons the user says to know. On the main page of the portal the user is provided with direct links to these persons. From there, she can explore

[6] `http://www.mozilla.org/firefox`
[7] `http://www.aifb.uni-karlsruhe.de`
[8] `http://modifyheaders.mozdev.org/index.html`
[9] `http://www.zope.org`

Fig. 2. The personalized AIFB portal. The visitor is greeted with his name (top), and may find shortcuts to all known persons working at the AIFB (lower right quarter).

the person's publication list, current contact data, given courses, projects the person works on, and so on. Each user who offers the portal appropriate data is rewarded with personalized access and will be able to access the information she looks for much faster than before. Screenshot 2 shows the AIFB portal being personalized for one of the authors.

As the extended `GET` request only provides us with the URL of the FOAF file, we first need to extract the URI of the visitor. We expect the visitor being connected with a *foaf:primaryTopic*-relation to the FOAF URL. Now we can search for information about the visitor, like her name or the persons she knows. The latter needs to be compared to the local knowledge base. Either some persons have the same URI in both the visitor's FOAF file and the server's knowledge base, or based on a inverse functional property (like *foaf:mbox_sha1sum*) we are able to infer the equality of two persons in the two ontologies. We neither force the visitor to use the same URIs as we do, nor to use any URIs at all (and often they do not, because it was considered impolite for some time to assign URIs to persons, and thus inverse functional properties were used for identification). The described extension was implemented in Python[10], using rdflib[11].

[10] `http://www.python.org`
[11] `http://rdflib.net`

5 Use cases

Access to a semantic profile of the person browsing could enable the personalization of the browsing experience in several interesting ways. A number of potential usage scenarios are listed below:

- At a basic level, the web server can use the profile information of the browsing person to prefill forms on the website. Many web forms ask for information such as name, affiliation, homepage, all of which have dedicated tags in the FOAF file. Instead of the user having to manually enter all this information every time for multiple websites, the visited websites could automatically retrieve this information from the user's FOAF file and prefill forms of the website. The user can then review the prefilled fields, modify them if required or add information for 'new' fields.
- A FOAF file also contains links to known persons. By logging visitors and comparing them to the user's acquaintances, she could be notified if someone she knows visited the site recently. As discussed in [16], knowing who I know has visited this site contributes to companionability (doing things with friends), sociability (doing things with people who are similar to me), for establishing authority, and possibly authenticity. Thus, knowing that friends have visited this site increases the value of the site to me, and generally leads to increased site traffic.
- Since my acquaintances are likely to have interests similar to mine, knowledge of which pages they have viewed on a web site can be used to suggest relevant pages to me. At the very least, I know about the goals and interests of my acquaintances and can use this information to better understand the content on the website. This could be especially useful for large web sites with only few pages of real value to me, or if I lack expertise in a particular area and need guidance.
- If the FOAF profile of the browsing user is extended with viewer preference parameters, such as color schemes, preferred languages or accessibility preferences, such as large font size for visibility-impaired users, the web server can automatically customize the served HTML pages for the needs of the user. Such information need not be contained within the FOAF file itself, but it can point to other standardized specifications of user preferences, such as CC/PP [13], and the web server can gather the information from there.

The above usage scenarios can be realized within the current web itself. Given a Semantic Web with richer kinds of information available, additional scenarios become feasible. Thus, knowledge of the tasks and activities of the user as well as knowledge of organizational hierarchies that the user is part of could help the web server to tailor the content it presents to the user.

With more heavyweight infrastructure on the server-side, the web site could support collaborative filtering over communities and topics. One interesting application of our personalization mechanism is the possibility of location-based messages or reminders for friends. For example, given an intranet or project website or just any website that is visited often by a group of friends, I might want to leave a message to notify me when someone I know has viewed a page or to leave a personalized message for them to be delivered when they access a particular page. This is essentially the virtual equivalent of leaving a post-it note on the community fridge.

Many of these scenarios have been explored in various prototype systems, particularly in the fields of human-computer interaction and computer-supported cooperative work. However, these systems have always been standalone and require non-trivial effort in set-up, meaning that they remain primarily interesting research prototypes. The main contribution of our work is that we propose a fairly simple personalization mechanism that relies on open standards and the Semantic Web, is straightforward to implement and therefore has the potential to be used on a large-scale in the real world outside laboratory settings.

6 Discussion

Users are basically giving up part of their privacy for the features suggested in this paper. However, they are now able to control what information they want to expose in a very fine granularity. The user is able to define several FOAF files easily and to switch between the one used by the browser on the fly, or to deactivate them completely. This way they could use another persona in their leisure time than they do at work. They could also decide to define rules that determine which persona to choose, if any, according to the website to be retrieved.

Besides having different personas, a user may also decide to grant different information about themselves to different websites. A discussion forum about the Lord of the Rings may get a list of all the fantasy articles the user has published, but not her academic merits, whereas a job search portal would be granted access to the latter knowledge but not to the first. To realize this, we cannot rely on simple static FOAF files. Instead we would need to enable each user to set up a web service that would decide, according to the user's settings and the websites description, which information to deliver. For example, if the websites specify their privacy practices using standards such as P3P [4], the user can use this information to decide what information to expose to the website. This goes beyond the proposal given in this paper, but we can easily extend it by simply adding the URI of the web service to the FOAF file. No further extension of the HTTP protocol would be required.

As FOAF files are public files on the web, every user could easily "hijack" the FOAF file of another person. In order to counter that problem, the web server may decide to encrypt each delivered page with the PGP public key of the user which would be included or referenced in the FOAF file. Only a browser that has access to the private key of the user would be able to decrypt the site. This also could replace sign-in systems, and free the user from remembering numerous different passwords, or using the same weak password and login combination in a plethora of different places. In fact, since users often tend to use the same or similar login/password combinations for multiple websites, relying solely on this authentication method is dangerous. A website provider could potentially use the login/password combination of the website's users to access a large number of other password-protected websites. For example, a merger between services like *flickr* and *Yahoo!* could actually provide the new service provider with a list of login/password combinations that were used in both websites, and thus are potentially used in many other websites as well, like *MSN*. Such knowledge could be harmful for both the user as well as the vulnerable web sites. A trusted infrastructure should therefore be in the interest of both the service providers and the users. By using a public key infrastructure as described above, the overall security of the web for the users would increase. This feature could be used independently from the personalization possibilities described in this paper.

In accordance to the spirit of the web, FOAF files are decentralized entities of varying quality. Some users may decide to offer their name, their interests, knowledge about their social networks and even such things as e-mail or real world addresses. Other users may opt not to to provide any of these, or simply lie about them. But this problem is already manifest on the web. If websites like news providers or public discussion forums ask for such information, they have no possibility to check the correctness of this information besides simple validity checks. Actually, the FOAF approach provides the user with the possibility to lie more consistently (and relieves them from entering the information again and again in different places, possibly forgetting or mixing up details), and thus to build much more trustable personas. This is usually enough for most websites – they often do not need to validate the correctness of the information, but rather they just require a basic means to identify their users and build models of them. That is the reason why cookies are in such wide usage, but cookies have several further flaws: they may be tampered with or 'stolen' by an unauthorized third-party [10]. Also cookies are not easily transferable from one computer to the other, whereas a browser on another computer would only require the URL of the FOAF file in order to personalize all the pages. Both parties, website providers and their users gain from this situation.

It should be noted that simply by using their publicly-accessible FOAF profiles, users do not expose any information to web sites that is not already available to the websites today. A website like *CNN*[12] already asks for the user's name and email for certain services. Equipped with this information they could use a search engine to locate the user's FOAF file and collect the information within. The approach described in this paper makes this connection more explicit, reliable, and gives more control to the user.

7 Conclusions

In this paper, we described a simple, backward-compatible extension to the HTTP protocol that enables a web server to use a visitor's FOAF information for considerable personalization of her web browsing experience. We demonstrated

[12] http://www.cnn.com

the feasibility of the proposed extension by implementing it on a proof-of-concept level for the Firefox browser and the AIFB portal. The information in the FOAF file becomes really exciting when websites can tell you whether your friends have visited the same site or the pages they viewed. We would like to explore this within the context of our own research institute portal and examine the usage of our personalization mechanism in practice. A user study (or extensive adoption of the proposed approach) will be required to understand how much information users are actually willing to provide in such a setting, and how useful this information really is in enhancing their web experience.

Acknowledgements

This work was supported by the European Commission in the IST programme under the SEKT project, Semantically Enabled Knowledge Technologies (IST-1-506826-IP) and by the German Federal Ministry of Education and Research (BMBF) under the SmartWeb project (grant 01 IMD01 A).

References

1. D. Beckett. RDF/XML syntax specification (revised), February 2004.
2. T. Berners-Lee, J. Hendler, and O. Lassila. The Semantic Web. *Scientific American*, (5), 2001.
3. D. Brickley and L. Miller. The friend of a friend (FOAF) vocabulary specification, June 2005.
4. L. Cranor, M. Langheinrich, M. Marchiori, M. Presler-Marshall, and J. Reagle. The platform for privacy preferences 1.0 (p3p1.0) specification, 2002.
5. F. Dawson and T. Howes. vcard MIME directory profile: Internet RFC 2426, 1998.
6. R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee. Hypertext transfer protocol – HTTP/1.1, 1999.
7. J. Hartmann and Y. Sure. An infrastructure for scalable, reliable semantic portals. *IEEE Intelligent Systems*, 19(3):58–65, May 2004.
8. R. Iannella. Representing vcard objects in RDF/XML, 2001.
9. M. Koch and K. Mslein. Identities management for e-commerce and collaboration applications. *Int. Journal of Electronic Commerce (IJEC)*, 9(3):11–29, 2005.
10. D. M. Kristol. Http cookies: Standards, privacy, and politics. *ACM Transactions on Internet Technologies*, 1(2):151–198, 2001.
11. D. Mulligan and A. Schwartz. Your place or mine? privacy concerns and solutions for server and client-side storage of personal information. In *Proceedings of Computers, Freedom and Privacy (CFP)*, Toronto, ON, Canada, April 2000.
12. `https://accountservices.passport.net`.
13. F. Reynolds, J. Hjelm, S. Dawkins, and S. Singhal. Composite capability/preference profiles (CC/PP): A user-side framework for content negotiation, July 1999.
14. Y. Sure, S. Bloehdorn, P. Haase, J. Hartmann, and D. Oberle. The SWRC ontology. In *Proc. of the 12th Portuguese Conf. on AI (EPIA 2005)*, volume 3803 of *LNCS*, pages 218–231, Covilha, Portugal, 2005. Springer.
15. M. Svensson. *Defining, Designing and Evaluating Social Navigation*. PhD thesis, Stockholm University, Department of Computer and Systems Sciences, 2002.
16. A. Wexelblat and P. Maes. Footprints: History-rich tools for information foraging. In *Proceedings of CHI 1999*, pages 270–277, 1999.

Semantic Web as enabling technology for m-commerce personalisation: Scenarios

Lillian Hella, John Krogstie

Norwegian University of Science and Technology (NTNU)
Sem Saelandsvei 7-9, NO-7491 Trondheim, Norway
{hella, krogstie}@idi.ntnu.no

Abstract. Trondheim will soon be a wireless city. This gives us an opportunity to test and evaluate personalisation in a large scale ubiquitous environment. To be able to decide on the contents of a personal profile we use scenarios as a starting point. All scenarios are made to fit into the local environment in Trondheim. The scenarios presented give an indication for what such a profile should consist of and the surrounding context. Semantic Web technologies will be investigated as enabling technology for using profile information to fulfill the scenarios.

1 Introduction

New types of networks and devices bring the whole Internet into everyday lives of users through wireless and mobile technologies. Mobile computing is increasing our capability to physically move computing services with us [1]. The concept of pervasive computing implies the capability to obtain the information from the environment in which it is embedded and utilise it to dynamically build models of computing [1]. Weiser defined a new type of computing which he called ubiquitous computing [2]: "the most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it." The main challenges originate from integrating mobility with pervasive computing. Any computing device, while moving with us, should build dynamic models of its various environments and configure its services accordingly [1].

The small size of mobile terminals make them an ideal channel for offering personalised and localised services [3]. It is important to enable the technology, generate user experience and catch the attention of new customers. Customers want to gain something from using m-commerce, for example better customisation, convenience, saving of time and better price deals [4].

Chapter 2 describes scenarios for one extended persona. The profile approach and Semantic Web technology is introduced in Chapter 3, while infrastructure for development and evaluation is presented in Chapter 4. Conclusions and future work is presented in Chapter 5.

2 Scenarios

Personas include not only what persons do, but also how they behave (personality, skills, properties, experiences etc). Personas combined with scenarios make the starting point for our stories.

2.1 Persona: Jason

The scenarios will be used as a starting point for development of services and future testing. Here we will focus on a single young to middle aged person. We have named him Jason. Jason is a well educated man living together with a friend. He does not like shopping in particular. However, he goes shopping when he has to, and his favourite shopping centre is “Solsiden centre”, with all its exclusive shops and nearby restaurants and café/bars. Many of his friends and colleagues live near by, and he often runs into familiar people whom he likes to meet or take a coffee with. He uses his mobile phone to plan meetings and other activities, and he finds this very useful. Hence he never shuts down or leaves his mobile phone unless he has to. Once his mobile blacked out, and he lost everything. Jason works in a company located in the centre of Trondheim, and his job involves meeting clients and colleagues, and also some travelling.

Jason is 35 years old, single without children, wealthy, educated, works for a consultancy company, wants special services and recognition, and owns a car and an apartment at Singsaker where he has one room for rent.

Attributes describe him as curious, demanding, impatient, impulsive, multi-tasker but easily distracted. He is experienced with computers and Internet, loves techno-gadgetry and is open for new impulses. Also, he cares about price/value relationship and may discontinue accounts if dissatisfied with services.

2.2 Scenarios separated in scenes

Here we present the stories with possible personalisations for Jason during two weeks.

Scene 1: Personal follow-up Monday Jason is out shopping in Trondheim. His profile status is set to available. When he enters Solsiden shopping centre he gets a sms on his mobile device from one of his favourite shops at the centre as he is identified through his Bluetooth mobile. They are giving him a special offer since he is a good customer. If he buys some jeans today, he will get bonus points that he can use as payment in all the stores in the centre. Jason is a registered customer at the shop, and this simplifies the purchase process. He tries out some new jeans that are right for him that the saleswoman has found in his size. He decides to buy one. As the offer is taken advantage of the purchase is registered at the shop. The merchandise he bought in the store is registered by scanning his device and stored in his profile.

Scene 2: Personal follow-up The following Friday Jason gets a new message about accessories and sweaters that will match with his previously bought jeans. As the lines are more frequently changed now, he is happy for getting the notification.

They ask if he wants a specimen waiting for him so he can try it out later that weekend.

Scene 3: GPS and collaborative community The second week he attends a project meeting with a project group that consists of five people. He wants to take the client out to lunch in the city centre. Jason requests for a list of five available restaurants. His GPS location is registered, and the ranked result list is influenced by a restaurant community Jason actively uses. Other factors that influence the list are the weather, number of people, preferences, previous experience and availability. Jason picks one, and is asked whether he wants to have a look at the menu, but declines. When the lunch is over, the accumulated bill for all people is deducted from Jason's mobile company account by scanning his mobile.

Scene 4: Entertainment and affirmative trading Jason is very interested in football. His favourite team is RBK. Therefore he has let a RBK service provider access parts of his profile. It finds which players he is most interested in, and how often he wants information. Primarily he gets information regularly on results and news. In the profile he decides that he also wants to be able to buy tickets and register for arrangements and trips. When he accesses the profile he usually uses his laptop. The second week RBK is playing a champions league match. He gets a message offering him tickets. As the match fits his schedule, he accepts. Payment information is exchanged and used to pay for tickets without visiting the sales office.

Scene 5: Service composition Jason has subscribed a birthday reminder service. There he has asked for a notification for his best friend's birthday. He receives a message some days before the big day and is asked if his personal agent can arrange something. The profile already knows his preferences and what gifts he has given to different people before, and he has access to parts of his friend's preferences, such as her age, interests and detailed wish list. Jason confirms that he needs suggestions, and his mobile returns an overview of things and activities he can do to please her. She is very fond of flowers, specifically roses. The profile has exchanged information about her membership at a beauty salon. They have registered that it's her birthday, and therefore give him an opportunity to buy a gift card with a special discount since she is already registered as a good customer. He thinks that the flowers and the beauty salon gift card is a good idea, but declines making reservations at one of her favourite restaurants. Instead, he finds a recipe for a three course dinner along with a list of ingredients he has to buy when preparing it for two persons.

3 Profiles and Semantic Web

We have divided a profile into three parts. On the lowest layer we find all the personal information. The information is common for every profile, and is what characterises each person. Name, address, telephone number and bank account are typical examples of what the personal information layer consists of, and as we could see this helped Jason e.g. in scene 1. The second layer consists of stable interests. That is, information about interests that don't change frequently. These are individual and consist of fields of interests one wants to get information and special offers on. Examples of this category can be film, music, sports and computer science. In scene 4 we saw how this was useful for Jason in relation to his favourite football club. On the top we find temporary interests. Examples of this category can be loan funding, real estate brokers and freezer, where one for the time being is interested in buying an apartment, or an instant need like restaurant reservations. Scene 3 illustrated how this was helpful for Jason when reserving a table at a restaurant.

The profile should be controlled by the profile owner, and he should be aware of what the profile consists of. We will investigate and demonstrate the utility of the Semantic Web approach using ontologies as a viable approach to support the personalisation services. An important goal with personalisation is not only to deny the wrong actors access, but also the get the right people access.

The Web is evolving towards a collection of services that can interoperate. It is becoming a provider of services rather than raw information. Seamless interoperability requires programs to be able to describe their own capabilities and understand other services' capabilities [5]. Ontologies are supposed to offer the semantic background for electronic information management and exchange. In m-commerce, there is a need for dealing with large amounts of information that can be distributed and heterogeneous. In electronic business like m-commerce, ontologies can provide content management, supply chain management, and value chain integration [6].

Ontologies, semantic web technology and profiles can be adopted to provide extensibility, flexibility, interoperability, and reusability. All the information, which is known about the user and his environment in a given point of time, make up the context information. We will separate the profile from the context, where the profile consists of information about this entity itself, but the where the combination of constitute the personalisation.

For interoperable systems to communicate there must be some agreement on coding as well as meaning of data. Agents have to share a model of what the data represent in order to obtain mutual understanding of the interchanged data [7]. Through reference to published ontologies, agents can understand and reason on the knowledge that provides user assistance [6]. The vision is to enable querying, where an agent gathers information presented at different resources on the Web and returns accurate information to the user.

4 Infrastructure for development and evaluation

The existing service platform (PATS) contains support of traditional telecommunications services (GSM, SMS, MMS, WAP) over Parlay gateways. Also other services such as location servers, streaming services and Map/POI-services are provided. A natural extension for us is an identity management/profiling component. Different development and execution environments is available on top of the service platform for the development of concrete services, for example the use of Service Frames for development of J2EE (Server) and J2ME (client) application. For service orchestration, one can use traditional modelling approaches and tools. We are also extending this with process modelling environments for supporting the modelling and enactment of so-called emergent workflow solutions [8].

For the evaluation of services, we are planning to use the Wireless Trondheim coordinated city WLAN-network which is currently being developed. The network supports high bandwidth to mobile units, and makes it possible to get location data on the user working in a nomadic modus to be fed into the PATS location server.

The long-term goal of Wireless Trondheim is to cover the whole of Trondheim with wireless broadband. The area of coverage for the first phase of development is limited to Midtbyen (City centre including e.g. the Solsiden shopping centre mentioned above) in Trondheim and its surroundings, along with NTNU grounds.

5 Conclusions and future work

Personalisation in ubiquitous environments will bring customers and service providers closer together. There is an increase in focus on each person as an individual. In a mobile environment personalisation will give service providers the opportunity to take advantage of this and increase their customer relationship. The profile can consists of layers of information; personal information, stabile interests and temporary interests. It will be important for the users themselves to control their profile, both content and which actors they have exchanged information with. We have described the approach using semantic web technology to increase users' abilities to easily express what information and services they need. The scenarios and personas will be further developed. This will be helpful to figure out what the profile should consist of. We also have to further investigate how we can separate the profile from the context, and how the profile and possible context can be used for personalisation. The main goal with the scenarios is not to think of killer apps, but rather to think of such personalisation issues.

References

1. Hansmann, U., Merk, L., Nicklous, M.S., Stober, T.: Pervasive Computing. Springer (2003)
2. Weiser, M., The Computer for the 21st Century, Scientific American (1991)
3. Tsalgatidou, A., Veijalainen, J.: Mobile Electronic Commerce: Emerging Issues. Lecture Notes In Computer Science; Vol. 1875, Proceedings of the First International Conference on Electronic Commerce and Web Technologies, Springer-Verlag (2000) 477 – 486
4. Plesman, M.: Serving the Mobile Customer How the Mobile Industry Is Preparing to Get Its Customers Online. Arthur D. Little, Telekommunikation, E-Business (2000)
5. McIlraith, S. A., Martin, D. L.: Bringing Semantics to Web Services, pp. 90-93, IEEE Intelligent Systems (2003)
6. State of the art and state of the practice including initial possible research orientations. Deliverable D8.1, INTEROP, Network of Excellence
7. Sheng, Q. Z., Benatallah, B., Maamar, Z., Dumas, M., Ngu, A. H. H.: Enabling Personalized Composition and Adaptive Provisioning of Web Services, Proceedings of 16th CAiSE, Volume 3084/2004, (2004) 322-337
8. Payne, T. R., Miller, L.: Calendars, Schedules and the Semantic Web. ERCIM News, No. 51 (2002) 16
9. Krogstie, J., Jørgensen, H.: Interactive Models for Supporting Networked Organisations. In 16th CAiSE, Riga, Latvia, Springer Verlag (2004)

Ontology-based Personalization for Multimedia Content[1]

Ph. Mylonas[1], D. Vallet[2], M. Fernández[2], P. Castells[2] and Y. Avrithis[1]

[1] National Technical University of Athens
School of Electrical and Computer Engineering
9, Iroon Polytechniou Str., 157 73 Zographou, Athens, Greece
{fmylonas, iavr}@image.ntua.gr

[2] Universidad Autónoma de Madrid, Escuela Politécnica Superior
Campus de Cantoblanco, 28049 Madrid, Spain
{david.vallet, miriam.fernandez, pablo.castells}@uam.es

Abstract. Personalization is a difficult problem related to fields and applications ranging from information retrieval to multimedia content manipulation. Challenge is greater, when trying to combine traditional personalization techniques with novel knowledge representations like ontologies. This paper proposes a novel contextual knowledge modeling, based on ontologies and fuzzy relations and exploits it in user profiling representation, extraction and use. The personalized results of the application of this methodology are then ranked accordingly. The performance of the proposed techniques is demonstrated through preliminary experimental results derived from a real-life data set.

1. Introduction

Personalization forms an interesting asset used in the field of information retrieval (IR), suffering though from ***information overload***, since IR usually tends to select documents, many of which are barely related to the user's wish [3]. Personalization uses information stored in user profiles, additionally to the user's current search or query, to estimate the users' wishes and select the set of relevant documents. In general no common distinction exists between different profiling algorithms. Handling of personalized information may be decomposed into three tasks tackled within this work: i) design of appropriate knowledge representation, ii) design, development and application of profiling algorithm and iii) presentation and ranking of results.

Successful extraction of user profiles, using ontological knowledge [5] is still considered an open issue, because it is difficult to apply in multimedia environments. In order to interpret user queries, we consider contextual information available from prior sets of user actions. We refer to this information as contextual knowledge or just context. This work deals with exploiting ontology-based contextual information, specifically aimed towards its use in personalization tasks. The structure of the paper is as follows: in section 2, we present our knowledge infrastructure, introducing the notion of fuzzy relations in ontologies. In section 3 we explain our user profiling algorithm

[1] This research was supported by the European Commission under FP6-001765 aceMedia.

and we extract user preferences based on usage history, fuzzy hierarchical clustering and ontological knowledge. In section 4, we rank the retrieved results, while in section 5 we provide early experimental results and in section 6 we present our conclusions.

2. Ontology-based Knowledge Representation

It is very difficult to create generic personalization solutions, without having a large knowledge at hand. Enriching this knowledge with contextual information results in a useful and representative set of user preferences. We define this set as the contextualized set of user preferences. We restrict the notion of context in this work to the notion of ***ontological taxonomic context***, defined on top of a "fuzzified" version of traditional ontologies. This context implements the necessary knowledge model and is strongly related to the notion of ontologies: an ontology can be seen as an attempt for modeling real-world (i.e. fuzzy) concepts and context determines the intended meaning of each concept, i.e. a concept used in different context may have different meanings. In general, ontologies may be described as follows:

$$O = \left\{C, \left\{R_{c_i,c_j}\right\}\right\}, i, j = 1..n,\ i \neq j\ ,\ R_{c_i,c_j} : C \times C \rightarrow \{0,1\}, i = 1\ldots n \qquad (\mathbf{1})$$

, where O is an ontology, C the set of concepts it describes and R_{c_i,c_j} the semantic relation amongst two concepts $c_i, c_j \in C$.

We define ontological context in the means of fuzzy taxonomic ontological relations. Although ontologies may contain any type of relations, only taxonomic relations are of our interest, since the use of such relations is necessary for the determination of the document's context [1]. Additionally, accurate representation of real-life information governed by uncertainty is only possible using fuzzy relations [6]. Consequently, we introduce a "fuzzified" definition of an ontology:

$$F(O) = \left\{C, \left\{r_{c_i,c_j}\right\}\right\},\ i, j = 1..n,\ i \neq j\ ,\ F(R_{c_i,c_j}) = r_{c_i,c_j} : C \times C \rightarrow [0,1] \qquad (\mathbf{2})$$

, where $F(O)$ forms a "fuzzified" ontology, C is the set of all possible concepts it describes and $F\left(R_{c_i,c_j}\right) = r_{c_i,c_j}$ denotes a fuzzy relation amongst two concepts.

Unfortunately, current ontology languages (OWL, DL and plain RDF) are not powerful enough to model such an ontology. Thus, we decided to enhance RDF, being a standardized, graph-modeled language, with novel characteristics like ***reification*** [7]. The proposed model is a graph, in which every node represents a concept and each edge between two nodes forms a contextual relation between the concepts. Additionally, each edge has an associated degree of confidence, implementing fuzziness. Describing the additional degree of confidence is carried out using "manual" reification, i.e. making a statement about the statement, which contains the degree information. In the next example concept ***holiday*** is related to concept ***sky*** with a fuzzy relation ***isRelatedTo*** and a degree of confidence equal to 0.75. Supposing an RDF namespace dom, we have:

```
<rdf:Description rdf:about="#s1">
		<rdf:subject rdf:resource="&dom;holiday"/>
		<rdf:predicate rdf:resource="&dom;isRelatedTo"/>
		<rdf:object>rdf:resource="&dom;sky"</rdf:object>
		<rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Statement"/>
		<context:isRelatedTo rdf:datatype="http://www.w3.org/2001/XMLSchema#float">
		 0.75</context:isRelatedTo>
</rdf:Description>
```

Following the above principles our knowledge model is able to utilize any type of real-life fuzzy relations between concepts. For personalization purposes, we utilize two of them, the ***specialization*** relation, ***Sp***, and the ***part*** relation, ***P***. Relation ***Sp*** is a fuzzy taxonomic relation on the set of concepts and $Sp(x,y)>0$ means that the meaning of ***x*** "includes" the meaning of ***y***. Relation ***P*** is also a fuzzy taxonomic relation on the set of concepts and $P(x,y)>0$ means that ***y*** is a part of ***x***. Combining the above relations, we construct a fuzzy taxonomic relation $T = Tr(Sp \cup P^{-1})$, which is suitable for the handling of user preferences. ***T*** implies that if the user query contains ***x***, then $T(x,y)$ indicates that documents that contain ***y*** will also be of interest. The transitive closure Tr is necessary, since the union of transitive relations is not necessarily transitive [6].

3. User Profiling

We illustrated the modeling of contextual dependence between concepts and relations using an RDF-based representation and a fuzzy taxonomic relation ***T***. We continue with the presentation, extraction and use of user preferences. In compliance with the fuzzy notation presented in [6], we adopt the following formal representation of user preferences ***P*** :

$$P = \{U^+, U^-\} \quad (3)$$

where U^+, U^- refer to the set of positive and negative preferences, respectively. Following the sum notation for fuzzy sets [6] U^+ and U^- are defined as follows:

$$U^+ = \{U_j^+\},\ j \in N_k,\ U^- = \sum c_i / p_i^-,\ i \in N_n,\ n = |C| \quad (4)$$

k is the count of distinct positive preferences contained in the user profile, p_{ij}^+ is the degree of participation of concept c_i in U_j^+, p_i^- is the degree of participation of concept c_i in U^- and $U_j^+ = \sum c_i / p_{ij}^+$, $i \in N_n$, $j \in N_k$, $n = |C|$.

This definition allows participation of a single concept in multiple preferences and to different degrees. As all relations existing in the ontology are defined on the set C of concepts, we define user preferences on the same set, i.e. user preferences are also concepts: $P \subseteq C$.

If the type of user action included in the user's usage history demands it (like a ***search*** action), the set of documents presented to the user prior or after to that action is also preserved. These constitute the history documents associated to the specific user profile and user preferences are derived directly from them. Each history document ***d*** is represented as a fuzzy set on the set of concepts that are related to it: ${}^{0+}d = \{c_1, .., c_n\} \subseteq C$ and preferences are mined by applying clustering algorithms on it. Most clustering methods belong to either partitioning or hierarchical, however the former require the number of clusters as input and thus are inapplicable [6]. The proposed approach may be decomposed into the following steps:

- Perform a fuzzy clustering of concepts in order to determine the count of distinct preferences that a history document is related to, according to the following steps:
    1. Turn each available concept into a singleton, i.e. into a cluster k of its own.
    2. For each pair of clusters k_1, k_2 calculate their distance $d(k_1,k_2)$.
    3. Merge the pair of clusters that have the smallest distance $d(k_1,k_2)$.
    4. Continue at step 2, unless termination criteria are met; termination criterion most commonly used is a threshold for the value of $d(k_1,k_2)$.
- Find the user preferences that are related to each cluster.
- Aggregate the findings for each cluster to acquire an overall result for each d .

The key element of the above algorithm is the ability to define a unique distance among any pair of clusters, given the input space and the clustering features. We propose the following distance estimation:

$$d(k_1,k_2) = \sum_{i \in ¥_F} \sqrt[k]{\frac{\sum_{x \in k_1, y \in k_2} r_i(x_i, y_i)^m}{|k_1||k_2|}} \qquad (\mathbf{5})$$

, where $r_i, i \in ¥_F$ is the metric that compares the i-th feature, F the overall count of features, $|k_1|$ the cardinality of cluster k_1 and μ a constant. Obviously, μ=1 approaches the mean value and μ=2 yields the Euclidean distance.

Still, this clustering method creates only crisp clusters and does not allow for overlapping among the detected clusters. In real life, a concept is related to a preference with a degree in $[0,1]$ and is also related to more than one distinct preference, making "fuzzification" of the partitioning necessary. We construct a fuzzy classifier, in the means of a function $C_k : C \to [0,1]$ that measures the degree of correlation of a concept c with cluster k. Then, we expand the detected crisp partitions to include more concepts. Partition k is replaced by cluster k^{fuzzy}, following again the sum notation for fuzzy clusters:

$$k^{fuzzy} = \sum_{c \in {}^{0+}d} c / C_k(c) \qquad (\mathbf{6})$$

Obviously $k^{fuzzy} \supseteq k$. The set of preferences that correspond to a history document is the set of preferences that belong to any of the detected fuzzy clusters of concepts.

4. Personalized Retrieval

Once user profiles are obtained by extracting user preferences from the semantically analyzed usage history, our approach to preference-based content retrieval [2] is based on the definition of a matching algorithm that provides a personal relevance measure prm(x,u) of a document x for a user u. This measure is set according to the semantic preferences of the user, and the semantic annotations of the document. In our model, the semantics of documents in the retrieval space are assumed to be described by a set of weighted domain concepts, attached to the documents as annotations.

The procedure for matching a content object to the user preferences is based on a cosine function for vector similarity computation. For this purpose, we build a vector-based representation of user preferences from the fuzzy sets defined in the previous section. The user preference vector ***p*** is defined by $p_i = \sum_j p_{ij}^{+} - p_i^{-}$, for each concept c_i. Then the expected degree of preference of user ***u*** for a document ***x*** is computed by:

$$\mathrm{prm}(x,u) = \cos(x,u) = \frac{x \cdot u}{|x||u|} \qquad (7)$$

, where ***x*** stands for the vector of annotations of the document, so that x_i is the weight of the annotation of the document by each concept c_i in the user profile.

The measure above can be used as is to rank documents, based only on user preferences, as well as to personalize an explicit user query ***q***, when combined with a query-based score without personalization sim(*x*,*q*), to produce a combined ranking [4]. In our approach, we adopted the ***combSUM*** model, by which the two rankings are merged by a linear combination of the relevance scores:

$$\text{score}(x,q,u) = \lambda \cdot \text{prm}(x,u) + (1 - \lambda)\,\text{sim}(x,q), \text{ where } \lambda \in [0,1] \qquad (\mathbf{8})$$

The choice of the λ coefficient in (8) provides a way to gauge the degree of personalization, ranging from λ=0 producing no personalization at all, to λ=1, where the query is ignored and results are ranked only on the basis of global user interests.

5. Experimental Results

In order to test the proposed techniques, we have conducted early experiments, which we describe next. The purpose of the experiments is to test the consistency of the preference learning by using them to personalize the output of a visual search engine on a corpus of images. The test measures the overall effectiveness of the preference learning approach described in section 3, followed by the personalized ranking step described in section 4. The dataset set up for the experiments included:

- A sample “fuzzified” ontology, in RDF format, containing more than 1000 concepts. Relationships between concepts were defined by relation ***T***.
- A set of 150 documents for usage tracking and preference learning, consisting of images with manual free-text annotations. A simple semantics extraction method was used to produce ontology-based metadata vectors from the textual annotations.
- A second set of 100 images for querying and retrieval with similar characteristics, but separated from the first one, in order to show non-trivial results, i.e. the system being able to predict user preferences for images that were not available at the time the user’s interest for specific documents was monitored.

Based on this corpus, the experiment consisted of the following steps:

1. A subject selected 9 images from the first set of images displaying works of art, which had annotations by concepts such as ***chapel***, ***fresco***, ***tower***, ***fabric***, ***Padua*** and others. The concept vectors attached to the selected images are automatically stored by the system as history documents.
2. The preferences extraction algorithm is applied and for the sake of simplicity the fuzzy hierarchical clustering method identifies only positive user preferences, i.e. U^{+}, yielding: $U_1^{+} = \mathbf{health}/0.91 + \mathbf{leaders}/0.88 + \mathbf{art}/0.90$.
3. The subject is asked to provide preference-biased ground truth data for a “search for similar” query on the second document collection, the query consisting of a

photo showing a horse. The user classifies each picture in the collection as relevant or non-relevant for the query, according to his own biased judgement.

4. The personalized search algorithm is run on the same query and collection, using an image-based search engine, the output of which is re-ranked by preference as described in section 4.

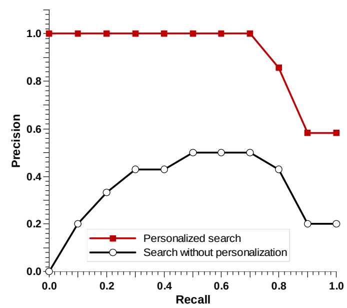

Figure 1. Recall/precision curve of the search with and without personalization

Figure 1 shows the performance of the ranked search results returned in step 4, compared to the results obtained without personalization. The poor precision of the search without personalization at the lowest recall levels is due to the fact that the image-based retrieval algorithm returns initially irrelevant results. Overall experiments show that the proposed ontology-based personalization is particularly helpful in difficult multimedia retrieval tasks.

6. Conclusions

We have implemented and tested a personalized retrieval and ranking framework, that can be exploited towards the development of more efficient personalization environments. Its core contribution has been the provision of personalized access to multimedia content. We based our efforts on a novel "fuzzified" ontological knowledge model, utilizing contextual information and fuzzy taxonomic relations, towards representing, extracting and using of user preferences. Early results on personalized content retrieval are very promising and form an interesting perspective.

References

1. Akrivas G., Stamou G., Kollias S., "Semantic Association of Multimedia Document Descriptions through Fuzzy Relational Algebra and Fuzzy Reasoning", IEEE Trans. on Systems, Man, and Cybernetics, March 2004.
2. Castells P., Fernández M., Vallet D., Mylonas P. and Avrithis Y., "Self-Tuning Personalized Information Retrieval in an Ontology-Based Framework", LNCS vol. 3762, pp. 977-986, 2005.
3. Chen P.M., Kuo F.C., "An information retrieval system based on a user profile", Journal of Systems and Software 54, 3-8, 2000.
4. Dwork C., Kumar R., Naor M. and Sivakumar D., "Rank Aggregation Methods for the Web", Proc. of the 10th Intl. World Wide Web Conference (WWW10), Hong Kong, 2001.
5. Gruber T.R., "A Translation Approach to Portable Ontology Specification", Knowledge Acquisition 5: 199-220, 1993.
6. Miyamoto S., "Fuzzy Sets in Information Retrieval and Cluster Analysis", Kluwer Academic Publishers, Dordrecht / Boston / London 1990.
7. W3C Recommendation, RDF Reification, http://www.w3.org/TR/rdf-schema/#ch_reificationvocab.

Using ACUITy to Personalize Content in Semantic Web Applications*

Amy Aragones[1], Jeanette Bruno[1], Andrew Crapo[1], Marc Garbiras[1],

[1] General Electric Global Research, 1 Research Circle, Niskayuna, NY 12065 USA
aaragones@research.ge.com, bruno@research.ge.com,
crapo@research.ge.com, garbiras@research.ge.com

Abstract. Work domains such as maintenance and logistics planning are characterized by open-ended problem solving and large quantities of heterogeneous and distributed information. Problem-solvers in these domains can benefit from semantic web applications for work-centric decision support. In this paper we describe the need for such technology and the missing links in the current state of the art. We also present Adaptive Work-Centered User Interface Technology (ACUITy), still in its formative stage, as a way to meet this need.

1 Introduction

One can argue that the Semantic Web will deliver the biggest benefits in dynamic work domains that are subject to variability in decision-making behavior and information needs. One example of such a domain is logistics and maintenance planning for complex systems, which is characterized by large amounts of distributed and heterogeneous information that must be accessed and comprehended in a timely manner. It is also constrained by many business rules, geographically distributed collaboration and problem solving and other factors that influence decision processes. There is often a need to analyze what-if scenarios to evaluate potential courses of action and to prioritize work at the plan or project level as well as at the task level.

Increasingly, decision-makers in maintenance planning are faced with an automated, proactive information flow that challenges them to cognitively process, analyze and take action on a large amount of information. In order to prevent information overload, it is critical to find ways to provide users with the right information at the right time and in the right format. What constitutes the right information, the right format and the right time will almost always depend on the user's objective and will often depend upon the preferences of the user and his or her unique experiences.

In domains like this, semantic technology can provide a more truly collaborative relationship between humans and computer systems to help manage the type, amount and content of information accessed and applied to solve planning problems. Personalization is a cornerstone of this collaboration. The development of user models has

* This work was partially funded by the Air Force Research Lab under contract F33615-03-2-6300.

been a key focus for many working toward personalized or adaptive information delivery. Other work (e.g. [13], [10]) explores how to infer user models from web access data and how to adapt content display and navigation in hypermedia (e.g. [4], [7]). In addition to these research efforts, it is useful to take a work-centric approach to semantic modeling for useful personalization, beyond information modeling and user modeling based on personal characteristics, navigation and information retrieval. In particular, we are exploring three additional dimensions.

First, we agree with Schwarzkopf [11] that there is a great deal to learn about how users make sense of information in semantic applications beyond information retrieval. Others are working in the areas of semantically-enabled searching, information management and annotation [11]. We can extend these sense-making activities to how users visually reorganize, reformat, or change other properties of information. For example, users who manually fuse information using pencil and paper or using work aids like spreadsheets find implicit meaning in data by laying out chunks of content in different ways, changing the format of the information and highlighting information to find new relationships. We would like to capture those layout and formatting choices in a semantic model to help us understand how information becomes useful.

Second, one would expect that user preferences revealed through information selection and layout, and feedback from users about usefulness and meaning are largely second-order effects. We also need to represent the first order effects associated with the intrinsic nature of the work that needs to be done and the context in which users find themselves faced with the tasks they will attempt to execute. What is the state of the domain in which the users are working and what needs to be done to change that state? Such factors are critical in predicting the relevance of information and the effectiveness of a particular format. Thus, we require not just models of the users themselves and their perspective on the world, but also higher-level models of the work domain and the problems being solved.

Third, we also need to develop upper-level models of the interaction mechanisms necessary to achieve mixed interaction between human users and decision support systems. The Semantic Web represents an important step forward in the very difficult problem of representing information so that it can be interpreted by both people and artificial agents. This is a pre-requisite for any meaningful collaboration between people and machines. Another pre-requisite, we believe, is that our human-computer interfaces (HCI) be mixed initiative. It is not acceptable for the program to be in control with the human simply providing the inputs and receiving the "right" answer generated. It is also not adequate for the computing platform to simply provide low-level tools like word processors, spreadsheets or search engines to assist the human in performing cognitive tasks without any understanding of what the person is trying to accomplish. Collaboration requires a shared semantics all the way up to the goals and objectives of the work being performed and all the way down to the way information is passed between user and system.

Thus, models of the user, the work and the interaction mechanisms between the human and the system must be represented and related. This representation is consistent with previous work in adaptive systems and adaptive hypermedia [2], [3].

The Adaptive Work-Centered User Interface Technology (ACUITy) research program has provided us with an opportunity to investigate the modeling of users, work domains and HCI constructs to achieve a collaborative environment in which personalized content can evolve in a meaningful and responsive way. In this paper, we give an overview of our emerging modeling approach and the architecture we have implemented to tailor content according to the problem being solved and the preferences of the human problem solver. Our goal is to invite further discussion by documenting the current state of our technology and sharing ideas for future directions, mindful that there are still many open questions to be answered.

2 Adaptive Work-Centered Support

First introduced by Eggleston and Whitaker [5], the goal of a Work-Centered Support System (WCSS) is to "provide an integrated and tailored support system that…offers support to work in a flexible and adaptable manner" by customizing the user experience and interaction according to the situated context in which work is accomplished. A WCSS also emphasizes the importance of the user interface itself as a work aid, regardless of automated decision support applications that may also deliver helpful information or take action on behalf of the user. That is, having a workspace where information can be visually represented and manipulated can be of significant benefit to users, whether or not collaborative agents are helping out in the background.

We extend Eggleston and Whitaker's approach by taking advantage of semantic web technology to achieve personalized decision support. We have derived core upper level concepts from the three work-centered design principles introduced by Eggleston and Whitaker [6], including:

1. *The Problem-Vantage-Frame Principle:* Effective decision support interfaces should display information that represents the perspective that the user requires on the situated work domain to solve particular types of problems.
2. *The Focus-Periphery Organization Principle:* Information that is the most critical to the user in the current work context should be displayed in the focal area of a decision support display to engage the user's attention; referential information should be offered in the periphery of the display to preserve context and support work management.
3. *The First Person Perspective Principle:* The user's own work ontology (terms and meaning) should be used to characterize information elements in the interface display.

Our vision of an "Adaptive WCSS" is much like a supportive collaborator who watches the user and provides whatever is needed while warning the user of pitfalls to be avoided. To interact successfully, human collaborators must establish several conditions. First, clear communication is only possible if there is a common vocabulary (shared semantics). Second, given that shared language, collaborators must agree on a model of the problem to be solved. It is this model of the problem that allows relevant information to be identified and communicated, alternative solutions to be identified or synthesized and evaluated, and the success of the process to be evaluated. In the next section we present the upper-level concepts we use as a framework to facilitate this kind of collaboration.

3. The ACUITy Architecture

ACUITy has three major components: the ACUITy Problem-Vantage-Frame (APVF) ontology, the ACUITy Controller and the User Interface (UI) Engine. The high-level architecture is shown in Figure 1.

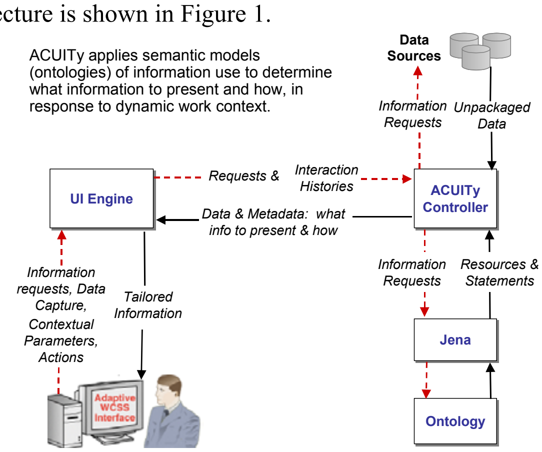

Figure 1. The ACUITy Architecture

3.1 An Introduction to the APVF Ontology

Work can be characterized as a series of unfolding problem-solving events [6]. Understanding the "intrinsic" nature of the work in terms of the type of problems being solved is critical to designing useful work aids. In the APVF ontology, we currently define a *problem* as something perceived not to be as desired; i.e., a state of affairs that requires the user to take action to change some aspect of the work domain. Although we regard the modeling of problems as a critical success factor, our modeling of problems is just in the beginning stages. There are many, many different ways that problems could be represented to varying degrees of detail. Thus, we decided to start from the other end of the issue and work backwards: to model how to communicate information between the application and the user and use that to discover what it is about the nature of problems that will be important to model. In other words, what should the system to know in order to effectively personalize information content and style?

Thus, at present, we have one-to-one mappings between problems and the sets of information needed to solve them, which we refer to as vantages. A *vantage* is a "window" into work domain information that provides the user with the particular perspective needed to solve a problem or problem set. More formally, a vantage is a collection of presentations of information that are relevant for a particular problem, along with their properties, which may include level of relevance, preferred position, size and other formatting considerations.

A WCSS frame "instantiates a vantage with specific display and control elements" [6]. We interpret this as the properties of a session with one or more vantages, each of which will have an associated problem set and a set of relevant information objects. In other words, a *frame* brings a set of vantages into relation via a set of common properties to mediate a particular work session. For each domain, the frame is specified as 'containing' vantage objects. The ACUITy Controller manages user sessions in terms of frames. With each user session, the user is given the option of creating a new frame instance or picking up their work from an existing session frame instance.

We enable a single frame to be populated by multiple vantages, one of which occupies the focus of the display while other secondary vantages are easily accessible in the periphery. We do this because a user may be working on a problem set that requires somewhat different and discretionary insight into the problem domain without losing the overall frame of reference. Figure 2 summarizes the relationships between the Problem-Vantage-Frame model constructs and work, user and interaction models.

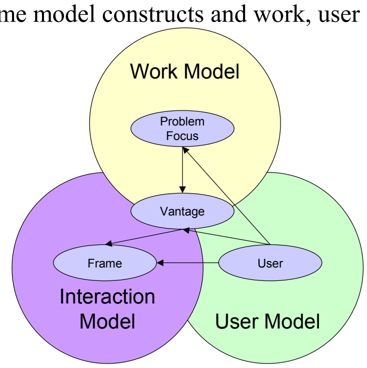

Figure 2. Problem-Vantage-Frame Concepts

A vantage is made up of a collection of presentation objects that are relevant for solving a particular problem set. A *presentation object* captures the what and how of displaying a certain set of information to the user. Presentation objects have the refining property "presentation nature", which specifies the general type of display. At present, ACUITy presentation natures include: 2d graphs, scrolling tables, text (including html and plain text documents as well as hyperlinks), display groups, external web applications, and user interaction objects such as forms, text entry fields, buttons, and various selection mechanisms (lists, check boxes, radio buttons, etc.). Although the type of the display object sets the base presentation style, the user may modify its look and feel unless precluded from doing so by ontology restrictions.

A *display object* is a type of presentation object that "encodes" work domain information for presentation to the user. A *script* captures the exact instructions for retrieving information, whether from the ontology or from an external data repository. Work domain information objects may also be refined by filters, which mask some of the information in the data source.

An *interaction object,* a type of display object, allows the user to take some action that will have consequence or side effect, either in the client user interface or on the server. In the Web interface, client-side effects are implemented as client-side JavaScript. Server-side effects may be specified using a variety of *script* types. Note that the principles of WCSS encourage a mixed-initiative interface in which the user may take many actions to access information useful to the work at hand. Thus a particular state of the user interface might have many interaction objects.

Data maps define how to transform data returned by a script or a work domain information object to the format required by a display object. A set of standard maps is provided for the most common transformations. A map can also specify use of a custom plug-in transformation.

A property of a presentation object that affects its look and feel is referred to as a presentation *parameter*. Unless restricted by the model, presentation parameters are accessible to the user at run time for customization.

The APVF ontology contains many other concepts and properties, which may be used and extended to meet the needs of domain-specific applications. It is represented in the Web Ontology Language (OWL) and is described in detail in [1].

ACUITy and domain-specific work models also utilize upper-level and publicly available ontologies. These ontologies define concepts of time, physical versus abstract, problems, scripts, processes, and remote data sources. Scripting capability includes support of custom Java code that can implement data access, data transformation, or side effects. This facilitates integration of ACUITy applications with existing information repositories and computational models.

3.2 The ACUITy Controller

The ACUITy Controller is a Java class (with supporting classes) that provides an API to the APVF ontology. It provides special-purpose reasoning over this knowledge base to determine the set of information relevant to the problem at hand or the context of work performed. The ACUITy controller queries the ontology to understand where to find data, how to obtain it, and how to bundle it. The controller also accepts inputs from the client UI Engine and updates the ontology accordingly.

The ACUITy Controller selectively persists instance data, enabling the learning described below. It also integrates procedural knowledge (scripts). Some examples of the type of procedural actions performed by the ACUITy Controller include:

- User actions: e.g. create a new session (“frame”).
- Auto-populating a new frame with vantages and vantages with presentation objects.
- Mapping functions for graph presentation. For example, transforming the results of a query into a data series (e.g. create a candle chart), or transforming raw data into presentation attributes (e.g. numeric values to graph labels).

One can extend the procedural actions provided by the ACUITy Controller by creating user-defined scripts as new instances of action classes in the ontology.

The ACUITy Controller is designed for mixed interaction; perhaps the most fundamental interaction with the ACUITy Controller is to ask questions and tell new information. For example, a missing property is the name we give to an inconsistency between the necessary conditions imposed by restrictions on an OWL class and an actual individual of that class. For example, our ontology might define the class Mother as a Woman with a someValuesFrom class Person restriction on the property hasChild. If it is known that Jane is a Mother but Jane has no hasChild property with value an instance of Person, we may conclude that Jane has a missing property.

The ACUITy controller identifies three basic types of actions that can be taken in the event of a missing property:

1. Automatically create a new individual of the class identified as the range of the missing property
2. Ask the user to identify an individual to be the object of a new statement with the individual missing the property as subject and the missing property as predicate.
3. Execute a script as specified by the application developer (i.e. do something else that may or may not resolve the missing property)

This gives the ACUITy Controller the ability to allow, without requiring, the user to provide missing information.

3.3 The UI Engine

The User Interface Engine accepts metadata from the ACUITy Controller and creates the application's user interface. It sits between the end client and the ACUITy Controller and is meant to be the single interface point between the two entities. As such it interacts with the controller to request information and instruct the controller to update information and/or process scripts in response to the user's actions.

The UI Engine is made up of controller interface components, client platform agnostic UI component models, and UI renderer components. At this point we have implemented a web client renderer to produce a well-formed HTML document from the UI Engine.

4. Personalizing Content in ACUITy

4.1 Users Finish the Design

In an ACUITy application the users themselves finish the design by deciding what information they require to solve a particular problem – defining the vantage they need on the problem domain – and changing the characteristics of the information display in order to interact with the data more effectively. This type of direct adaptation is regarded as a critical need in open-ended, information intensive work [4]. ACUITy captures in a centralized way the experience of users in open-ended problem-solving domains as they gather information from many disjoint sources not precisely identified at design time.

The user can reconfigure the display by adding and removing presentation objects. The approach can be extended to permit ad-hoc additions of information sources. Customization of information display includes, but is not limited to, the hiding, ordering, and sorting of data table columns, the selection of graph series types, e.g., line versus bar, color, and labels, and the type of enumerated selection lists, e.g., drop-down list versus checkbox versus tabs. The user can duplicate and then modify visualizations as desired. Information in disjoint tables and graphs can also be brought into relation by the creation of shared highlight regions, similar to data brushing in statistical graphics.

4.2 Learning Defaults and Patterns

Learning from accumulated instance data is an implicit benefit of the semantic modeling approach taken by ACUITy. As users customize the content and visual characteristics of the information that they view in particular problem-solving settings, these changes are stored in the ontology with their context. This past history creates the opportunity for a reasoner to infer what information content is most appropriate based on new information that was unavailable during the initial design of the web application. This special purpose reasoner then uses this instance data to personalize both default content and appearance for new sessions with similar contexts.

Learning can occur at different levels. A single user's preferences can be learned from that user's past behavior. When user models include grouping of users according to role or other shared attributes, learning can occur across peer groups. Sample size and predominance thresholds can be used to control the conditions under which learned values are used. This level of learned defaults will allow new users to benefit from the experience of more experienced users.

Learning a default from instance data is the recognition of a simple pattern. While not yet implemented in ACUITy, peer group learning will provide the basis of establishing "best practice" information displays. Recognizing beneficial patterns of usage across groups of users can lead to new classes of display objects explicitly available to developers and users, whereas learned defaults are only implicitly available. Abstraction of useful patterns might even extend across application domains.

5. Applications Created Using ACUITy

We have used ACUITy to prototype several web applications of interest to General Electric, Lockheed Martin and the US Air Force. From a developer's perspective, our preliminary experience is that ACUITy is a powerful and very flexible environment for exploiting semantic technology to create real-world decision support systems. The APVF ontology provides developers of new web applications a starting point from which they can create information-rich displays by relatively simple model extensions – essentially one models the UI rather than programs it. With respect to the user-interface, the developer is also "finishing the design." For example, a new data table can be added to a display through a few simple steps. The behavior and attributes necessary for the table to be constructed and displayed, as well as those that allow the user to customize the table display according to their preferences, are inherited.

We have also prepared a Hello World application that will accompany the release of ACUITy to open-source in 2006. The Professor/Student Course Management (PSCM) application illustrates the use of semantic technology to implement work-centered decision support and the benefits of ACUITy to both application users and developers. We will demonstrate this application at the European Semantic Web Conference 2006 Poster and Demo Session.

Our prototypes are currently in various stages of user testing and we expect that the results of those evaluations will guide future development on the reasoning algorithms by which user preferences are learned. These initial systems will also accumulate semantically-tagged instance data that we can use to better understand the relationship between problems and vantages and the ways in which decision-makers make sense of and use information.

6. Discussion and Future Directions

As mentioned earlier, our modeling of problems is only in the early stages. A significant challenge in this regard is one of observation: how can we detect the type and nature of the problem on which users are working and the domain context? In our current work domain of interest, maintenance and logistics planning, there are some very interesting possibilities. New complex systems, such as aircraft, locomotives, medical devices and power systems are being engineered with sensors that detect properties of a deployed fleet of equipment. Information from these sensors can be used in prognostics and health management, which can potentially tell us a great deal about the problems users face day-to-day and the state of the domain in which they are working. Ambiguity is unavoidable in inferring what the user is currently working on, but a well-designed mixed interaction approach should serve to close the gap (i.e. perhaps the system can ask the user to confirm what he or she is working on).

We would also like to leverage existing efforts to expand our user models (for example, UserML and GUMO [9], FOAF [8] and work in semantic portals) and our representation of reasoning and rules for adaptation.

One of our most significant accomplishments to date has been the ACUITy architecture, in which new semantic constructs and reasoning algorithms can be rapidly prototyped in the context of work-centric web applications. This architecture is not only agile with respect to the logic that can be applied, but also with respect to the application domain. We also believe that the upper level concepts we have formalized in the APVF ontology are valuable contributions that can be expanded upon in open source and linked to other efforts in user modeling, adaptation and domain modeling. One of our next steps is to develop an end user-friendly ontology editor for domain-specific extensions of our APVF ontology. We plan to model problems in more depth and implement more sophisticated reasoning and learning capabilities; we would also like to expand our ability to allow users to perform ad hoc information searches and queries and integrate with web services. Now that the ACUITy framework is in place, we believe that we are well positioned to make longer, quicker strides in those directions and demonstrate incremental value to end users and enterprises from semantically-enabled systems.

References

1. Aragones, Amy, Bruno, Jeanette, Crapo, Andrew and Garbiras, Marc. "An Ontology-Based Architecture for Adaptive, Work-Centered User Interface Technology," IP.com: http://ip.com/pubView/IPCOM000134526D.
2. Benyon, D.R.: Adaptive Systems; a solution to usability problems. User Modeling and User Adapted Interaction, (1993) http://www.dcs.napier.ac.uk/~dbenyon/umuai.pdf.
3. Brusilovski, Peter, Cooper, David W. Domain, Task and User Models for an Adaptive Hypermedia Performance Support System. Intelligent User Interfaces IUI'02, (2002).
4. Carmagnola, Francesca, Cena, Frederica, Gena, Cristina, Torre, Haria: A Multidimensional Approach for the Semantic Representation of Taxonomies and Rules in Adaptive Hypermedia Systems. PerSWeb05 (2005) http://www.win.tue.nl/persweb/index.html.
5. Eggleston, R.G., Young, M.J., and Whitaker, R.D.: Work-Centered Support System Technology: A New Interface Client Technology for the Battlespace Infosphere. Proceedings of NAECON 2000, Dayton, OH, 10-12 (2000) 499-506.
6. Eggleston, R., and Whitaker, R.: Work-Centered Support Systems Design: Using organizing frames to reduce work complexity, Proceedings of HFES 46th Annual Meeting, Human Factors and Ergonomics Society (2002) 265 - 269.
7. Fransincar, F., Houben, G.: Hypermedia Presentation Adaptation on the Semantic Web. Proceedings of the AH 2002, LNCS, Springer Verlag (2002) 85-94.
8. The Friend of a Friend (FOAF) project: http://www.foaf-project.org/.
9. Heckmann, Dominik, Schwartz, Tim, Brandherm, Boris, Kroner, Alexander: Decentralized User Modeling with UserML and GUMO. 10th International Conference on User Modeling (2005) www.l3s.de/~dolog/dasum/DASUM05.Heckmann.pdf.
10. Kay, Judy, Lum, Andrew: Ontology-based User Modeling for the Semantic Web. PerSWeb05 (2005) http://www.win.tue.nl/persweb/index.html.
11. Schwartzkopf, Eric: Personalized Interaction with Semantic Information Portals. http://km.aifb.uni-karlsruhe.de/ws/LLWA/abis/schwarzkopf.pdf.
12. Vicente, Kim J.: HCI in the Global Knowledge-Based Economy: Designing to Support Worker Adaptation. ACM Transactions on Computer-Human Interaction, 2 (2000).
13. Zhou, Baoyao, Hui, Siu Cheung, Fong, Alvis C.M., Web Usage Mining for Semantic Web Personalization. PerSWeb05 (2005) http://www.win.tue.nl/persweb/index.html.

Personalization in the EPOS project

Leo Sauermann, Andreas Dengel, Ludger van Elst, Andreas Lauer, Heiko Maus, Sven Schwarz

Knowledge Management Department
German Research Center for Artificial Intelligence DFKI GmbH,
Kaiserslautern, Germany
{firstname.surname}@dfki.de

Abstract. In this work we present the results of the EPOS project with regard to the needs of personalization in the Semantic Web. Focus of this work is the subjective view of an individual person, expressed in a *Personal Information Model (PIMO)*. It is matched both with personal resources (files, e-mails, and websites) of the user and organizational knowledge (ontologies). A user observation component gathers actions of the user to calculate the current context with regards to current goals and matching elements in the user's PIMO. Combined, the representation of the user's stored information and the current context provide a thorough representation of the user. Desktop applications can use this representation to provide personalized services. Three special purpose applications were implemented: a search engine, a context-sensitive assistant, and a tool for filing new information. An evaluation of this approach showed that it increases productivity and indeed reflects the subjective view of users. Also, the approach satisfies most of the requirements of an Adaptive Educational Hypermedia System *AEHS*. Parts of this work are published as open source projects. .[1]

1 Introduction

As Baldoni et al. described in [1]: personalization in the web requires semantic data markup. The same can be said for the desktop computers: we miss the markup.

A prerequisite for personalization is a representation information about the person itself and about the current work context of the person. On desktop computers, we miss the semantic markup of resources like files, e-mails, or browsed websites. Additionally, we miss the representation of the user's current work context, what resources is the user accessing at the moment and what are the goals of the user. On the Semantic Desktop as described in [13, 5] these possibilities exist. The EPOS project is such a Semantic Desktop and addresses these two problems, providing the following solutions:

[1] This work was supported by the German Federal Ministry of Education, Science, Research and Technology (bmb+f), (Grant 01 IW C01, Project EPOS: Evolving Personal to Organizational Memories).

– A representation of the user's personal information items, including e-mails, files, and other data sources using RDF
– A representation of the user's mental model in a formal representation, using several layered ontologies.
– A desktop service to capture the current actions of the user, representing the actions using RDF and then calculating the current context of the user.

In this paper we will give an introduction of the EPOS scenario and describe the solutions relevant for personalization. Note that we reference elements from the OWL and RDF/S specifications without introducing them [3, 9].

1.1 The scenario of EPOS

EPOS aims at supporting knowledge workers doing knowledge-intensive tasks. As an example, a consultant within a German IT company is investigated. The knowledge worker handles information from various information sources, local data sources like files stored on his desktop computer or contacts in his local address-book, organizational data from shared file repositories or ontologies and web resources. While the organization asks for universally applicable and standardized persistent structures, processes, and work organizations to achieve and maintain universally accessible information archives, the individual knowledge worker requests individualized structures and flexibility in processes and work organization in order to reach optimal support for the individual activities.

A problem of conventional file-systems (for example on the windows operating system) is that they do not allow the user to place the same file into two folders, if the folders represent topics and the files represent articles, then the user cannot express the fact that an article covers two or more topics using these structures. A solution to this problem can be reached by using a multi-criterial classification approach (placing the same file in multiple classes), which is possible using graph models like RDF. In EPOS, we allow the user to classify a file stored on the local hard disk by connecting it to one or more entities in the users's Personal Information Model (PIMO).

2 The Personal Information Model (PIMO)

The *Personal Information Model* as created in EPOS was driven by four requirements:

– Sound formal basis: The PIMO must support various knowledge services, among them logics-based services (e.g., ontology-based information retrieval). Therefore, the PIMO must employ an expressive representation language and has to wipe out the contradictions and redundancies of the native structures.
– Bridge between individual and organizational Knowledge Management: The PIMO has to incorporate global ontologies, but also has to reflect the changes and updates of native structures. The PIMO itself should be a source of input for OM-wide ontologies.

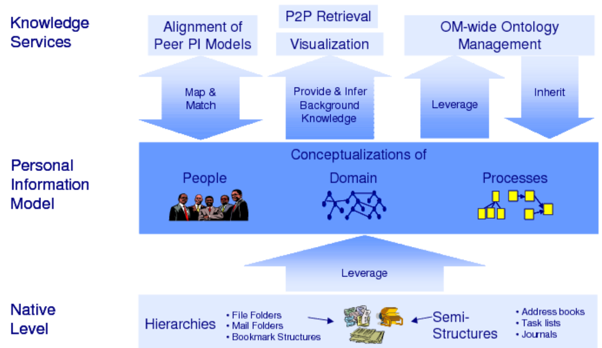

Fig. 1. The Personal Information Model as *semantic middleware* between native structures and knowledge services.

– Maintenance: Adequate means have to be provided that assist the user with stepwise formalization of native structures and inspection of the PIMO.

Figure 1 shows how such a Personal Information Model is embedded in the EPOS information landscape and what basic functionality has to be provided to link the PIMO to the native structures as well as to the envisioned knowledge services. The central part of the PIMO are formal representations of mental models, namely people, concepts, processes, etc. Native resources (files, e-mails, web documents) are expressed using a separate ontology and according services to transform the native data into this ontology. The services for peer-to-peer ontology alignment and organizational-memory (OM) wide ontology management are described in [19].

The PIMO ontology framework created in the EPOS project consists of six components, see figure 2. The first half of these components represent mental models on a conceptual level using formalized domain ontologies. Itself it consists of three layers: upper-level, mid-level and domain ontologies. Native resources (files, e-mails, web-pages) are represented using native data vocabularies. For example, the mental model of the person "Heiko Maus" is formally represented in many native resources, the address book entry of Heiko Maus is a native representation of the mental concept of the living person, which is expressed in a domain ontology or personal model.

2.1 Native structures and data

The personal workspace with its *native structures* like file- and mail-folder hierarchies reflects the worker's personal view of his or her information space. The underlying conceptualizations are therefore a valuable aid not only to guide the worker's information management tasks like storage and retrieval, but also to the internalization and, ultimately, utilization of new information. Furthermore, due

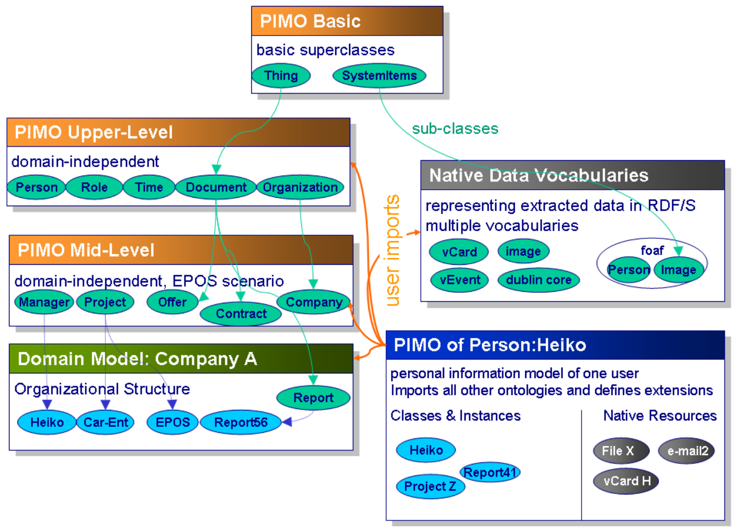

Fig. 2. PIMO ontology components

to their continuous development by the knowledge workers, the *personal* structures provide an excellent input for the acquisition of *organizational* knowledge. However, today's native structures also have some serious drawbacks. They are often built ad hoc, which means they only reflect a snapshot of the worker's view. They lack formal semantics. Therefore, they are hard to exploit by automatic information services.

A framework was created to transform several native structures to the RDF format, this was implemented as part of the gnowsis semantic desktop[12]. By the end of 2005, we have implemented adapters for Microsoft Outlook (e-mails, appointments, address book), IMAP e-mail servers, the Thunderbird address book, several file formats and other applications. External contributors have commited adapers to the Flickr.com photo website (by Anja Jetzsch and Florian Mittag) or relational databases (by Richard Cyganiak). Our approach to extracting the data from native applications using *virtual rdf graphs* was published in[14]. To improve the performance of this approach and to open our results to a wider community, the code for data extraction was moved to the Aperture project on sourceforge [2]. In EPOS, we focused on these data sources of native structures:

- several file formats (bibtex, Microsoft Office, PDF, etc.)
- Microsoft Outlook items (calendar, address book, e-mails, todo list)
- IMAP e-mails
- web-sites browsed in the Firefox browser

[2] (http:
aperture.sourceforge.net)

– weblog entries in the Wordpress blogging system

The data extracted from these native sources is then described using a set of data-oriented RDF/S vocabularies, represented in the layer of *Native Data Vocabularies*. These were selected during creation of the adapter software by the adapter developers. For example, to represent address book items from Microsoft Outlook a vCard[3] RDF/S representation was taken. Note that these vocabularies were matched to the other PIMO ontologies using subclass relations. The relations between native resources and other concepts in the ontology layers can be compared to occurrence-relations in Topic Maps[18].

2.2 PIMO-Basic, PIMO-Upper, PIMO-Mid and Domain Ontologies

Apart from the native structures, the mental models are represented in EPOS using a multi-layer approach. A similar approach was used by Huiyong Xiao and Isabel F. Cruz in their paper on "A Multi-Ontology Approach for Personal Information Management", where they differentiate between *Application Layer, Domain Layer and Resource Layer*. Alexakos et al. described "A Multilayer Ontology Scheme for Integrated Searching in Distributed Hypermedia" in [4]. There, the layers consist of an *upper search ontology layer, domain description ontologies layer, and a semantic metadata layer*.

In EPOS, the ontology layers consist of

- PIMO-Basic: defines the basic language constructs. The class pimo-basic:Thing represents a super-class of other classes.
- PIMO-Upper: A domain-independent ontology defining abstract sub-classes of Thing. Such abstract classes are PersonConcept, OrganizationalConcept, LocationConcept, Document, etc.
- PIMO-Mid: More concrete sub-classes of upper-classes. The EPOS mid-level ontology serves to integrate various domain ontologies and provides classes for Person, Project, Company, etc.
- Domain ontologies: A set of domain ontologies where each describes a concrete domain of interest of the user. The user's company and its organizational structure may be such a domain, or a shared public ontology. Domain ontologies should sub-class from PIMO-Mid and PIMO-Upper to allow integration.
- PIMO-User: the extensions of above models created by an individual for personal use.

The first three layers were created once by members of the EPOS team and are well suited and quite fix for such knowledge work scenarios, the domain ontologies are created for real domains and change frequently. For example, a domain ontology was created to represent the organizational structures at the DFKI KM lab, named "*Organizational Repository*".

[3] http://www.w3.org/TR/vcard-rdf

2.3 The PIMO of an individual user

Using above prerequisites, the *Personal Information Model* of a user can now be created by assembling the different parts for a single user.

- PIMO-Basic, PIMO-Upper, PIMO-Mid are imported unchanged
- One or more domain ontologies are imported. The "Organizational Repository" of a company can be represented as domain ontology.
- The personal mental model of the user is represented in the user's own domain ontology, called PIMO-User. There personal concepts, ideas, projects, contacts etc. are represented and matched to domain ontologies. The user can create his own classes and instances here.
- the native resources on the desktop of the user (files, e-mails, address-book, etc) are converted to data vocabularies using adapters. They are matched to the personal mental model and to domain ontologies.

Hence the *Personal Information Model* (PIMO) of a user can be defined as the sum of imported upper and mid-level ontologies, domain ontologies, one personal mental model of the user (PIMO-User), and the native resources found in heterogenous data sources. It fulfills the requirements stated above: formal basis, bridge between personal and organizational knowledge, maintenance.

Mappings between ontologies were either realized by using subclass/subproperty relations to map classes or by using the custom property pimo:hasOtherRepresentation to express the fact that one instance A1 of ontology O1 is represented in another ontology O2 in the instance A2. A1 pimo:hasOtherRepresentation A2 would be the according triple. These mappings were primarily used to match instances created by the individual user in his individual model to instances in domain ontologies. For example, the user creates an instance for the project "Car-Entertainment" and later connects it to the instance in the organizational repository representing the same project. As mentioned above, automatic algorithms to do this were evaluated in [19].

Based on the PIMO of a user, we can now look at the activities of the user, from which we can derive a model of the user's context.

2.4 User activity and user modeling

In [16] we presented our approach to capture user activity and represent the user's context and goals.

We identified four different levels of abstraction of user activity: The first level, called *Workspace Level*, represents the operating system and the applications that provide access to files, objects and information structures. Observation at this level results in workspace events such as various mouse clicks, entering of some text, or starting and handling applications. Data handled in this interaction is described in the *Native Data Vocabularies*. The *User Action Level* contains such user actions as "create new text document", rather than atomic mouse-clicks or actions like start text editor or activate File-new menu. Those user actions will be inferred from a series of workspace events described before.

While he interacts with the computer, he always has some higher medium term goal in mind such as compiling an offer to a customer or writing a project proposal. The documents handled can be represented in the PIMO ontologies. Those user goals are captured in the *Task Concept Level* and are represented by task concepts which are concepts in an ontology about such user goals. EPOS will elicit the users goal(s) from a sequence of the user actions needed to be carried out to achieve this goal. And, last but not least, the *Process Level* connects to the organizational structures processes which are explicitly modelled in the domain ontology representing the company. If there is a Workflow-Management System (WfMS) available that also interprets the company ontology, we can connect / assign the user to running workflows. Workflows can be semantically described using the same set of task concepts as were eliciting from the users behavior [15]. In EPOS, the integration with the WfMS was realized by a gnowsis adapter [17].

2.5 Relation to existing approaches

The EPOS system as whole is an implementation of a Semantic Desktop, as defined in [13]. Competing semantic desktop implementations, like Haystack, are also discussed there. It is data and application-centric but misses the requirements of personalization, which we find in [1]. In this work titled "*Personalization for the Semantic Web*" we find a formalism for an *Adaptive Educational Hypermedia System* (AEHS). An AEHS is a Quadruple of Document Space *DOCS*, User Model *UM*, Observations *OBS*, Adaptation Component *AC*.

Via the EPOS context elicitation system, the requirements of UM (User Model) and OBS (Observations) are satisfied. The UM can be either seen as the sum of all data represented in one user's PIMO or only by taking the elicitated context. The OBS requirement is satisfied by the user actions that we described in short above and in detail in [16, 15]. Rules were used in all stages of the architecture. A good example for the usage of semantic web rules for personalization is the gnowsis desktop search, which also fulfills the AC requirement.

3 Personalized special purpose applications

Given the formal representation of the user model in the PIMO framework different use cases were identified in the EPOS project. Three such use cases and their respective implementation will be described in this paper, in all three cases, the PIMO framework supplies the ontology data.

3.1 A drop-box for filing

As identified by Indratmo and Cruz[7] and earlier Barreau and Nardi[2], filing information is a crucial task in personal information management. The EPOS solution is to provide a special folder for the user, called *Drop-Box*, where the user can file information. The name is derived from the Mac OS drop-box. When the knowledge worker downloads a new file from the internet, it is stored into the special drop-box folder. EPOS analyzes the content of the file and searches for documents that are similar to the file (using the *Brainfiler* text classification system by Brainbot AG). The documents again match to concepts of the user which again are mapped to the domain ontologies. Note that the training of documents to concepts has to be done beforehand. The result of this matching is that the system can suggest which elements of the PIMO of the user match against the dropped file. The user selects which concepts are really describing the file and the file is moved to a folder that is related to one of the concepts. This approach allows the user to use multiple classifications, breaking with hierarchical file structures. By using the drop-box frequently, the categories are further trained.

3.2 Gnowsis desktop search

For information retrieval, a desktop search tool was created, called *gnowsis desktop search*. Its main input is a fulltext-search field as known from common desktop search tools (google desktop). The search reaches across all parts of the PIMO framework, domain ontologies, native data sources and the PIMO-User. Internally, the search engine represents the search results in a RDF graph model, inspired by the Roodolf RDFS model for the Google api[4]. Our vocabulary contains classes to represent the search request, each returned hit, and more, see here[5]. Representing the search results as RDF before rendering them as HTML output allows EPOS to run personalization rules before rendering the results. These rules are expressed using the Jena[8] Rule syntax as described in the Jena documentation. We also embedded the possibility to call additional SPARQL[10] queries from within the rule engine, hence a rule can decide to expand the search by invoking another search to the ontologies. Personalized sets of these rules can be used to expand the search results (increasing recall values) or to filter out unwanted results (increasing precision). In the EPOS scenario, rules were used to include defined ontology mappings (hasOtherRepresentation links, see above). This is a shortened version of the rule set used in the evaluation of the system:

```
# found something? -> infer other representations via SPARQL\\
(?hit retrieve:item ?x) -> \\
querySparql('CONSTRUCT { ?x pimbasic:hasOtherRepresentation ?y } ')

# found a project? -> also show members \\
(?hit retrieve:item ?project), (?project rdf:type org:Project) ->\\
querySparql('CONSTRUCT   {
  ?project org:containsMember ?m.
}).
```

[4] http://nutria.cs.tu-berlin.de/roodolf/rdfs
[5] http://www.gnowsis.org/ont/gnoretrieve

3.3 Context-sensitive assistance system

This assistant system should react to the current work context of the user (topic, content, project, etc.) , providing help that matches the current goal of the user. It should provide help to the user without explicit invocation of it by the user (*pro-active*). Also, the user is not always aware that assistance is available and therefore the assistant has to work in a fashion that is not disturbing the user, disrupting the user or distracting the user (*non-intrusive*). The approach is similar to the Lumière project[6] by Microsoft Research, which in part was implemented in the Microsoft Office assistant. Also Rhodes' "Margin Notes" project[11] aims at a similar goal: while the user is web-browsing, margin notes loads relevant information from the local files of the user and shows them in a side-bar in the browser.

The EPOS assistant is implemented in a sidebar component. The user interface of the EPOS assistant can be shown or hidden, as the user wishes. The sidebar contains also other components like a desktop search field and an ontology overview, so the assistant is not taking much screen space. It updates itself when the user context model changes based on the user actions (as described above). When the user accesses a resource (for example a web-resource in the browser), related concepts are identified (using text analysis of the resource) and loaded. The assistant then shows only the relevant documents, people, projects and tasks that are related to the current work context. Again, these concepts are taken from the PIMO-Mid ontology level. The assistant therefore benefits from the integration realized in the PIMO framework. It is *pro-active* because it reacts to context changes detected automatically by the context observation service and it is *non-intrusive* because the shown information relates to the current work context and the decent screen space that is occupied.

4 Evaluation of the approach

The EPOS project was evaluated using different methods. First Mark Siebert and Pierre Smits of Siemens Business Services (SBS) adapted the gnowsis desktop search system and compared it to other systems. The rule system provided by the gnowsis desktop search was adapted by them to a scenario at SBS, changing the rules so that they return documents based on a search situation the user is currently in. The results of this evaluation will be published 2006. Then a case study was done where eight knowledge workers at DFKI were observed and questioned over a longer period. Their feedback was collected via two ways: First, a daily interview (questionnaire) provided subjective, qualitative statements, and, second, measuring click counts and explicit user feedback during a one week period delivered qualitative measures. Results of the case-study were:

- The *drop-box* component increased productivity as it allows to file items faster than without the assistance.
- The possibility to add multiple categories to a document was used, in the mean 2.5 categories were attached to a file, which is significantly more than the single category a hierarchical file system provides.

– The *gnowsis desktop search* was used very frequently. Users answered in the questionnaire that they found unexpected information and that the categories provided by the PIMO helped during retrieval.
– The subjects stated, that the context-sensitive assistance came up with unexpected, surprising information items (e.g., documents) revealing new, useful, cross-references.
– The participants agreed that the PIMO reflects their personal mental models.

5 Conclusions and Outlook

The EPOS project addressed the problem of knowledge management in heterogenous environments. Personalization services for desktop applications need semantic markup of the resources found on the desktop pc and on the web. For the scenario of knowledge work in the consulting business, EPOS and the gnowsis framework provide a layered ontology solution, the PIMO framework. It represents information with focus on the subjective view of the user. The activity of the user is observed and captured in a detailed user context model[15]. This model can be retrieved and used for assistance, as shown in the pro-active context-sensitive assistant system. Two other special purpose applications were presented, a drop-box for filing information, and a desktop search system that allows a rule-based expansion of search results. An evaluation of these three components showed that they can improve the productivity of knowledge workers. Parts of EPOS match to components needed for an AEHS and we suggest that our approach can assist Semantic Web personalization as such. Implementations of our work are published as open source projects at www.opendfki.de and will be continued in follow up projects.

The architecture of EPOS and the PIMO framework provide building blocks that enable personalization on the Semantic Desktop. The EU project NEPOMUK aims at standardizing the Semantic Desktop platform and the results of EPOS contribute to this effort. The PIMO ontology as it was developed for EPOS is improved and extended on a regular basis and can be downloaded[6]. Currently, parts of the system are re-implemented under the codename "gnowsis beta", a pre-release was already published in March 2006 and a more stable release can be presented at the ESWC2006 workshop.

Acknowledgements We want to thank Mark Siebert and Pierre Smits who conducted the evaluation of the gnowsis desktop search at Siemens and have helped to keep the system installable and simple. Richard Cyganiak and Anja Jentzsch have accompanies the gnowsis work ever since. Special thanks are directed to our student contributors (in random order): Benjamin Horak, Frank Osterfeld, Dominik Heim, Daniel Bahls, Daniel Burkhart, Günther Noak, Ralf Biedert, Moritz Plößl, Florian Mittag, Emmanuel Gasne, and many more.

[6] http://ontologies.opendfki.de/repos/ontologies/pim/

References

1. M. Baldoni, C. Baroglio, and N. E. Henze. *Personalization for the Semantic Web*, volume 3564, chapter p. 173, pages 173–212. Springer, Aug 2005.
2. D. Barreau and B. A. Nardi. Finding and reminding: File organization from the desktop. 1995.
3. D. Brickley and R. Guha. Rdf vocabulary description language 1.0: Rdf schema. w3c recommendation 10 february 2004. http://www.w3.org/TR/rdf-schema/.
4. K. V. C. Alexakos, B. Vassiliadis and S. Likothanassis. *A Multilayer Ontology Scheme for Integrated Searching in Distributed Hypermedia*. Springer, 2006.
5. S. Decker, J. Park, D. Quan, and L. Sauermann, editors. *Proc. of Semantic Desktop Workshop at ISWC*, volume 175 of *CEUR Workshop Proceedings*, 2005.
6. E. Horvitz, J. Breese, D. Heckerman, D. Hovel, and K. Rommeltse. The lumière project: Bayesian user modeling for inferring the goals and needs of software users. Microsoft Research.
7. Indratmo and J. Vassileva. Human and social aspects of decentralized knowledge communities. In *Proc. of Semantic Desktop Workshop at the ISWC, Galway, Ireland, November 6*, 2005.
8. B. McBride. Jena: Implementing the rdf model and syntax specification. In *Proc. of the Semantic Web Workshop WWW*, 2001.
9. D. L. McGuinness and F. Harmelen. Owl web ontology language overview w3c recommendation 10 february 2004. http://www.w3.org/TR/owl-features/.
10. E. Prud'hommeaux and A. S. (edts). SPARQL query language for RDF. W3c working draft, W3C, 2005. http://www.w3.org/TR/rdf-sparql-query/.
11. B. J. Rhodes. Margin notes: Building a contextually aware associative memory. In *Proceedings of the International Conference on Intelligent User Interfaces(IUI2000)*, January 2000. Software Agents Group, MIT Media Lab.
12. L. Sauermann. The gnowsis-using semantic web technologies to build a semantic desktop. Diploma thesis, Technical University of Vienna, 2003.
13. L. Sauermann, A. Bernardi, and A. Dengel. Overview and outlook on the semantic desktop. In *Proc. of the 1st Workshop on The Semantic Desktop at ISWC*, 2005.
14. L. Sauermann and S. Schwarz. Gnowsis adapter framework: Treating structured data sources as virtual rdf graphs. In *Proceedings of the ISWC 2005*, 2005.
15. S. Schwarz. A context model for personal knowledge management. In *Proceedings of the IJCAII WS. on Modeling and Retrieval of Context*, Edinburgh, 2005.
16. S. Schwarz and T. Roth-Berghofer. Towards goal elicitation by user observation. In *Proceedings of the FGWM WS. on Knowledge and Experience Management*, Karlsruhe, 2003.
17. R. Shkundina and S. Schwarz. A similarity measure for task contexts. In *Proc. of the WS Similarities - Processes - Workflows, ICCBR 2005*, Chicago, 2005.
18. G. M. e. Steve Pepper. Xml topic maps (xtm) 1.0. Specification, TopicMaps.Org, 2001.
19. L. van Elst and M. Kiesel. Generating and integrating evidence for ontology mappings. In *Proc. of EKAW 2004*, 2004.

Verifying the compliance of personalized curricula to curricula models in the semantic web*

Matteo Baldoni, Cristina Baroglio, Alberto Martelli
Viviana Patti, and Laura Torasso

Dipartimento di Informatica — Università degli Studi di Torino
C.so Svizzera, 185 — I-10149 Torino (Italy)
{baldoni,baroglio,mrt,patti,torassol}@di.unito.it

Abstract. In this work we propose the introduction of a decoupling between personalized curricula and curricula models. A curricula model is formalized as a set of time constraints, while personalized curricula are formalized by means of an action theory. Given this framework, it is possible to make various interesting verification tasks automatic. In particular, we will discuss the possibility of verifying the compliance of personalized curricula to models, by using temporal reasoning. Compliance verification allows to check the soundness of a curriculum customized w.r.t. available resources and user goals against a model that expresses temporal learning dependencies at the knowledge level.

1 Introduction

The Semantic Web is concerned with adding a semantic layer to resources that are accessible over the internet in order to enable sophisticated forms of re-use and reasoning. In the last years standard models, languages, and tools for dealing with machine-interpretable semantic descriptions of Web resources have been developed. In this context a strong new impulse to research on personalization can be given: the introduction of machine-processable semantics makes the use of a variety of reasoning techniques for implementing personalization functionalities possible, widening the range of the forms that personalization can assume.

Learning resources are particular kind of resources specifically useful in an educational framework. Especially with the development of peer-2-peer and service oriented e-learning architectures, it become fundamental to explore solutions for personalizing w.r.t. the user's needs the retrieval and the composition of learning web resources. In our opinion sophisticated personalization functionalities

* This research has partially been funded by the European Commission and by the Swiss Federal Office for Education and Science within the 6th Framework Programme project REWERSE number 506779 (cf. http://rewerse.net), and it has also been supported by MIUR PRIN 2005 "Specification and verification of agent interaction protocols" national project.

should combine lesson learnt in the community of traditional educational systems (especially for what concerns the re-use of learning resources), and the new possibility of running reasoning techniques developed in the AI community over the semantically annotated learning resources.

In recent years, the *educational systems* community has focussed greater and greater attention to the problem of separating the *contents* of learning resources, from the *means* that is necessary for taking advantage of the contents. The chief goal is to enable a reuse of the learning resources, where re-use is more and more often intended as a process by which the contents of a new complex learning resource, e.g. a course, are assembled, at least partly, starting from already encoded contents, the optimal situation being a complete decoupling of the resources from the platforms used for playing them. A first significant step in this direction is represented by the birth of SCORM [1] and of Learning Design [13, 14]. The former allows to build a new course (formally, a new SCO) on top of existing SCOs or assets. The latter, is focussed on the design of processes and workflows among a group of actors that take part to the learning activities. These tools, however, suffer the lack of a machine-interpretable information about the learning resources, for enabling forms of automatic composition and of verification, possibly based on reasoning.

Standard languages for semantic annotation like RDF [16] and LOM [12] can be used for filling this lack and adding some meta-data to the resources. In particular by meta-data we can supply information on the learning resources at the *knowledge level*, e.g. knowledge about the *learning objectives* of the resource and its *prerequisites*. Given such kind of annotation, we can interpret a learning resource as an action, that can profitably be used if the learner has a given set of competences (preconditions); by using it, the learner will acquire a new set of competences (effects). As we have shown in previous work [3, 2], given an annotation of resources with preconditions and effects one can rely on a classical theory of actions and applying different reasoning techniques for offering different kind of personalization functionalities. For instance, one could use *classical planning* for performing curriculum sequencing, i.e. for selecting and sequencing a set of resources which will allow a user to achieve her/his learning goal [3]. Moreover it possible to exploit *temporal projection* for validate a student-given curriculum verifying whether all the preconditions are respected [2]. Last but not least it is possible to exploit *procedural planning* for performing curriculum sequencing at the level of university courses, in order to help a student to customize a curriculum offered by the University w.r.t his/her interest [2]. In our previous work all these tasks were accomplished by exploiting the metaphor of learning resources as actions and the representation at the knowledge level of the student learning goal and knowledge profile. We exploited a reasoning engine based of the logic language DyLOG [4], that provided a unified framework for performing both classical and procedural planning and temporal projection.

In this work we aim at taking a step further on this line of research and we focus on a kind of verification that can be profitably combined with the curriculum sequencing personalization functionalities investigated in previous work, by leading to implement sophisticated personalization applications in a unified framework. Given a semantic annotation of the resources based on the metaphor of resources as actions, we will focus on a new kind of reasoning, which can be accounted as a *compliance* verification of personalized curricula w.r.t. a curricula model. Personalized curricula are intended as learning paths through learning resources personalized w.r.t. specific user need, e.g. they could be the result of a curriculum sequencing method that exploits the planning techniques mentioned above. Curricula models specify general rules for building such paths and can be interpreted as constraints. These constraints are to be expressed in terms of knowledge elements, and maybe also on features that characterize the resources. If such resources are courses, they should not be based, generally speaking, on specific course names. So a constraint might impose a lab course to be attended after a theory course on the same topics but not that the course C123 should follow C122.

Verifying the compliance a curriculum to a model means checking: first of all, that the resources are sequenced in such a way that their preconditions are respected, that the learning goal is achieved in the end, and that along the sequence the constraints imposed by the model are satisfied. In the following we present a preliminary proposal for a knowledge representation that suits the outlined problem domain and sketch the techniques by which the comparison of courses to constraint-based schemas can be performed.

Compliance verification can be useful in many practical cases where the need of personalizing learning resource sequencing w.r.t. to the student desire has to be *combined* with the ability to check that the result of personalization fit some abstract constraints, possibly imposed by a third party. A given University could, for instance, certify that the specific curricula that it offers for achieving a certain educational goal -that built upon the local university courses- respect some European schemes defined at the abstract level of competence. Such automatic checking of compliance combined with curriculum sequencing techniques could be used for implementing processes like cooperation in curriculum design and curricula integration which are actually the focus of the so called *Bologna Process* [8], promoted by the EU ministers responsible for higher education: "Curriculum design means drawing up of a common study path aimed at reaching the educational goals that have been jointly defined. In these schemes the partners offer specific segments which complement the overall curriculum designed". Further use cases are sketched in the conclusions.

2 Knowledge representation and verification

In this section we discuss about the possible formal representations of *specific curricula*, intended as sequences of learning resources (e.g. documents or entire courses) and *curricula models*, intended as specifications of general schemata for achieving a certain *educational goal*, where relationships among competencies are described.

2.1 Description of resources based on an action theory

Let us consider a specific curriculum as a sequence of resources, that are homogeneous in their representation. Based on work in [2, 3], we represent such resources in an action theory, taking the abstraction of resources as simple actions. We interpret a learning resource as the action of acquiring some knowledge elements (effects); it can be used only if the user owns given knowledge elements or competencies (preconditions). Thus, a resource can be described in terms of knowledge elements. For instance let us use a classical STRIPS-like notation for describing the resource called *db_for_biotech* with prerequisites *relational databases* and effects *scientific databases* as:

ACTION: db_for_biothec(),
PREREQ: relational_db, EFFECTS: scientific_db

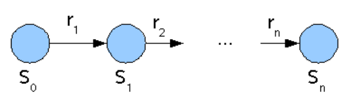

Fig. 1. The labels on the edges, r_1, r_2, ... r_n, represent learning resources. The states S_i represent sets of competences that are available at a given time.

As mentioned in the introduction, the idea is to introduce a semantic annotation of learning resources that describe both their pre-requisites and effects, as done in the curriculum sequencing application in [3].

A curriculum is a sequence of resources/actions. Actions in the sequence cause transitions from a state to another, starting from an initial state up to a final state. The *initial state* represents the initial set of competences that we suppose available before the curriculum is taken (e.g. the basic knowledge that the student already has). This set can also be empty. The subsequent states are obtained by applying the actions (resources) that tag the transitions. Each of such actions has a set of preconditions that must hold in the state to which the action is applied and cause some modifications that consist in an update of the state. The prerequisites of action r_i must hold in the state S_{i-1}. The state S_i is obtained by adding to S_{i-1} the effects of r_i. See Figure 1. We assume that competences can only be added to a state after executing the action of attending a course (or more in general reading a learning material). The intuition behind this assumption is that no new course will ever erase from the students memory the concepts acquired in previous courses, thus knowledge grows incrementally. Formally, it correspond to assume that the domain is monotonic.

2.2 Curricula models

Curricula models are to be defined on the basis of knowledge elements as well. In particular, we would like to restrict the set of possible sequences of resources, by imposing constraints on the order by which knowledge elements are added to the states, e.g. "a knowledge element α is to be acquired before a knowledge element β", "a knowledge element α is guaranteed to be acquired", "the acquisition of α implies that also β will be acquired subsequently". Therefore we will represent a curriculum model as a set of *temporal constraints*. Being defined on knowledge elements, a curriculum model is independent from the specific resources that are taken into account, then, it can be used in an open and dynamic world like the web. A set of similar constraints defines a schema that can be used for checking user specific curricula intended as sequences of actual resources.

A natural choice for representing temporal constraints on action paths is linear-time temporal logic (LTL) [7]. This kind of logic allows the verification that a property of interest is true for all the possible executions of a model, which in our case corresponds to the specific curriculum. This is often done by means of model checking techniques [6]. The curriculum that we mean to check is, indeed, a Kripke structure; as thus, it is easy to verify properties expressed as temporal logic formulas. Briefly, a Kripke structure identifies a set of states and transition relation that allows passing from a state to another (see Figure 1). In our case, the states correspond to the competencies that are owned at a certain moment. Since we assume the domain is monotonic in the sense pointed out in the previous subsection, states will contain *all* the competencies acquired up to that moment. The transition relation is given by the actions that are contained in the curriculum that is being checked. Since the sequence is linear and shows no branch, then, it is possible to reason on the states and with LTL logic it is possible to verify that a given formula holds starting from a state or that it holds for a set of states.

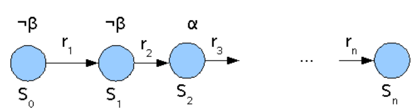

Fig. 2. β can hold only after α becomes true, therefore, in the states that follow S_3 β can either hold or not hold.

For example, the fact that a knowledge element β cannot be acquired before the knowledge element α is acquired, can be written as the LTL temporal formula $\neg\beta\ U\ \alpha$, where U is the *weak until* operator (see Figure 2). Given a set of knowledge elements to be acquired, such constraints specify a partial ordering of the same elements. Other kinds of constraints might be taken into account. For instance, that a knowledge element will be acquired sooner or later ($\Diamond\alpha$, eventually operator). A curriculum model is meant to allow the achievement of

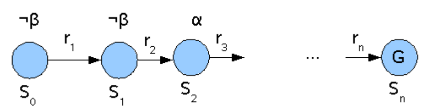

Fig. 3. A curriculum that allows the acquisition of the learning goal $\mathcal{G}$.

a given *learning goal*, that consists in a set of knowledge elements. We expect that the learning goal will hold in the final state of every curriculum that matches with the model (see Figure 3).

2.3 Compliance Verification

Given a representation of a user specific curriculum as sequence of actions/resources $(r_1, r_2, ..., r_n)$ with preconditions and effects, based on knowledge elements in an action theory ($\mathcal{A}$), and a representation of a curricula model, based on temporal constraints ($\mathcal{T}$) and a learning goal ($\mathcal{G}$), it is possible to apply different reasoning techniques for performing various interesting tasks. Besides planning, that we have already explored in previous works [2, 3], in this formal framework we can verify the *compliance* of s user specific curriculum to a model. The verification could be based on temporal reasoning techniques, like temporal projection, and on model checking techniques. Verifying the compliance means, in simple words, to check whether the curriculum respects the model, i.e that the sequence $r_1, ..., r_n$ is sound w.r.t. the precondition and effect relations specified in $\mathcal{A}$, that the sequence allows reaching the goal $\mathcal{G}$, and that the sequence respects the temporal constraints in $\mathcal{T}$. Intuitively, we can think of combining temporal projection and model checking by verifying

$$\mathcal{A} \models_{AL} \mathcal{G} \textbf{ after } r_1, ..., r_n \qquad (*)$$

where AL is any action logic that supports temporal projection, and

$$r_1, ..., r_n \models_{LTL} \mathcal{T} \qquad (**)$$

where LTL is a linear-time temporal logic.

3 Possible implementation

In the following we discuss the possibility of exploiting existing technology and languages for developing a system that can perform the forms of verifications described above. In particular we will deal both with the selection of languages for the representation of models and of curricula, and with the exploitation of existing tools for performing the verifications. A semantic representation of an action is quite simple and mostly consists of two lists of knowledge elements: those required for using the resource and those supplied by the resource. In order for the knowledge elements themselves to have a semantic value, they can be implemented as terms in shared vocabulary (the simplest form of ontology). RDF can be used as an implementation language. Resources are used to define curricula. In this work we focus on curricula obtained by sequencing resources, therefore we represent a curriculum as an action sequence. This sequence can be considered as a simple kind of program that contains no branch or loop or recursive call. For what concerns action representation, there is a wide choice of action languages that are valuable candidates: we could use a logic programming action language, like $\mathcal{A}$ by Gelfond and Lifschitz [9], DyLOG [4], or GOLOG [15], all of which provide proof procedures that support temporal projection (*).

Given that a curriculum has passed the temporal projection test, we can use a model checker to verify the temporal constraints (**). Model checking is the algorithmic verification of the fact that a finite state system complies to its specification. In our case the specification is given by the curriculum model and consists of a set of temporal constraints, while the finite state system is the curriculum to be verified. Among the various model checkers that have been developed, it is worthwhile to mention SPIN [11] and NuSMV [5]. SPIN, in particular, is used for verifying systems that can be represented by finite state structures, where the specification is given in an LTL logic. The verification algorithm is based on the exploration of the state space. This is exactly what we need for performing the second step of our compliance test, provided that we can translate the curriculum in the internal representation used by the model checker. In the case of SPIN, the internal representation is given in the Promela language. For example, we can represent the knowledge elements as boolean variables, therefore actions as transitions that modify the values of some of these variables. The constraints will be temporal formulas that use such variables. The verification that the constraint should along the whole curriculum is performed automatically by the model checker.

In the case of linear curricula it would be easy to integrate in the temporal projection algorithm the direct verification of the constraints. The opposite solution of integrating the temporal projection into a model checker, which is the one that we mean to pursue, has the advantage of allowing the extension of the compliance test to curricula that have a more complex structure. In fact, curricula might contain tests, branching points, and repetitions. For example, if the curriculum corresponds to a learning resource that has been assembled on the basis of other learning resources (for instance a SCORM object), it might contain, as well as a program, also loops. As well as in the two-steps solution described above, it would be necessary to have a translation mechanism that allows turning the representation of the action theory into the internal formalism, used by the model checker [10].

For the sake of completeness, hereafter, we report a part of the Promela code for an example. The code allows the execution of both temporal projection and model checking. Temporal projection is handled as a deadlock verification: if the sequence is correct w.r.t. the action theory, no deadlock arises, otherwise a deadlock will be detected. The complete example and an explanation of it are available at `http://www.di.unito.it/~alice/ccompliance/`. This curriculum passes the compliance test under the temporal constraints $\neg f7\ U\ f5$ and $\neg f8\ U\ f5$. In the web site it is possible to retrieve also examples of curricula which fail the test.

```
mtype = { course1, course2, course3, course4, course5 };
mtype = { done, stop, success, fail }
chan attend = [0] of { mtype };
chan feedback = [0] of { mtype };
bool f1, f2, f3, f4, f5, f6, f7, f8;

init {    f1 = true; f2 = false; f3 = false; f4 = false;
          f5 = false; f6 = false; f7 = false; f8 = false;
          run TestCompliance();
          run UpdateState();  }

inline Curriculum4() {
    attend!course1; feedback?done;
    attend!course2; feedback?done;
    attend!course5; feedback?done;
}

proctype TestCompliance() {
    Curriculum4()
    feedback!stop; feedback?success;
}

proctype UpdateState() {
    do
    :: attend?course1 -> if
       :: (f1) -> f2 = true; f3 = true; f4 = true; feedback!done;
       fi;
    :: attend?course2 -> if
       :: (f3) -> f4 = true; f5 = true; feedback!done;
       fi;
    :: attend?course3 -> if
       :: (f2 && f6) -> f7 = true; f8 = true; feedback!done;
       fi;
    :: attend?course4 -> if
       :: (f2 && f5) -> f7 = true; feedback!done;
       fi;
    :: attend?course5 -> if
       :: (f2 && f4) -> f7 = true; f8 = true; feedback!done;
       fi;
    :: feedback?stop -> if
       :: (f4 && f5 && f8) -> feedback!success;
       :: else -> feedback!fail;
       fi;
       break;
    od  }
```

The above program is hand-coded but, as the modularity of the example witnesses, it would be easy to produce an automatic translator able to turn the description of sets of courses and the description of sequences of resources into Promela code. Such code could, then, be validated according to curricula models encoded as sets of temporal constraints.

4 Conclusions

In this work we have presented a two-level representation of curricula, aimed at capturing the distinction between curricula and models of curricula that define general rules or constraints to be satisfied. We have shown that by implementing curricula models as temporal constraints, and curricula as sequences of actions, it is possible to verify the compliance of a curriculum to a model by exploiting reasoning techniques that combine temporal projection and model checking.

The possibility of verifying the compliance of curricula to models is extremely important in many applicative contexts where the need of personalizing learning resource sequencing w.r.t. to the student desire has to be combined with the ability to check that the result of personalization fit some abstract models. In this sense we can say that the compliance verification we propose is complementary w.r.t the capability of applying planning techniques for building from a set of available resources, personalized curricula aimed at reaching a given learning goal. Representing models as sets of constraints gives great freedom in the definition of specific curricula because it cuts away the undesired curricula without imposing unnecessary constraints. The same freedom is not supplied if we represent, as in [2], models as procedures. Procedures have a prescriptive nature that over-rules the possible solutions; the greater flexibility introduced by the use of temporal constraints has a positive effect on the possible personalization of the solutions, by allowing a greater autonomy in selecting among alternatives.

Concerning use cases, we have already mentioned the Bologna process. Another practical application could be helping a teacher that must teach a same topic to different classes, with background and purposes that vary. For instance, to teach Java to a University class as well as to professionals that work in an information technology enterprise. The teacher might be interested in the fact that all students of both classes acquire a same set of competences, with known time constraints, however, since the target students are so different it is useful to prepare two different courses exploiting different learning resources. The University students must, in fact, be taught also the theoretical background concerning object-oriented programming. On the other hand, the professionals will surely be more interested in more practical lessons, containing many real-world examples of application. The teacher might select public-domain (semantically annotated) learning resources from on-line repositories and use them to compose two different curricula personalized w.r.t. the different student targets. Nevertheless, by applying the approach that we have proposed he would have the possibility of verifying that the built curricula respect an abstract curriculum schema, derived from the expertise and the experience of the teacher himself.

We are working at the actual development of a system on the line of the sketch described in the previous section. Moreover, we are thinking to an extension (both from a formal and an implementation perspective), in which hierarchies of knowledge elements are used instead of plain vocabularies. Hierarchies allow a representation of knowledge elements at different levels of abstraction, thus they would allow other forms of verification. In order to include them, it might be necessary to integrate forms of ontological reasoning in the framework.

Acknowledgements

The authors would like to thank prof. Nicola Henze for the precious discussions.

References

1. ADLnet. Scorm specification. Available at http://www.adlnet.org.
2. M. Baldoni, C. Baroglio, and V. Patti. Web-based adaptive tutoring: an approach based on logic agents and reasoning about actions. *Artificial Intelligence Review*, 2004.
3. M. Baldoni, C. Baroglio, V. Patti, and L. Torasso. Reasoning about learning object metadata for adapting scorm courseware. In In L. Aroyo and editors C. Tasso, editors, *Proc. of EAW'04: Methods and Technologies for personalization and Adaptation in the Semantic Web*, pages 4–13, 2004.
4. M. Baldoni, L. Giordano, A. Martelli, and V. Patti. Programming rational agents in a modal action logic. *Annals of Mathematics and Artificial Intelligence*, 2004.
5. Alessandro Cimatti, Edmund M. Clarke, Fausto Giunchiglia, and Marco Roveri. NUSMV: A new symbolic model checker. *International Journal on Software Tools for Technology Transfer*, 2(4):410–425, 2000.
6. O. E. M. Clarke and D. Peled. *Model checking.* MIT Press, Cambridge, MA, USA, 2001.
7. E. A. Emerson. Temporal and model logic. In *Handbook of Theoretical Computer Science*, volume B, pages 997–1072. Elsevier, 1990.
8. Education European Commission and Training. The bologna process. http://europa.eu.int/comm/education/policies/educ/bologna/bologna_en.html.
9. M. Gelfond and V. Lifschitz. Representing action and change by logic programs. *Journal of Logic Programming*, 17:301–321, 1993.
10. Fausto Giunchiglia and Paolo Traverso. Planning as model checking. In *ECP*, pages 1–20, 1999.
11. Gerard J. Holzmann. The model checker SPIN. *Software Engineering*, 23(5):279–295, 1997.
12. Learning technology standards committee. draft standard for learning object metadata, 2002.
13. IMSGlobal. Learning design specifications. Available at http://www.imsglobal.org/learningdesign/.
14. R. Koper and C. Tattersall. *Learning Design: A Handbook on Modelling and Delivering Networked Education and Training.* Springer Verlag, 2005.
15. H. J. Levesque, R. Reiter, Y. Lespérance, F. Lin, and R. B. Scherl. GOLOG: A Logic Programming Language for Dynamic Domains. *J. of Logic Programming*, 31:59–83, 1997.
16. W3C. RDF Primer. Available at http://www.w3.org/TR/rdf-primer/.

Personalizing Relevance on the Semantic Web through Trusted Recommendations from a Social Network

Tom Heath, Enrico Motta

Knowledge Media Institute, The Open University,
Walton Hall, Milton Keynes, MK7 6AA, United Kingdom
{t.heath, e.motta}@open.ac.uk

Abstract. Personalization efforts to date have centred on presenting web users with novel items by predicting what they may find relevant. This approach has utility where the user is unsure of exactly what they are looking for, but not where they have a particular information need to satisfy or a particular item to locate. Furthermore, by operating purely on a predefined database of users and items, systems using this approach represent closed worlds and offer poor scalability to new data sets. To address these limitations we propose a technique for personalizing relevance in information seeking activities, based on an understanding of how people seek information and recommendations from their social network. We then describe technical work in progress, based on Semantic Web technologies, that aims to realize this perspective.

1 Introduction

Whilst web sites offering personalization based on conventional web technologies are numerous, the emerging Semantic Web provides an opportunity for richer personalization features to be developed. The availability of structured data adhering to common ontologies enables the integration of user-relevant content from more diverse sources. More importantly however, by allowing users to describe aspects of their context (such as the social networks they are part of) in a standardized way, Semantic Web technologies enable new forms of personalization. In this paper we describe our approach to personalizing relevance in information seeking, through use of Semantic Web technologies and recommendations from social networks.

1.1 Approaches to Personalization on the Web

Personalization on the web has been approached in a number of different ways [1]. Some web sites allow a personalized user experience through changing color schemes or selecting news channels to be displayed on an individual homepage [2]. An alternative approach involves using *recommender systems* [3] [4] [5] that have some knowledge of the user's preferences to select and present content presumed of interest to them, thereby highlighting items of which they may not be aware.

Despite being widely used, these recommender system-based approaches to personalization have a number of limitations. Firstly from a technical point of view, they operate as closed worlds, whereby recommendations can only be given about items that exist within the system, and only the purchase histories of users within the system can contribute to recommendations based on so-called *collaborative filtering* [3]. Secondly, these forms of personalization by speculative recommendation only support the user in carrying out tasks where the solution is unknown or poorly defined, rather than tasks where they have a particular information need to satisfy or a particular item to *locate* [6]. This ignores the role personalization could play in supporting information seeking activities through tailoring relevance to the individual.

1.2 Relevance in Information Seeking

Despite the vast extent of online resources, locating the required piece of information can still present challenges to the user; it may not yet be available on the web, or where a query yields many search results it may be difficult to identify the most appropriate. Resolving these issues requires the identification of sources of additional information not currently available on the web, or means of filtering information based on relevance to the individual's information needs.

Literature on information retrieval has traditionally viewed relevance as a measure of the suitability of a result to the information need of the user *as it is expressed in a query issued to the system*. This relationship between document and query has been referred to as *topical relevance* [7]. However, it would be desirable to measure suitability of the result in relation to the abstract information need of the user [8], whether or not this has been adequately expressed in the query. Building systems that enable such a *personal relevance* requires additional knowledge about the user to be taken into account that may be difficult to express via keyword search.

Attempts to address this issue include [9], where results on a job-seeking site are filtered according to a user profile generated from page view data. However, this approach only enables filtering and personalization to be carried out on items within the closed world of jobs already listed on the site.

1.3 Relevance through Recommendations from a Social Network

Our approach to personalizing relevance is based on identifying the members of their social network the user is most likely to trust as an information source in a given scenario, and using recommendations from these people to personalize search results. Whereas search engines and recommender systems attempt to identify items appropriate to the user, we advocate an approach that identifies the most appropriate sources as a means to identify relevant items.

This source-centric approach using known members of a social network (that we call *Known Person Recommendation*) allows for more complex reasoning to be carried out about the appropriateness of a source than is possible with collaborative filtering systems where other users are unknown. Furthermore, a relevance system driven by social networks is constrained in scope only by the knowledge of the members of the network, and the ability to infer the source most appropriate to the task. Consequently the approach is not limited to selected domains, as is often the case with existing closed world recommender and personalization systems.

In previous research [10] we identified five factors (*expertise*, *experience*, *impartiality*, *affinity*, and *track record*) that determined from which members of their social network a person would seek recommendations. The criticality and subjectivity of the task were found to influence which factors were most attended to. These findings have informed the technical implementation described below.

2 A System for Personalized Relevance on the Semantic Web

We are currently developing a system to test personalized relevance in locating information about travel resources such as hotels, restaurants, and cultural sights (referred to here as "travel objects"). An architectural overview is shown in Figure 2.

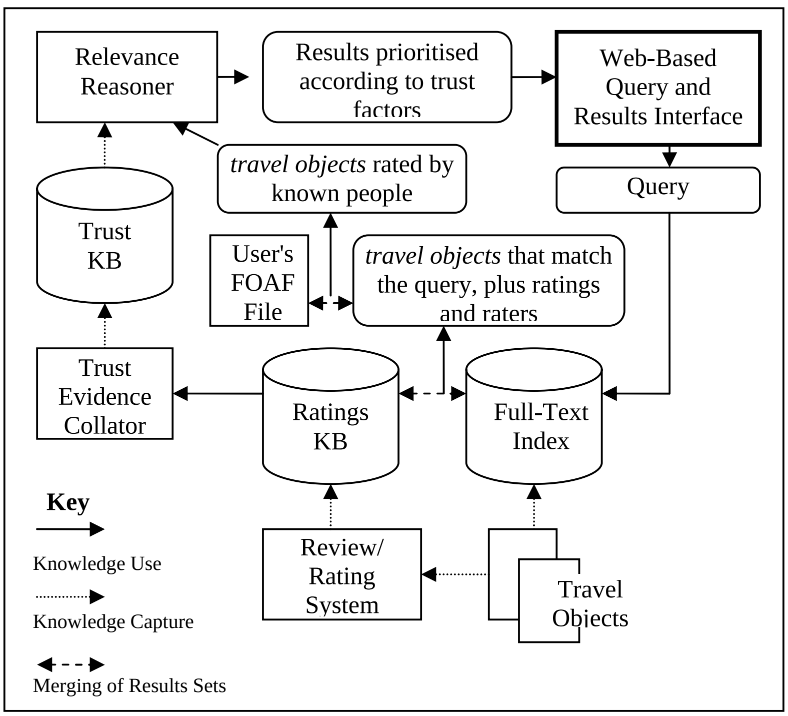

Fig. 1. System Architecture for Personalizing Relevance on the Semantic Web

The system is being piloted in the travel domain to evaluate the feasibility of personalizing relevance in low-criticality high-subjectivity tasks. Consequently, the trust factors being prioritized are **experience** and **affinity**, found in [10] to be most relied upon in such tasks. If successful, the system will be applied across scenarios with a wider range of characteristics, making use of the full range of trust factors.

Users will provide two types of input to the system: firstly semantic descriptions of their social networks using the FOAF vocabulary [11]. This will give the user freedom and control in how they manage this information, and how much is divulged. Secondly users will use the review/rating system to provide reviews of travel objects. This will serve to populate the knowledge base with information about travel objects that users may wish to locate, and provide data from which trust relationships between a user and members of their social network can be inferred. The extent of these trust relationships will form the basis for providing personalized relevance to users of the system. Whilst it is theoretically possible for users to explicitly provide information about trust relationships the demands this would place on the user are deemed prohibitive.

Use of Semantic Web technologies [12] as a platform for developing such a system helps overcome the limitations of closed world recommender systems in a number of ways. Firstly by acting as a large distributed database, the Semantic Web enables any item to be included within a recommender system, not just those within a centrally administered catalogue. Secondly, providing recommendations are made available in standard formats, users can access the knowledge of members of their social network without all members having to subscribe to the same services, as these can be harvested from multiple locations and easily integrated to form a knowledge base.

2.1 Inferring Trust Relationships from Reviews of Travel Objects

Trust relationships will be inferred based on the affinity between the user and each member of their social network, and the relevant experience of each member of the network.

The experience a member of ones network has of a particular location will be determined by the number of travel objects they have reviewed/rated from that location. Where possible the URI of a travel object will be de-referenced to obtain a machine-readable description of its location. Where such a machine-readable description is not available, tags provided by the user and full-text indexing of the content at the object's URI will offer a best guess as to where it is located.

One potential confound of relying on number of objects rated to infer experience is variation in users' enthusiasm for rating and reviewing objects. The use of evidence about affinity in addition to experience will help to mitigate negative effects of this variation. Whether an affinity exists between a user and a member of their social network will be determined automatically by a combination of the following factors: the extent to which both parties have rated the same tourism objects (i.e. the overlap in rated objects), and the correlation between the ratings given by each party.

Due to the reliance on ratings from users for both population of the system and generation of evidence of trust relationships, the system may suffer from bootstrapping problems. In this case it may be possible to fall back on alternative sources of evidence for factors such as experience. For example, where individuals have shared their photos online and have tagged these with place names or locations, this could be used to infer that they have some experience of this location. Similarly if an individual has studied or worked in a particular place, as expressed in their FOAF file, this would suggest they have some experience of the locality. Whilst in these cases no specific tourism objects would be available to be recommended by the system as the users had not yet provided ratings, the user could still benefit from identification by the system of potential sources of information.

3 Conclusions and Future Work

Implementation of the system described above is underway. However, a number of challenges remain in both the implementation and adoption of such a system. Whilst we believe the system to be based on sound principles, it is not clear how well the system will scale from a user perspective. A point may be reached where the system contains a volume of knowledge that renders even the relevance mechanisms proposed here unworkable. Such a scenario is analogous to how search engine algorithms have had to be modified as the web has grown in size.

Furthermore, users' views on provision of social network information on the public web will need to be taken into account. Whilst this is fairly widespread among Semantic Web early adopters and users of services such as Tribe.net[1], it is not sufficiently common at present to enable widespread deployment of the proposed system. It also remains to be seen whether the majority of users will be prepared to make this information available for use by the system. Encryption of FOAF files may be required to overcome this issue.

Despite the challenges described, the Semantic Web remains the most appropriate platform for implementing this form of personalized relevance. Achieving the same degree of functionality using conventional web technologies would require the creation of a single system with a vast range of functionalities, and adoption of this one system by all users. It is unlikely that such a closed world approach would gain sufficient uptake by users to reach critical mass.

Following implementation and deployment the system will be evaluated to assess its effectiveness in personalizing relevance for users locating travel information. In addition to collecting quantitative measures such as number of users and number of items rated, users' qualitative experiences of using the system will be recorded. Results provided by the system will also be compared to a baseline set of results from a conventional search engine.

If deemed acceptable and valuable by users, such a system for personalized relevance has the potential to democratize recommendation and personalization on the Semantic Web. At present, personalization is largely limited to closed worlds. Organisations such as e-commerce sites that collect data about user preferences have little incentive to make this data available for use by competing services. However, a system that enables reviews to be made in an open, public world allows many competing services to use this knowledge in providing novel services to the user.

[1] http://www.tribe.net

Acknowledgements

This research was partially supported by the Advanced Knowledge Technologies (AKT) project. AKT is an Interdisciplinary Research Collaboration (IRC), which is sponsored by the UK Engineering and Physical Sciences Research Council under grant number GR/N15764/01. The AKT IRC comprises the Universities of Aberdeen, Edinburgh, Sheffield, Southampton and the Open University.

References

1. Mobasher, B., Cooley, R., Srivastava, J.: Automatic personalization based on Web usage mining. Communications of the ACM 43 (2000) 142-151
2. Manber, U., Patel, A., Robison, J.: Experience with personalization on Yahoo! Communications of the ACM 43 (2000) 35-39
3. Herlocker, J. L., Konstan, J. A., Riedl, J.: Explaining Collaborative Filtering Recommendations. In: Proc. CSCW2000 (2000) 241-250
4. Linden, G., Smith, B., York, J.: Amazon.com Recommendations: Item-to-Item Collaborative Filtering. IEEE Data Engineering Bulletin (2003) 76-80
5. Balabanovic, M., Shoham, Y.: Fab: content-based, collaborative recommendation. Communications of the ACM 40 (1997) 66-72
6. Heath, T., Dzbor, M., Motta, E.: Supporting User Tasks and Context: Challenges for Semantic Web Research. In: Proc. ESWC2005 Workshop on End-User Aspects of the Semantic Web (UserSWeb) (2005)
7. Eisenberg, M., Schamber, L.: Relevance: The Search for a Definition. In: Proc. 51st Annual Meeting of the American Society for Information Science (1988)
8. Park, T. K.: Toward a theory of user-based relevance: a call for a new paradigm of inquiry. Journal of the American Society for Information Science 45 (1994) 135-141
9. Bradley, K., Rafter, R., Smyth, B.: Case-Based User Profiling for Content Personalisation. In: Proc. International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH2000) (2000) 62-72
10. Heath, T., Motta, E., Petre, M.: Person to Person Trust Factors in Word of Mouth Recommendation. In: Proc. CHI2006 Workshop on Reinventing Trust, Collaboration, and Compliance in Social Systems (Reinvent06) (2006). Available from http://www.cs.ucl.ac.uk/reinvent06/
11. Brickley, D., Miller, L.: FOAF Vocabulary Specification.http://xmlns.com/foaf/0.1/ (2005)
12. Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web. Scientific American 284 (2001) 34-43

Author Index