Workshop on End Users Aspects of the Semantic Web

29th May 2005, Heraklion, Greece
http://kmi.open.ac.uk/events/usersweb/

The aim of this workshop is to look at how an "ordinary user" might be able to tap into the resources of the Semantic Web, find out about the value of these resources for their work practice or their general web use, and feel compelled to use and perhaps even contribute to Semantic Web resources**.**

A substantial part of current research is going into the creation and aggregation of semantic content. The content is necessary but insufficient condition for the Semantic Web. It is a means to improving the end user's interaction with knowledge repositories. Thus, this workshop considers not only the usual "What content?" and "How to author the content?" questions, but also *"Why, for which purposes, and how could content be (re-)used and re-purposed?"*

Users often move between several modalities and use tools, each designed for a particular purpose and audience. The pervasiveness of the standard Web is partly due to its appeal to non-specialists and immediate feedback when authoring HTML content. We want to look at the developments in making the Semantic Web more accessible and comprehensible to the end users. How can we facilitate the participation of these non-specialists in the development of the Semantic Web and its transplantation from a research incubator into everyday practice? What role does "instant gratification" to the user play in getting him or her involved in specifying and carrying out complex tasks within the Semantic Web? What tools and interfaces are likely to provide such a reward and thus help to break the barriers to the adoption of distributed environments and simplify interaction with large knowledge repositories?

We are very pleased that we received such a positive response and as a result we can offer a subset of fourteen papers that were accepted as either short or long research papers. We also have a paper accompanying UserSWeb's invited talk, which is this year given by Marja-Riitta Koivunen, an author and lead researcher on the Annotea project affiliated with the W3C.

The submitted papers can be broadly classified in the following themes:

- Information Extraction, Web Mining and Ontology Mapping
- Semantic Portals and Semantic Navigation,

- Collaborative Filtering and Knowledge Sharing,
- Semantic Services and Interfaces for Information Delivery

I would like to thank all reviewers who helped with what was not an easy task of commenting on the submitted papers with the aim of improving and clarifying them. In addition to the members of the Organizing Committee, we are grateful to Michele Pasin (UK), Tom Heath (UK), Yuangui Lei (UK), Jan Paralic (Slovakia), Peter Bednar (Slovakia) and Peter Butka (Slovakia) for their assistance.

Also, I would like to express my gratitude to Tim Chklovski, Hideaki Takeda and Maria Vargas-Vera for their contribution to making the initial vision of a workshop focused on the user aspects of the semantic web a reality. I am especially grateful for many interesting discussions, comments and suggestions that made this workshop proposal successful at ESWC 2005.

Finally, a share of thanks belongs to the participants of the very first UserSWeb workshop, which has been held in conjunction with the European Semantic Web Conference (ESWC 2005, http://www.eswc2005.org) in a beautiful, historical Crete.

Martin Dzbor

TABLE OF CONTENTS

ORGANIZING COMMITTEE

Chair & main contact:

Martin Dzbor, *KMi, The Open University, UK*

Co-chairs:

Hideaki Takeda, *National Institute of Informatics, Japan*
Maria Vargas-Vera, *KMi, The Open University, UK*

Program Committee:

Hamish Cunnigham, *DCS, University of Sheffield, UK*
Jörg Diederich, *L3S, University of Hannover, Germany*

Carole Goble, *DCS, University of Manchester, UK*
Yoshinori Hijikata, *Osaka University Japan*
Judy Kay, *University of Sydney, Australia*
Atanas Kiryakov, *OntoText Lab, Sirma Group, Bulgaria/US*
Henry Lieberman, *MIT, US*
David Robertson, *AIAI, Edinburgh Univ. UK*
York Sure, *AIFB, University of Karlsruhe, Germany*
Michael Uschold, *Boeing, US*

Annotea and Semantic Web Supported Collaboration

Marja-Riitta Koivunen, Ph.D.
Annotea project

Abstract

Like any other technology, the Semantic Web cannot succeed if the applications using it do not serve the needs of the users. Annotea is a Semantic Web based project for which the inspiration came from users' collaboration problems in the Web. It examined what users did naturally and selected familiar metaphors for supporting better collaboration.

The selected metaphors were a good match also for demonstrating the use of the Semantic Web technologies. Metadata was generated in the form of Annotea objects. It enhanced collaboration by adding flexibility to the applications and easy creation of different views. Furthermore, Annotea objects also let users to make the metadata available beyond its original purpose for many other Semantic Web applications.

The first phase of Annotea introduced Web annotations and replies, that formed reply threads, and the second phase, bookmarks and topics. All of these concepts are commonly used familiar metaphors that are general enough to suit for various purposes. As a result, normal users can easily create RDF metadata that can be merged, queried and mixed with other metadata.

1 Introduction

The original Web supports information sharing and collaboration between wide varieties of users, without requiring them to be computer scientists. The Semantic Web (SW) [5] focuses in providing more semantically exact data for machines and agents so that, as a consequence, the agents can better support users in finding the right information. But first the metadata containing the semantics needs to be generated.

Often the metadata is generated by the users of the SW. While it can be done in a decentralized manner together with other users it can still be tedious. Humans are not at their best in providing or understanding complex, machine readable information. Furthermore, they are seldom motivated to provide the information just to help the Semantic Web. For the Semantic Web applications to succeed they need to bridge the gap between the needs of the human users and the requirements of the SW machines and agents.

With Annotea [12] we wanted to experiment how we could enhance the collaboration in the Web with the help of the Semantic Web technologies [20, 6] that offer flexible tools for sharing the user data and semantics, easy extensibility, and effortless merging and querying of the data. The idea of this Semantic Web Supported Collaboration (SWSC) was to support and enhance users' natural collaboration tasks and habits while examining and demonstrating the possibilities of the Semantic Web [13]. In that way, the metadata generation would not feel like an extra effort.

As Annotea tools were targeted for normal users we wanted to use familiar metaphors to support the collaboration. We also wanted users to be able to create metadata as an integral part of the tasks that they were already motivated in performing instead of explicitly creating metadata for the Web. The created metadata is gathered into Annotea objects: annotations, replies, bookmarks, and topics. These concepts were created in two phases.

During the first development phase we focused on examining how to help users, especially users in W3C working groups, to review and discuss the Web documents in the document context in addition to discussion lists. We developed Web annotations and replies that could be used for sharing comments, questions, discussion threads etc. on the context of the Web documents or other Web resources including the annotations and the replies themselves. The basic Annotea architecture was developed during this phase [12]. An important part of the architecture is the ability to store and retrieve the metadata from several annotation servers.

During the second phase we concentrated in enhancing organizing and grouping. The Web users typically grouped resources by organizing links to them under HTML text headings. They could not easily share and reuse categories and other semantics attached to the resources. Annotea shared bookmarks and topics [15, 16] were developed to support the sharing of links to interesting Web documents or other resources and the sharing of link categories. Annotea design allows the users to use simple topics that they find natural even though the topics could be informal and very subjective, especially at the beginning of new work. When the users' understanding about the domain gradually evolves, the organization of the topics can evolve also. When the user learns about similar concepts in more standard ontologies and feels comfortable in using them she can create links that tie her own topics to these concepts.

The SW technologies help Annotea to fulfill the users' needs in many ways. The use of standard SW metadata makes it easy to share annotations, bookmarks and topics with other users, share bookmarks and topics between different browsers, and query and present the annotation, bookmark and topic data in various views. Furthermore, with the SW technologies the users get additional benefits: the metadata can be easily combined with others users' metadata and it is also ready to be used by many other applications.

In the following sections, we will describe in more detail the basic Annotea components, the Annotea metaphors and discuss how the users can benefit from the metaphors and the generated metadata.

2 Basic Annotea Components

Fig.1 presents the basic Annotea architecture. We have various RDF metadata stores storing Annotea objects, a user interface providing different views to the objects in the context of the Web documents or other Web resources, and users collaborating via these objects.

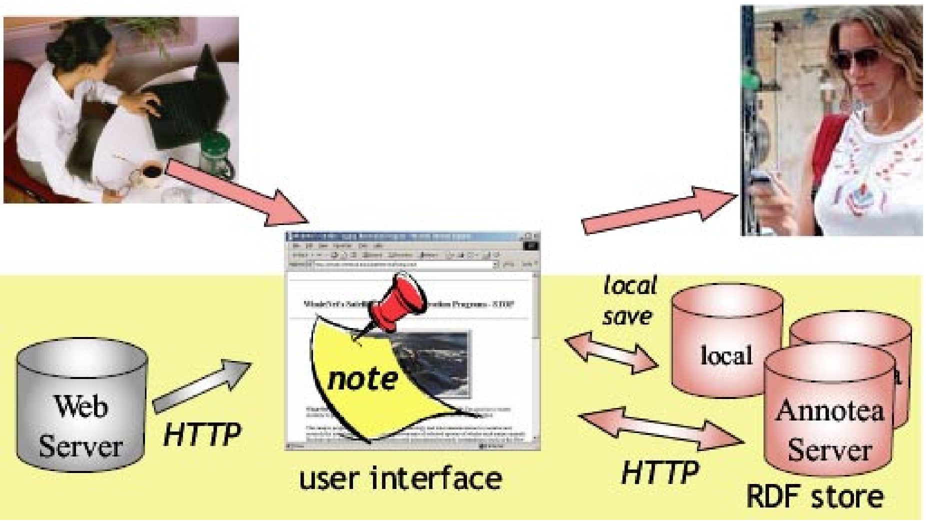


Fig. 1. The basic Annotea architecture.

2.1 Annotea objects

Annotea metaphors encourage users to create Annotea objects, such as annotations, replies, bookmarks, and topics. These are Web resources that have a URI, contain some RDF metadata, and normally include a property referring to some other Web resource. For instance, annotations have an **"annotates"** property for annotating a resource, such as a Web page or even another annotation. Bookmarks [14] have a "**bookmarks"** property for bookmarking Web pages or other resources such as annotations. In addition, the Annotea objects typically include a small set of core properties, such as a Dublin Core "**description"** for the description of the bookmark or a "**creator"** for the creator of the object. Other properties can be added if so wished.

2.2 Web browser user interface

The content of the Annotea objects can be presented in any Web browser user interface as XML text. However, to be usable for any user the normal Web browser needs to support Annotea metaphors. Currently, Annotations are supported in several browsers but they don't always have the same user interface or functionality. We developed the annotations and replies originally into Amaya [1] while Annozilla [3] provides a good implementation of annotations in Mozilla. We started also the bookmarks development with Amaya but currently the main Annotea shared bookmarks development is done in Firefox/Mozilla as Annotea Ubimarks [2]. In the future we wish to collaborate more with Annozilla development so that bookmarks and annotations offer a seamless user experience.

Some tools use Annotea objects but have extended them for their purposes. For instance, FilmEd [9] added time codes to be able to annotate films. Other browsers or tools that are not knowledgeable of the extensions may not be able to present the objects or parts of the objects. While the SW technologies help to make Annotea easily extendible the Web standards and browsers need to catch up in providing easier means for presenting these extensions to users.

2.3 Annotea metadata stores

Annotea objects metadata can be stored either locally, in Annotea servers or as published collections of Annotea objects in Web documents. The user can select from which of these metadata stores she wants to retrieve the Annotea objects. Similarly she can select a store for writing the created Annotea objects.

For historical reasons the current implementations use the Annotea servers for storing annotations and publish the bookmark or topic collections as Web documents. We started by providing servers for annotations but noticed that it is better to allow users to start without first figuring out how to install a server. In addition, a Web document containing annotations can be useful for archiving purposes because a version of the document can be saved with the related annotations for that document in the current review cycle.

Our future goal is to make this difference disappear and use Annotea servers also for storing bookmarks and similarly use the Web documents for storing collections of annotations.

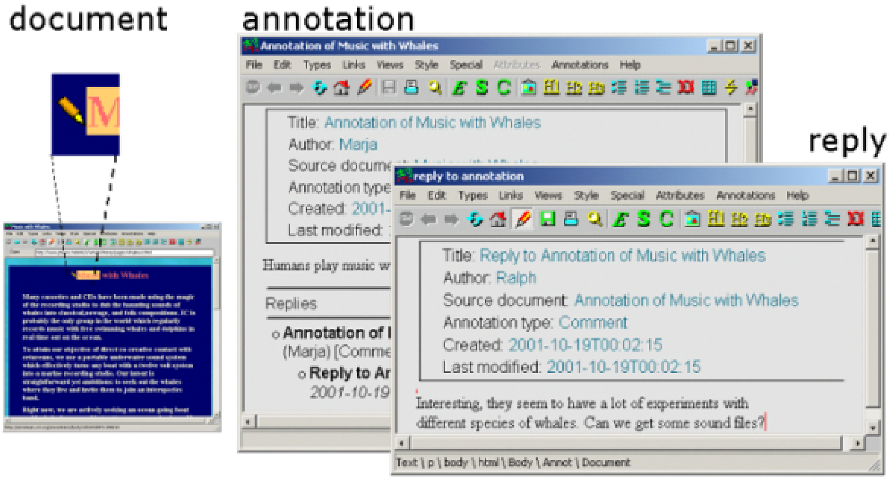
3 Web Browser User Interface and Annotea Metaphors

Annotea uses several familiar metaphors to help users to attach the Semantic Web information to the Web resources as Annotea objects. All the metaphors were developed primarily to help solve user problems and secondarily to create metadata for the Web. We believe that this approach lead us to use more simple objects that can still be extremely beneficial, especially if the simplicity helps larger groups of users to provide the metadata.

This chapter first explains annotations and replies, then bookmarks and topics and finally discusses ways to mix and extend the metaphors.

3.1 Annotations and replies

After looking the W3C standard review process for a couple of months at the beginning of 1998 the author generated a couple of user scenarios where working group members and editors could see review comments as annotations in the context of the reviewed Web documents. The development of Annotea annotations started from those scenarios and targeted specifically to help collaboration between groups of reviewers or other similar users while allowing a user to belong to several of these groups.


document
annotation
reply
Annotation of Music with Whales
File Edit Types Links Views Style Special Attributes Annotations Help
reply to annotation
File Edit Types Links Views Style Special Attributes Annotations Help

Title: Annotation of Music with Whales
Author: Marja
Source docume
Annotation type
Created: 2001-
Last modified:

Humans play music w

Replies
- Annotation of I
(Marja) [Comme
  - Reply to Ar
  2001-10-19

Title: Reply to Annotation of Music with Whales
Author: Ralph
Source document: Annotation of Music with Whales
Annotation type: Comment
Created: 2001-10-19T00:02:15
Last modified: 2001-10-19T00:02:15

Interesting, they seem to have a lot of experiments with different species of whales. Can we get some sound files?

Text \ p \ body \ html \ Body \ Annot \ Document

Fig. 2. Amaya interface to annotations and replies.

Unlike centralized annotation services, such as ThirdVoice [25], or even early versions of Mosaic [21] which by default offered one annotation server, Annotea was focusing in providing a mechanism where every Web user could customize the view to the annotations according to the collaborative groups they belonged to. The Annotea users could control whether they wanted to see annotations and select which servers they wanted to subscribe to retrieve the annotation metadata from. In addition, Annotea provided means to do more detailed filtering of the annotations, for instance, to selectively show the annotations created by a certain user.

Semantic Web technologies were used for implementing Annotea annotations for two reasons. They suited well for attaching information to resources and they helped develop SW technologies as part of W3C Live And Early Adoption (LEAD) philosophy [4].

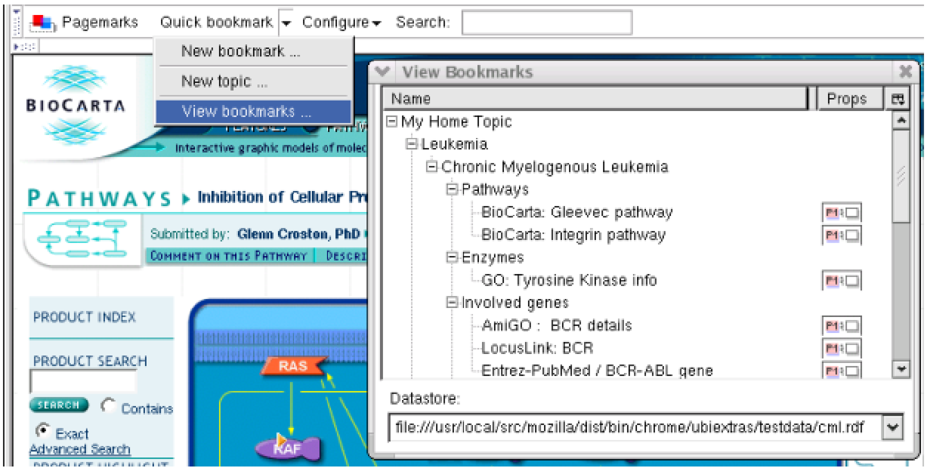
Semantic Web made it easy to offer some control and flexibility for the users. For instance, we provided a default set of annotation types but also demonstrated how the groups themselves could define their own types if they so wished [17]. When users wanted to reply to annotations we easily extended Annotea to add reply objects with thread views as well.

The hardest part in this approach was to get the user interfaces implemented. Annotation content itself was relatively easy to present by using Web technologies, but making it as an integrated part of a browser and user experience was much harder. At the first phase we concentrated in co-operating with Amaya [1] browser developers to show a sample browser implementation. Several implementations were done to other browsers by other developers based on Amaya but with somewhat different functionalities and user interfaces.

Annotea annotations are implemented in browsers but they can also be implemented as part of other tools. For instance, the developers of the SWOOP ontology editor [24] found that the users were often confused of how to properly use the concepts provided by ontology. They added Annotea annotations as an integral part of the tool to make it easy to annotate the ontologies with clarifying explanations and to point out potential problems.

3.2 Bookmarks and topics

During the first phase many informal discussions were performed with users and additional user scenarios were developed. The need for categories was high on the wish list, especially status categories were needed to mark the processed annotations. During the second phase, we selected to focus our scarce resources to broaden the scope and add new Annotea objects, bookmarks and topics. This would help us to make sure that our approach was extensible and the metaphors could work together. In addition, this approach gave us a chance to experiment with some other ideas before going back and improving the annotation implementations.

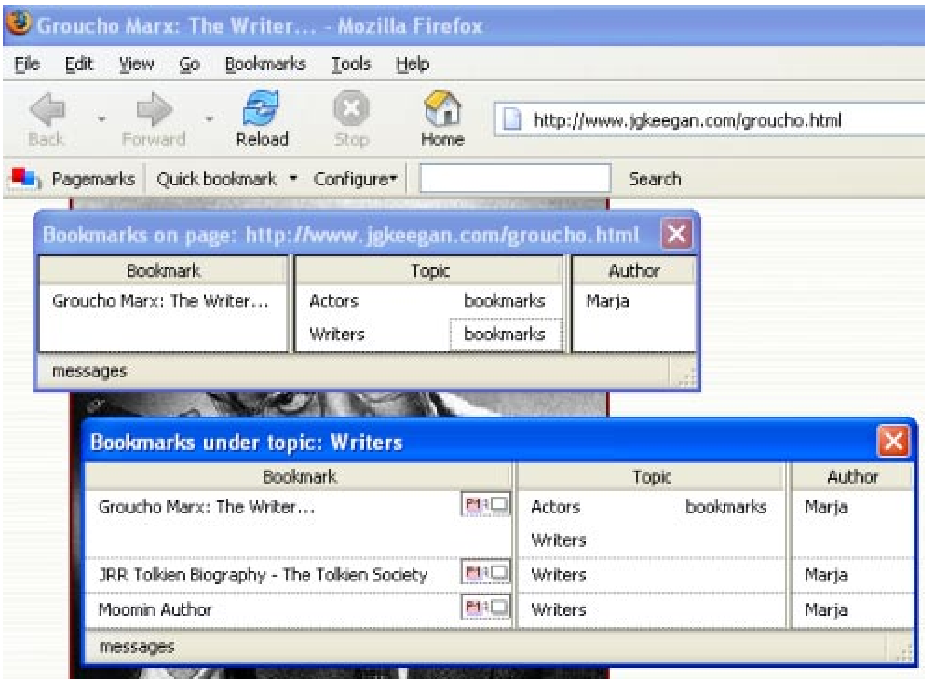


Datastore: file:///usr/local/src/mozilla/dist/bin/chrome/ubiextras/testdata/cml.rdf

Fig. 3. Bookmark and topic hierarchy.

The shared bookmark metaphor with topics was perfect for Annotea as shared bookmarks could easily be seen as a variation of annotations. Furthermore, most users were not only familiar with bookmarks but had actually used traditional bookmark implementations. In addition, traditional browser bookmark user interfaces have a lot of enhancement possibilities that can benefit from the SW approach. For instance, the user interface can utilize the document context to remind the user that she has previously bookmarked the page or let the user define, share and link to other users' topics or categories. Furthermore, many other applications can utilize the bookmark metadata if it becomes widely available.

Annotea topics allow users to create and maintain shared classifications or informal categories [18]. A bookmark can be cataloged under one or more topics and presented to the user in a topic hierarchy (see Fig. 3.). When a user browses pages she can see that someone has bookmarked a page from the pagemark icon that opens up to show a list of bookmarks.



Bookmarks on page: http://www.jgkeegan.com/groucho.html

Bookmark	Topic		Author
Groucho Marx: The Writer...	Actors	bookmarks	Marja
	Writers	bookmarks	

messages

Bookmarks under topic: Writers

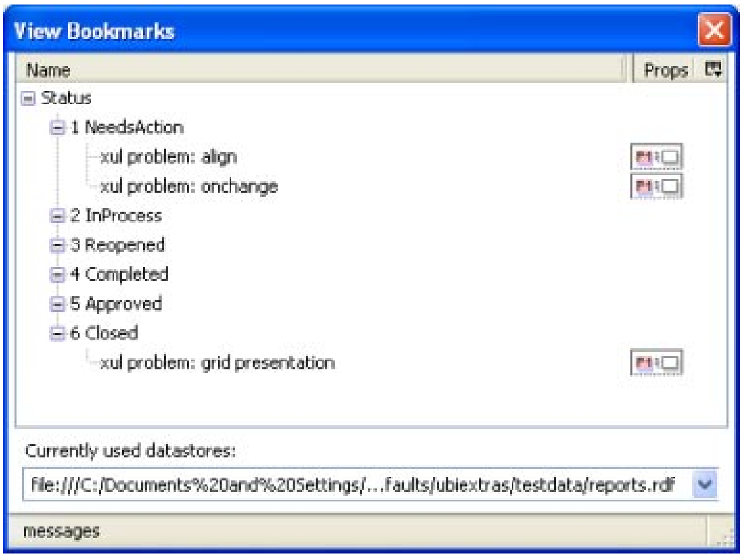
Bookmark	Topic		Author
Groucho Marx: The Writer...	Actors	bookmarks	Marja
	Writers		
JRR Tolkien Biography - The Tolkien Society	Writers		Marja
Moomin Author	Writers		Marja

messages

Fig. 4. Following the link chain from bookmarks on this page to related topics and bookmarks.

The Annotea topics can support early phases of innovations and research by letting the topics to be as vague as needed at the beginning and let the user to refine them as more learning happens or link them to concepts in well established ontologies when those are discovered and understood. Users collaborating in similar or related research areas can benefit from this information. They can see bookmarks on a current page and find related topics and other bookmarks to possibly interesting documents under these topics. For instance, in Fig. 4. the user sees that the page about Groucho Marx has been bookmarked by looking the “Pagemarks” icon on the left side of the toolbar, opens the Bookmarks on page window sees two topics “Actors” and “Writers”, and the bookmarks related to “Writers” to find other “Writers”.

The Annotea topics can easily define concepts outside the conventional categories, such as status (see Fig. 5.). If the topics are separated from the bookmark stores the user can define which aspects she is interested in at each moment by subscribing those topic hierarchies. Similarly, the presented bookmark stores can be selected in a certain domain area or by the research group depending on how they are organized. There are many possibilities for enhancing this user interface.



- Status
  - 1 NeedsAction
    - xul problem: align
    - xul problem: onchange
  - 2 InProcess
  - 3 Reopened
  - 4 Completed
  - 5 Approved
  - 6 Closed
    - xul problem: grid presentation

Currently used datastores:

file:///C:/Documents%20and%20Settings/...faults/ubiextras/testdata/reports.rdf

messages

Fig. 5. Using topics for attaching status values for bookmarked XUL problems.

3.3 Mixing metaphors

The Annotea objects, especially annotations and bookmarks, do not differ very much from each other and can be easily thought as variations of the same class. However, we have also had long and intense discussions with users seeing annotations and bookmarks as a totally different concept and explaining that it would be confusing if bookmarks were annotations. On the other hand we have users who want to immediately extend the bookmarks so that they can refer not only to a Web resource with a URI but also to a part of a document in a similar way as annotations.

The SW metadata of the objects is easy to extend in different ways but designing the user interface to be both simple and expressive enough is critical. Our current view is to keep things simple and see what happens when users start using these interfaces more. Our hope is to be able to experiment with Annozilla and Ubimarks. Maybe the users can create annotations to point to parts of the document and then bookmark that annotation to give it a category and make it easy to find. This would match nicely the way the annotations and bookmarks are combined in Fig. 6 examining the real world usage of annotations and bookmarks.

3.4. Extending Annotea objects

New properties can be easily added to Annotea objects by using the SW technologies, however, SW does not yet offer good standard presentation and interaction definitions. Furthermore, if we want to support normal users to be able to do the extensions much more research is needed. Even if the users would select from a list of predefined properties, or define a similar property to an already existing one, they would still need to do some learning of the properties.

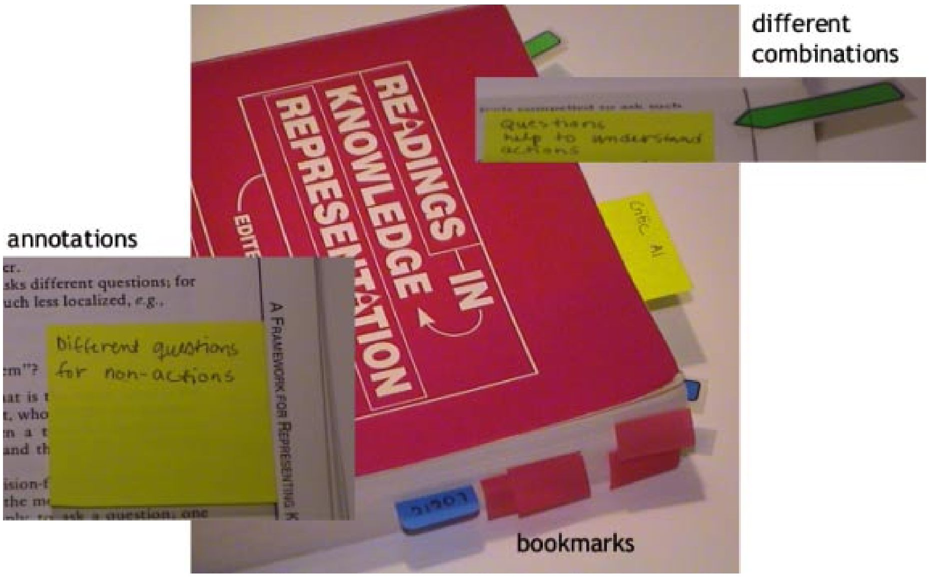

different combinations
Questions help to understand actions
annotations
Different questions for non-actions
READINGS IN KNOWLEDGE REPRESENTATION
A FRAMEWORK FOR REPRESENTING K
bookmarks

Fig. 6. Bookmarks and annotation concepts mix in the real world.

Ontology editors and browsers, such as Protege [22], make the definition of new classes, properties, instances and presentations for them simple. However, the users of these kinds of editors need to have an understanding of classes and properties. While new Annotea objects or subclasses of them could be defined with such an editor, a person with some familiarity with the basic ontology concepts and as well as requirements for the Annotea objects is needed. With ontologies, it is often easy to define and standardize the user interface appearance as the set of possible properties are predefined.

A definition language is needed for presenting the added properties in different views. Currently, the properties not known to the developer beforehand are often presented in an unordered list with property name, string value or a link. Another approach is to show only the known properties and there are use cases for both.

Another aspect we want to be able to extend is the addressing mechanisms and some of the projects using Annotea have already implemented extensions. For instance, we want to be able to use SVG for defining piece of an image that is annotated or time codes to define part of a video. We also want to experiment with different context information. While the metadata extensions need some design, they are easy to accomplish with the underlying SW metadata. However, the user interface is problematic.

We need an extension mechanism for adding user interface definitions related to the metadata extensions for already existing Annotea implementations. So far, we have experimented a little bit with IsaViz [11] by adding simple presentation rules for presenting the properties and their order. In the future, we hope to be able to add piece of presentation code as part of Annotea object extension to make the user interface extensions simpler. The user interface extension could be a combination of presentation rules and scripting, for instance using definitions similar to XUL [26] or XForms [8].

In most cases, we don't expect normal users to write the Annotea object extensions but when new extensions are provided by experts they should be able to easily use them and understand them. Here, the direct benefit from SW technologies is for the developers. When it is easy to add new functionality, test them and change them according to user feedback, it is more probable that users will get what they need.

4 Sample scenarios benefiting from Annotea metadata

Annotation and bookmark metaphors make it easy for users to do what they are already familiar with. Annotations and bookmarks also solve user problems related to collaboration which motivates their creation. As a result from using Annotea objects for their own needs the users create SW metadata that can be easily reused by various other applications. Here are couple of examples of such applications.

Spam annotations support collaborative spam filtering:
Spam messages in discussion lists can be annotated by trusted users and the messages filtered away while not loosing any information from the archives. This is used at W3C discussion lists [19].

Bookmark and topic collections can be used as user controlled profiles:
Users can use parts of their bookmark collections as user profiles when they visit services, for instance, a user visiting Amazon.com can ask similar books to the ones he has bookmarked on the Web in addition or in place of the information the service already has gathered of the user.

Using bookmarks and topics for finding and categorizing related information:
Data mining techniques can be used to find related resources by using connection paths provided by Annotea topic objects. This works even when the user is not subscribing the data stores and following the links like in Fig. 4. The topic objects can also help in naming the automatically found clusters of information in user understandable ways.

Annotea bookmarks and topics objects could be used to enhance tools providing automatic collaborative browsing, such as Magpie [7]. For instance, the tool could provide links to related projects both automatically and by utilizing user generated bookmarks. With the help of topic object data these projects can be presented in category hierarchies initiated from users' own understanding.

Organizing search engine results:
Users can use bookmark or annotation collections with search engines to organize the search results. For instance, when using Google the bookmarks of an expert user group in a searched domain can be used to first show the resources in that domain

[23]. Alternatively given topics and their related topics found from the Web can be used to organize the results.

Easy feedback channel for normal users:
Annotations and bookmarks can be used as a feedback channel for many services and they can also be integrated with other SW applications. For instance users could bookmark the resources in applications, such as Museum of Finland [10], by using Annotea topic objects. If the application provides a bookmark server, the data from that can be used to analyze and further develop the used ontologies.

5 Conclusions

SW technologies can support users directly by helping them to generate reusable standard metadata or indirectly by helping the developers provide different views to the data. If we want a wide variety of users to contribute to the SW by providing metadata and benefit from the metadata they need both motivation in the form of helping them in their tasks and good metaphors that hide or make the technology understandable. With Annotea we used SWSC starting from analyzing users' collaboration needs for motivation and Annotea metaphors for making the necessary SW technologies understandable.

Annotea metaphors successfully hide the underlying SW technologies from the users so that they can use SW fluently without even knowing about it. Users do need to know how to subscribe the data stores containing the various Annotea objects. Stores can be local files, global servers and or Web documents containing the metadata. Web documents offer users an easy alternative to get started without investing in installation of a server. They can also be used to archive snapshots of the selected Annotea objects outside the server.

SW offers the developers an easy and flexible interface for merging metadata from several different sources and doing queries against it. Different views to the data can be created easily, and it is easy to let the users follow tracks of data from the information on the current page to possibly related information. Extending the SW data in Annotea objects is easy as well. Defining a simple user interface for extensions is relatively easy, but adding more complex user interface definitions to the extensions needs more research.

The biggest direct benefit from the use of SW technologies and metadata is that the user generated metadata can be easily combined and reused in many other applications, such as user profiles for services, data mining and search engine applications.

Acknowledgements

Many people have contributed to Annotea in one way or other during its development and helped to make it what it is today. I want to give special thanks to Jose Kahan, Eric Prud'hommeaux and Ralph Swick.

References

[1] Amaya home page, http://www.w3.org/Amaya
[2] Annotea Ubimarks homepage, http://www.annotea.org/mozilla/ubi.html
[3] Annozilla home page, http://annozilla.mozdev.org/
[4] Berners-Lee, T. Web Architecture from 50,000 feet, http://www.w3.org/DesignIssues/Architecture.html#Collaboration
[5] Berners-Lee, T., Hendler, J. and Lassila, O. The Semantic Web: A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities, Scientific American, May 2001.
[6] Brickley, D., and Guha R.V. (eds.). RDF Vocabulary Description Language 1.0: RDF Schema, W3C Recommendation 10 February 2004. http://www.w3.org/TR/2004/REC-rdf-schema-20040210/
[7] Domingue, J., Dzbor, M., and Motta, E. Collaborative Semantic Web Browsing with Magpie,In Proc. of the 1st European Semantic Web Symposium (ESWS), May 2004, Greece, http://kmi.open.ac.uk/people/dzbor/public/2004/ESWS-domingue-dzbor-motta-final.pdf
[8] Dubinko, M., et. al. XForms 1.0, W3C Recommendation 14 October 2003, http://www.w3.org/TR/xforms/
[9] FilmEd homepage, http://metadata.net/filmed/
[10] Hyvönen, E. et. al , Finnish Museums on the Semantic Web: The user's Perspective on MuseumFinland, In Proc. of the Museums and the Web 2004 Conference, http://www.archimuse.com/mw2004/papers/hyvonen/hyvonen.html
[11] IsaViz homepage http://www.w3.org/2001/11/IsaViz/
[12] Kahan, J., Koivunen, M., Prud'Hommeaux, E., and Swick, R. Annotea: An Open RDF Infrastructure for Shared Web Annotations, in Proc. of the WWW10 International Conference, Hong Kong, May 2001 http://www10.org/cdrom/papers/488/index.html
[13] Koivunen, M. and Swick, R. Metadata Based Annotation Infrastructure offers Flexibility and Extensibility for Collaborative Applications and Beyond, In Proc. of the KCAP 2001 Conference, http://www.w3.org/2001/Annotea/Papers/KCAP01/annotea.html
[14] Koivunen, M., Swick, R., Kahan, J., Prud'hommeaux, E., An Annotea Bookmark Schema, 2003, http://www.w3.org/2003/07/Annotea/BookmarkSchema-20030707
[15] Koivunen, M., Swick, R., and Prud'hommeaux, E. Annotea Shared Bookmarks, 2003, In Proc. of KCAP 2003, http://www.w3.org/2001/Annotea/Papers/KCAP03/annoteabm.html
[16] Koivunen, M., Annotea shared bookmarks: Semantic Web at your fingertips, In Proc. of the ISWC 2004 Conference Demonstrations Session. http://www.annotea.org/ISWC2004/annoteademo.html
[17] Koivunen, M. Defining New Annotation Types in Amaya, January 2004, http://www.w3.org/2001/Annotea/User/Types.html
[18] Koivunen, M. Scenario: Organizing CML cancer research knowledge by using Annotea shared bookmarks, 2003, http://www.w3.org/2003/12/cmlcase/cml.html
[19] Koivunen, M., AnnoSpam: Filtering Spam According to Annotations, demo slides, September 2003

[20] Lassila, O., and Swick, R. (eds.), Resource Description Framework (RDF) Model and Syntax Specification, W3C Recommendation, 22 February 1999 http://www.w3.org/TR/1999/REC-rdf-syntax-19990222.

[21] NCSA Mosaic: Annotations Overview, 1997, http://archive.ncsa.uiuc.edu/SDG/Software/XMosaic/Annotations/overview.html

[22] Protege home page, http://protege.stanford.edu/

[23] Shiraishi, N. (2004) The RDF Trust Model Using RDF Bookmark and it's Application. In Proc. of WWW2004 Workshop on Content Labeling -Technical and Socio-Cultural Challenges and Solutions. http://web.sfc.keio.ac.jp/~kaz/www2004/papers/ns.pdf

[24] SWOOP Ontology Editor home page, http://www.mindswap.org/2004/SWOOP/

[25] ThirdVoice, 1998, http://c2.com/cgi/wiki?ThirdVoice

[26] XML User Interface Language (XUL) home page, http://www.mozilla.org/projects/xul/

MoRe Semantic Web Applications

Maksym Korotkiy$^{\alpha}$ and Jan L. Top$^{\alpha\beta}$

$^{\alpha}$Vrije Universiteit Amsterdam, Department of Computer Science
De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands
$^{\beta}$Wageningen Centre for Food Sciences
P.O. Box 557 6700 AN Wageningen, The Netherlands
maksym@few.vu.nl jltop@few.vu.nl

Abstract. We present *MoRe* – a framework that allows one to extend a domain ontology with a remotely invocable reasoning service applicable to concepts defined in that ontology. Our approach bridges the gap between ontology and application developers. We allow any reasoning service to be wrapped by a *MoRe* ontology extension, the services ranging from generic logics-based reasoners to specific black box software components. The application developer directly accesses these reasoning services through documents stated in terms of domain concepts rather than dealing with remote procedure calls. We describe a case that applies *MoRe* to an OWL-ontology of units of measure, and we demonstrate how this extended ontology can be integrated into a unit conversion application.

1 Introduction

Ultimately, the Semantic Web [1] aims to significantly improve the experience of web application end-users. To achieve this, the Semantic Web is to provide an environment that enables an application developer to advance web applications beyond what we observe nowadays. Ontologies are the keystones of the Semantic Web, and ontology developers play a crucial role in developing that enabling environment.

It is believed that the current approaches to the Semantic Web make it rather difficult to develop applications needed so much to materialize the Semantic Web vision [2, 3]. A number of solutions to facilitate design of Semantic Web applications has been proposed, ranging from authoring [4], browsing and annotation frameworks [5, 6] to infrastructures for the Semantic Web Services [7]. These approaches address specific aspects of the Semantic Web application development. In *MoRe* we take a step back to see how ontologies can be extended to make them more attractive for application development in general.

Presently, application developers do not profit much from the increasing availability of domain ontologies. The latter are typically devised for representation of static domain knowledge, whereas applications require problem-specific answers and computations. Generic reasoners and query languages associated with formalisms like RDFS or OWL are often not expressive, efficient and transparent enough to be used in applications. We propose *MoRe* [1] – a simple approach to extend ontologies with application-oriented, but

[1] “MoRe” used to be an acronym but with the development of our approach its original interpretation has become irrelevant.

still generic concepts and reasoning services. This provides for solutions, arbitrarily on the continuous scale between domain- specific ontologies (using generic reasoners) to task-specific applications (using dedicated procedures). The objective of our approach is to simplify application development by means of increased (re)usability of ontologies.

To improve the usability of ontologies *MoRe* extends them by providing an elementary mechanism for attaching a reasoning service to these ontologies. We believe that the availability of a readily accessible reasoning service allows the application developer to faster evaluate an ontology and to incorporate it more readily into an application.

Our approach can be illustrated by providing an example from e-Science, our field of application. e-Science aims at providing automated support to researchers in performing experiments and constructing models and theories. A simple but very important quality requirement for scientific work is correct and consistent use of units of measure. Traditionally, an application developer would construct a specific algorithm for unit conversion, using an internal database of units and their values in terms of reference units. With the availability of a units ontology, the application developer could instead access this ontology to determine for example the conversion factor between two units. In this case, he would profit from the shared knowledge provided by the ontology, but he would still have to write specific code to employ this knowledge in the application.

In *MoRe* we extend the units ontology with an associated reasoning service. For unit conversion, we define an additional but still generic concept `ConversionExpression`. In this case, the application developer only needs to specify a document with the following content (simplified):

```
UnitsOntology
    ConversionExpression
       sourceUnit:      inch
       destinationUnit: yard
```

The appropriate middleware detects the ontology underlying the document, locates it on the Web and applies the associated reasoning service to this input document. After that the middleware sends the following document back to the application:

```
UnitsOntology
    ConversionExpression
       sourceUnit:      inch
       destinationUnit: yard
       factor:          0.02777778
```

The resulting document contains a new fact (value of the `factor` property) allowing the developer to convert inches to yards.

This example provides a simple illustration of the application of our approach. The main motivation behind *MoRe* is to make ontologies more (re-)usable to application developers. Moreover, *MoRe* can help to overcome the lack of expressiveness, efficiency and transparency of present ontology languages (such as OWL) and associated generic reasoners. We emphasize that we do not claim to replace or improve existing formalisms, but rather to provide a pragmatic framework for applying more or less specific algorithms were needed.

In this paper we first introduce the “Unit Converter” scenario in Section 2 in which we outline the main steps a developer takes to employ an ontology in the application at hand. Then in Section 3 we outline the main ideas behind *MoRe* and in Section 4 we apply our approach to the “Unit Converter” scenario. After that, in Section 5 we discuss relationships between *MoRe* and present approaches to ontologies and Semantic Web Services. Also we elaborate on how both the application and the ontology developers are effected by *MoRe* and how they benefit from it. Finally, we conclude with Section 6.

2 The “Unit Converter” Scenario

To demonstrate the effect of *MoRe* on application and ontology developers we employ the “Unit Converter” scenario. In the scenario we consider a task of developing an ontology-based unit conversion application – Unit Converter – that assists a user with conversion between different units of measure.

To develop the Unit Converter, an application developer begins with the application domain analysis. As a part of the analysis, the developer searches for existing ontologies covering the target domain. Let us assume that the developer has found an ontology of units of measure describing the application domain.

The ontology of units of measure can for example be utilized to organize the unit space in a way familiar to the end-user, to select subsets of units that can be converted to each other and, finally, to determine a conversion expression between two given units of measure. In this scenario we elaborate on the last application of the ontology.

Let us assume that the ontology describes a number of units of measure (yard, inch etc) and a conversion factor between a unit and a corresponding reference unit. For example, the ontology states that yard unit has the `SI unit factor` property with value 0.9144 and for inch unit the value is 0.0254. In this example, the `SI unit-` part of the property refers to meter unit (meter is the standard SI-unit for length), so the previous sentence means that 1 yard = 0.9144 meter and 1 inch = 0.0254 meter. An application developer can use the two property values to compute a conversion factor between yard and inch: 1 yard = 0.9144 / 0.0254 inch.

At present we see two main styles of employing ontologies into applications. The first approach is to extract relevant information from an ontology in application-specific form (most often a database) and then employ traditional techniques to access the data and to apply application logic to them. The pseudocode in Fig. 1 demonstrates distinctive features of such an approach. It includes the use of a data query language and computation of the conversion factor in the application.

The main advantage of this approach is that as soon as relevant data is extracted from the ontology, the application developer is able to apply conventional (and well-known) techniques to access the data. The major drawback is that such an approach degrades an ontology to the level of data and makes it difficult to use domain knowledge captured by it.

```
computeConversionFactor (srcUnit, dstUnit, factor)

  srcFactor=db.query(‘‘
          SELECT SI_unit_factor
          FROM ...
          WHERE unit=srcUnit’’
      ).get(‘‘SI_unit_factor’’)

  dstFactor=db.query(‘‘...WHERE unit=dstUnit’’).get(...)

  factor= srcFactor/dstFactor
```

Fig. 1. Pseudocode of a traditional DB-based approach. Using general purpose ontology middelware leads to a similar solution.

```
computeConversionFactor (srcUnit, dstUnit, factor)

  reqDoc=
      ‘‘MyConversionExpression
          type          ConversionExpression
          hasSourceUnit srcUnit
          hasDestUnit   srcUnit’’

  resDoc=MoRe.process(reqDoc)

  factor = resDoc.getProperty(‘‘hasConversionFactor’’)
```

Fig. 2. Pseudocode of a *MoRe*-based approach.

The second way to exploit an ontology in an application is to employ general purpose ontology middleware, such as Jena [8] or Sesame [9], that provides storage, reasoning and query facilities for ontologies.

However, in our scenario the reasoning capabilities of the associated ontology languages do not allow us to compute the conversion factors in a feasible way (we elaborate on this in Section 4). As a consequence, the second approach would be very similar to the previous one, only the queries would be expressed in a different language and applied not to a database but to a stored ontology.

In both cases, the application developer has to incorporate part of the domain knowledge into the application. Nevertheless, it is natural to expect that the way a conversion factor is computed is part of the units of measure domain. *MoRe* allows an ontology developer to incorporate such domain knowledge as a domain-specific reasoning procedure connected to concepts from the units of measure domain. If an application developer would have such a *MoRe*-ontology to his disposal, the pseudocode depicted in Fig. 2 could be used.

The major difference with the previous cases is that the application developer now reuses domain knowledge about the conversion factor via the reasoning service provided by the *MoRe*-ontology. The second distinction is that the application developer does not need an additional query language to utilize the domain knowledge captured in the ontology. He only needs to refer to an instance of `ConversionExpression` from the extended units ontology. We will elaborate on this in Section 4 and now we introduce the main ideas behind *MoRe* in the following section.

3 *MoRe* in a Nutshell: Documents, Ontologies and Handlers

This section provides a compressed description of the *MoRe* framework. Section 4 fills in missing details and describes how the proposed approach is applied to the introduced scenario.

In *MoRe* we use the notion of `Document` to provide a unified view of Semantic Web resources. All documents share the same structure (a collection of object-property-value triples) and syntax (XML-RDF). A document describes a particular situation in a domain and explicitly refers to exactly one *MoRe*-ontology. This ontology extension defines a reasoning service applicable within that domain.

A *MoRe*-ontology (Fig. 3) extends conventional ontologies by serving as a reasoning service provider. To achieve this, a *MoRe*-ontology exposes exactly one handler providing an entry point to a reasoning service. In *MoRe* we assume that the handler can be invoked via HTTP POST-request with one document as an attachment. The handler processes the attachment and delivers another document as its output. Essentially, a handler is a black box and a *MoRe*-ontology provides the information sufficient for its invocation: the handler's URL. A *MoRe*-ontology can reuse reasoning services provided by other *MoRe*-ontologies. This aspect will not be discussed further in this paper.

Conceptually, in *MoRe* we make use of a subset of RDFS (`Class`, `Property`, `subClassOf`, `subPropertyOf`, `type` and `label`) extended with concepts representing main *MoRe* concepts (`MoReOntology`, `Document`, `Handler`, `URL`) and relationships between them.

Having obtained a document, *MoRe*-middelware is able to identify the extended ontology underlying it and to apply the reasoning service described in this ontology. The outcome of the reasoning service depends on domain-specific knowledge captured in the ontology and the input document that represents a particular situation in the domain. The ontology also provides terminology for input and output documents.

4 Applying *MoRe*

In this section we elaborate on the use scenario presented earlier and describe in detail how *MoRe*-ontologies can be developed to support development of Semantic Web applications. We also show how an application should be designed to utilize extended ontologies.

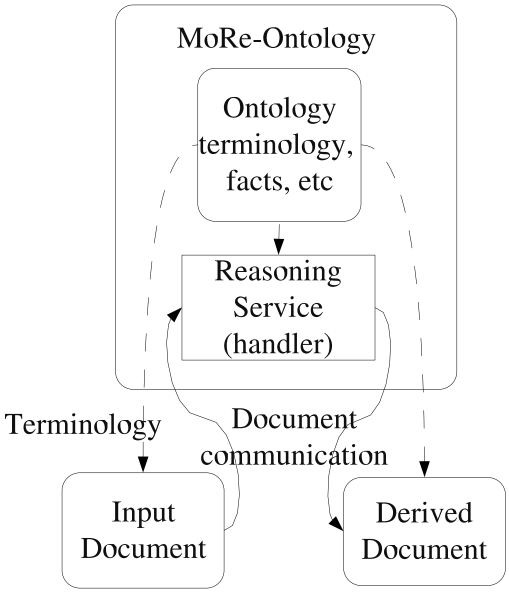


Fig. 3. Relationship between *MoRe*-ontology, document and handler.

4.1 Designing a *MoRe*-Ontology

We now describe the detailed design of a *MoRe*-ontology that extends the ontology of units of measure. In our case we employ UnitDim [2] for this purpose. One of the problems we faced during our initial attempts to exploit the knowledge captured in UnitDim was the difficulty to access domain knowledge expressed in terms of OWL restrictions. Another reason to extend UnitDim into a *MoRe*-ontology is the inability to compute conversion expressions between units using state-of-the-art general purpose reasoners. We have addressed the former problem by creating the UnitRS *MoRe*-ontology extension and the latter has been addressed by the UnitCS ontology extension. Since the general design steps for both UnitRS and UnitCS are similar, in this paper we only elaborate on the UnitCS ontology.

The fact that we develop a *MoRe*-ontology for an application may seem to contradict the idea that ontologies should be application independent. However, we will see that the developed *MoRe*-ontology does not lose any generality but only gains utility in our approach. Moreover, we do not create a *MoRe*-ontology from scratch, but rather extend UnitDim. This demonstrates how an existing ontology can be made more attractive for application development and still preserve its generality.

Unit Conversion (UnitCS) Ontology UnitDim describes a quantitative relation between every unit and a single reference unit. The relation is called `SI_unit_factor`

[2] Rijgersberg, H., Top, J.: UnitDim: an ontology of physical units and quantities. http://www.atoapps.nl/foodinformatics. Sec. News (2004)

and represents a conversion factor between the unit and its counterpart from the SI System of Units. Two such relations can be combined to determine the conversion factor between any two units. In principle, we could have used a general OWL-reasoners to do so. Unfortunately, the OWL language cannot express this domain knowledge in a feasible way. Given a subset of N units, such that any two units can be converted to each other, in OWL we would have to use the complete enumeration of conversion factors. This would result in N^2 property values instead of the more feasible N `SI unit factor` values combined with a capability to infer the rest.

An application can employ the UnitCS ontology to describe documents which then can be subjected to the reasoning service defined in UnitCS. The UnitCS ontology employs the conceptualization defined in two ontologies:

- *MoRe*-Ontology – defines the *MoRe* framework (concepts and their interpretation). All *MoRe*-ontologies reuse this ontology. The handler defined in the *MoRe*-Ontology implements the core of what we will be referring to as *MoRe*-middleware.
- UnitDim – is a conventional domain ontology expressed in OWL. The UnitCS reasoning service exploits domain knowledge captured in UnitDim about relationships between units of measure.

From the ontological point of view, the UnitCS ontology extends UnitDim to infer a conversion expression between convertible units. To achieve this we introduce the `ConversionExpression` class to represent a ternary relationship between two units and a corresponding conversion factor. The `ConversionExpression` class has three properties:

- `hasSource` – points to a source unit, the unit to which we apply the conversion factor;
- `hasDestionation` – points to the destination unit to which the `hasSource`-unit is to be converted to;
- `hasFactor` – contains the numerical value of the conversion factor.

We design the handler for the UnitCS ontology in such a way that for every instance of `Conversion Expression` contained in the input document, the handler determines `SI unit factors` for both the source and destination unit and combines them to compute the corresponding `hasFactor` value. More precisely:

$$Factor_{srcUnit,dstUnit} = factor_{SIUnit,srcUnit}/factor_{SIUnit,dstUnit}.$$

The computed factor allows us to use the following conversion expression

$$srcUnit = Factor_{srcUnit,dstUnit} \cdot dstUnit.$$

Note that the UnitCS ontology could have contained an OWL reasoner instead of our custom-built handler if it could have provided the required functionality. In that case we could see the UnitRS ontology as a wrapper around UnitDim and the standard OWL reasoning mechanism.

An application exploiting the UnitCS ontology needs to have access to the *MoRe*-middleware in order to utilize the reasoning capabilities of the UnitCS ontology. The application uses the terminology defined in UnitCS to create a document describing an instance of the `ConversionExpression` class (Fig. 6). In addition the document contains a URL that points to the UnitCS ontology.

Having created the input document, the application submits it to the *MoRe*-middleware. The middleware analyzes the document, locates its underlying ontology and invokes the handler of that ontology. All inferred facts (statements) are added to the document, which is then returned to the *MoRe*-middleware. The middleware, in turn, returns the result document to the application. The application updates its state according to the newly obtained information.

The above scenario demonstrates how the *MoRe*-framework allows the application developer to abstract from calls to remote procedures. The essential point is that the developer can stay at the conceptual level of domain terminology when requesting external domain knowledge.

4.2 Building the Unit Converter Application

We have employed the UnitRS and UnitCS ontologies in the Unit Converter [3] tool. Figure 5 depicts the architecture of the Unit Converter. In the figure we can see three distinct layers:

- The layer of traditional ontologies is at the top. The UnitDim ontology is the only traditional ontology employed in our application.
- The application layer forms the bottom layer. Inside this layer we can distinguish two sub-layers that represent the application logic and the user interface (UI). The UI sub-layer accepts user's commands and displays the relevant the application state. In our case the user is able to perform three actions: select source (step 1 in Fig. 4) and destination units, and ask for a conversion expression (step 2 in Fig. 4. Two of the actions are connected to the application logic layer in which there are two components responsible for determining 1) a set of convertible units and 2) a conversion expression between two units.
- The application and ontology layers are connected by the *MoRe*-layer. The UnitRS ontology provides a unit retrieval service which makes it easier for an application developer to access domain knowledge captured in UnitDim. UnitCS extends UnitDim with a new concept (`ConversionExpression`) and provides a reasoning service capturing domain knowledge about this concept.

  The application layer interacts with the *MoRe* layer in two ways. First, it employs ontological terminology defined in *MoRe*-ontologies for interfacing purposes (documents). Second, it communicates with *MoRe*-middleware. The middleware is responsible for delivering the input document to the corresponding ontology (its reasoning service) and communicating the output document back to the application layer.

[3] The Unit Converter is accessible via http://www.cs.vu.nl/∼maksym/MoRe/

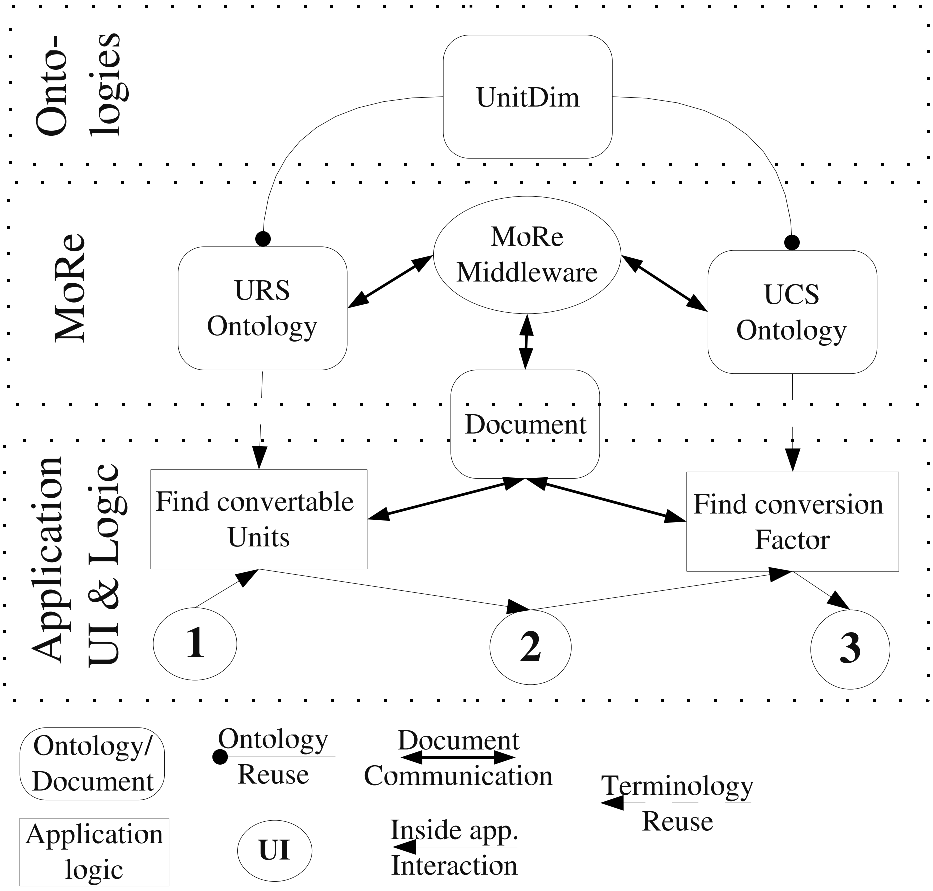
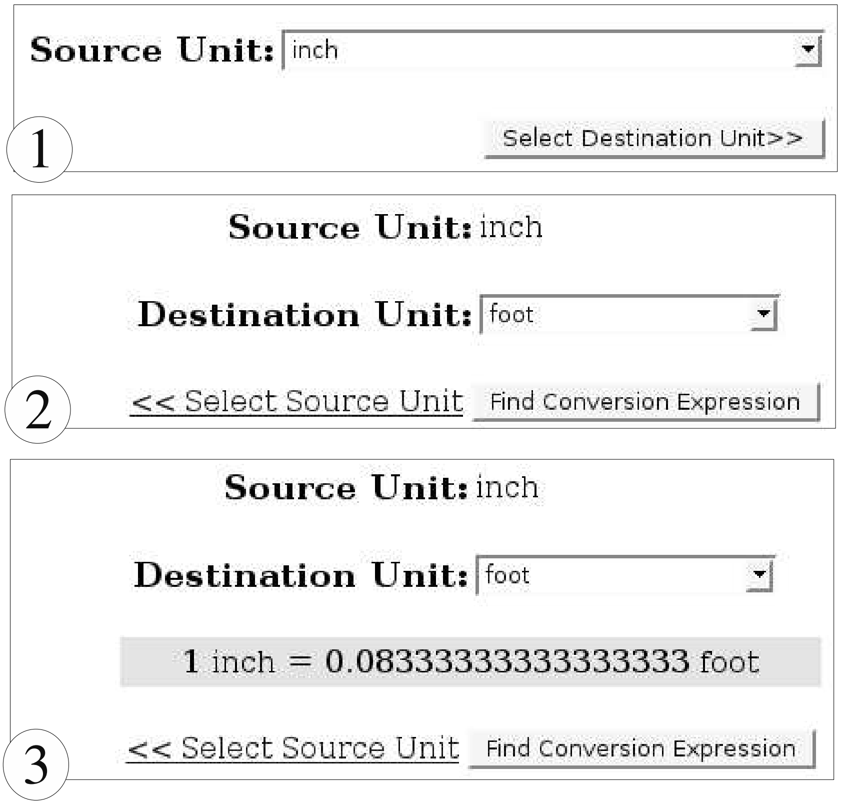


Fig. 4. Main steps of the Unit Converter applicaton


1
Select Destination Unit>>
2
<< Select Source Unit
Find Conversion Expression
3
<< Select Source Unit
Find Conversion Expression

Source Unit: inch

Source Unit: inch

Destination Unit: foot

Source Unit: inch

Destination Unit: foot

1 inch = 0.08333333333333333 foot

Fig. 5. The architecture and the user interface of the "Unit Converter" application. Numbers on the UI screenshots correspond to the UI components in the application layer.

```
  rdf:Description rdf:about=''ceInst0''
*  hasFactor 0.0277776
   hasSource rdf:resource=''inch''
   rdf:type rdf:resource=''ConversionExpression''
   hasDestination rdf:resource=''yard''
  rdf:Description
```

Fig. 6. A simplified example of a document communicated between the application and *MoRe* layers. Initially the document does not contain a line with "*" which is added by a reasoning service.

Fig. 6 contains a fragment of input and output documents communicated between the "Find Conversion Factor" component of the application layer and the *MoRe* layer. The input document does not contain a line marked with "*". The value of the `hasFactor` property is computed by the UnitCS handler and added to the input document. The initial situation reflects the state of the application after the user has selected the source and destination units. The output document contains a new fact (`hasFactor` property value in our case) which is used to update the application state (step 3 in Fig. 5).

5 Discussion

Despite its brevity, Section 3 describes the main ideas underlying the proposed framework. Nevertheless, we believe that the potential impact of *MoRe* on ontology and application developers can be significant. In this section we discuss the effect of *MoRe* on the current approaches to using ontologies in applications. We also clarify the difference between the proposed approach and the usual view on Semantic Web Services.

5.1 Explicating reasoning mechanisms

The proposed framework does not compete with existing approaches to ontology languages but rather complements them by explicating the link between an ontology and an applicable reasoning mechanism. We believe that this can improve the flexibility of ontologies and make it easier to develop ontology-based applications.

In Section 4 we have explained that state-of-the-art ontology languages such as RDFS and OWL cannot determine a conversion expression between units of measure. We believe that this problem is caused not only by lack of expressiveness of the language but that it is rather a manifestation of the inflexibility of these languages. In *MoRe*, the ontology developer can attach an arbitrary reasoning service to an ontology, ranging from general purpose reasoners to dedicated, goal-specific algorithms. This allows one to handle the limitations of present ontology languages. These limitations become visible when designing real-world applications, either in terms of limited expressiveness or of limits in performance.

MoRe enables development of domain ontologies which are easy to apply. We submit that the success of knowledge-intensive ontologies will be determined primarily by their usability in applications and only secondarily by their definition as a general standard.

5.2 The Black-Box Approach to Capturing Domain Knowledge

In *MoRe* we apply the black-box model to represent a reasoning mechanism. Such a non-declarative approach makes it impossible to reuse parts of the knowledge captured within the black-box. We argue, however, that any declarative language also requires its own black-box to interpret language statements.

In MoRe *we leave it to the user to decide where to draw the border between a declarative and non-declarative representation of domain knowledge.* Usually, it is much easier to initially capture knowledge in a procedural way because it does not restrict the user to a particular declarative representation. Later on, some parts of the procedural knowledge can be exposed in a declarative way. In this way, *MoRe* allows an evolutionary transition from procedural to declarative knowledge representation. Moreover, sometimes it is just impossible to express domain knowledge in a declarative way. For example, a neural network can be trained to organize domain objects into categories, in an inherently non-declarative way. Current approaches do not allow the user to benefit from advances in such non-symbolic areas as evolutionary computing, machine learning and neural networks. We believe that in *MoRe* we enable the user to combine declarative representation techniques with computational (AI) methods and we are going to investigate this ability in our future work.

5.3 Software Components

One more benefit of the black-box approach is that any software component can be represented in this way. This provides us with a link between software engineering and ontologies. *MoRe* makes it possible for software component developers and ontology engineers to combine their efforts to create reusable domain ontologies. In many cases a software component can be relatively easily modified to become part of the Semantic Web. The interface of the component must be reformulated to express input arguments as RDF-XML documents. The terminology employed in the interface becomes part of the component ontology and the logic implemented in the component defines an applicable reasoning mechanism. We believe that *MoRe* will allow us to bridge software engineering and ontological design, improving reusability of the former and flexibility of the latter.

Additionally, *MoRe* allows application developers to abstract from the level of explicit calls to remotely invocable procedures to the ontological level, in which documents are created according to ontologies and reasoning is applied to those document transparently to developers and end-users.

5.4 *MoRe* and the Semantic Web Services

The proposed approach is based upon ontologies and reasoning mechanisms readily-available on the Web. We rely on well-established Web standards such as URI, HTTML, XML and RDF to make the proposed framework operational. This results in a superficial similarity between *MoRe* and the existing approaches to Semantic Web Services such as OWL-S and WSMO. A detailed overview of present approaches to the Semantic Web Services is given in [10]. Here we highlight the major differences between *MoRe* and the Semantic Web Services in general and OWL-S in particular.

MoRe *is a general extension to ontologies.* In *MoRe* we provide the user with a general mechanism to attach a reasoning mechanism to a domain conceptualization. It should not be compared with, for example, OWL-S, which provides a conceptualization of the domain called “Semantic Web Services”.

MoRe does not address problems of automatic discovery and composition addressed by other SWS-techniques. Instead *MoRe* provides a simple framework enabling reuse of ontology-based reasoning services. If desirable, additional reasoning services can be plugged in into *MoRe* to address, for example, the automatic discovery task typical for the Semantic Web Services.

MoRe *promotes different usage patterns for ontologies.* In Semantic Web Services, ontologies are used to annotate a Web Service. To use this resource an agent has to understand the annotation, reason about it and, finally, exploit the resource. In *MoRe* an ontology is rather seen as a language that is used to express documents. We believe that the language and the document have a value of their own. An ontology becomes a resource directly exploitable by an agent. *MoRe* enables us to incorporate any computational activity into the reasoning stage transparently to the agent.

At the operational level, OWL-S primarily uses an RPC-style of interaction with the service, focusing on the procedural approach to service specification. This, along with the extensive exposure of internal service details in the process model contrasts with *MoRe*, where we rely on the document-based interaction style with one predefined entry point to the reasoning mechanism. We believe that such an approach is more flexible and offers more opportunities to manage complexity by hiding details of reasoning.

6 Conclusions

We have proposed *MoRe* – a framework for development of ontology-based applications by enabling an explicit link between domain terminology and the appropriate reasoning mechanisms. The proposed framework is based on documents and ontologies containing explicit references to remotely invocable reasoning services (handlers), providing a simple and flexible foundation for development of ontology-based Web applications.

We start with the observation that at some point any ontology requires a directly invocable reasoning mechanism. *MoRe* provides a framework for linking this mechanism, represented as a black-box, to terminology defined in the ontology. This approach not only allows us to incorporate inherently non-declarative reasoning mechanisms, for example based on neural networks, evolutionary computing or any software component into an ontology, but also empowers the user to decide where to draw a border between declarative and procedural representation of domain knowledge.

We believe that *MoRe* helps us to bridge the gap between ontological domain knowledge and application development. On the one hand, it provides a pragmatic application-driven view of extending ontologies. On the other hand, it facilitates application development by enabling easy integration of reasoning services into end-user software solutions. *MoRe* supports an evolutionary development path from existing (legacy) applications to ontology based services.

We are yet to obtain a definitive answer about the practical implications of *MoRe* and a number of technical design decisions. In particular the question on how to combine several ontologies with their respective reasoning services is to be answered. Our next step will be to further validate and refine the method by applying it in the domain of e-Science, in particular in managing scientific knowledge in models and experimental data.

References

1. W3C: Semantic web. (http://www.w3.org/2001/sw/)
2. McBride, B.: Four steps towards the widespread adoption of a semantic web. In: Proceedings of the First International Semantic Web Conference, Sardinia, Italy (2002)
3. Etzioni, O., Gribble, S., Halevy, A., Levy, H., McDowell, L.: An evolutionary approach to the semantic web. In Poster Presentation at the First International Semantic Web Conference (2002)
4. Quan, D., Huynh, D., Karger, D.R.: Haystack: A platform for authoring end user semantic web applications. International Semantic Web Conference (2003) 738–753
5. Popov, B., Kiryakov, A., Kirilov, A., Manov, D., Ognyanoff, D., Goranov, M.: KIM - Semantic Annotation Platform. International Semantic Web Conference (2003) 834–849
6. Dzbor, M., Motta, E., Domingue, J.: Opening up magpie via semantic services. In McIlraith, S.A., Plexousakis, D., van Harmelen, F., eds.: International Semantic Web Conference. Volume 3298 of Lecture Notes in Computer Science., Springer (2004) 635–649
7. Motta, E., Domingue, J., Cabral, L., Gaspari, M.: IRS-II: A Framework and Infrastructure for Semantic Web Services. International Semantic Web Conference (2003) 306–318
8. HP Labs Semantic Web Activity: Jena Semantic Web Toolkit. (http://www.hpl.hp.com/semweb/)
9. Broekstra, J., Kampman, A., van Harmelen, F.: Sesame: An architecture for storing and querying rdf data and schema information. In D. Fensel, J. Hendler, H. Lieberman, and W. Wahlster, editors, Semantics for the WWW. MIT Press. (2001)
10. Cabral, L., Domingue, J., Motta, E., Payne, T.R., Hakimpour, F.: Approaches to semantic web services: an overview and comparisons. In Bussler, C., Davies, J., Fensel, D., Studer, R., eds.: ESWS. Volume 3053 of Lecture Notes in Computer Science., Springer (2004) 225–239

Ontology-based information extraction for market monitoring and technology watch[⋆]

Diana Maynard[1], Milena Yankova[1], Alexandros Kourakis[2], Antonis Kokossis[2]

[1]Department of Computer Science, University of Sheffield, UK
[2]University of Surrey, UK
diana,milena@dcs.shef.ac.uk
a.kourakis,a.kokossis@surrey.ac.uk

Abstract. The h-TechSight Knowledge Management Portal (KMP) enables support for knowledge-intensive industries in monitoring information resources on the Web, as an important factor in business competitiveness. The portal contains tools for identification of concepts and terms from an ontology relevant to the user's interests, and enables the user to monitor them over time. It also contains tools for ontology management and modification, based on the results of targeted knowledge extraction from the web. The platform provides a means for businesses to keep track of trends and topics of interest in their field, and alert them to changes. In this paper we focus on the tools for targeted search and ontology management, driven by an ontology-based information extraction system, which has been evaluated over a test set of 38 documents and achieves 97% Precision and 92% Recall.

1 Introduction

The growing pervasiveness of Knowledge Management (KM) in industry marks an important new watershed. KM has become embedded in the strategy, policy and implementation processes of institutions and organisations worldwide. The global KM market has doubled in size since 1991 and is projected to exceed US$8.8 billion in 2005. KM applications are expected to save Fortune 500 companies around $31 billion, and the broader application cost has similar projected forecasts. Although the tools and resources developed in h-TechSight are targeted towards SMEs, there are important implications for the growth and dispersion of such new technologies to industry as a whole. h-TechSight aims to pave the way for such development by providing a variety of knowledge management tools in its portal. In this paper, we focus particularly on the underlying Information Extraction (IE) technology, and show how enhancing traditional IE with ontological information can lead to more interesting and useful acquisition of knowledge and benefit real users in industry.

The h-TechSight KMP is a knowledge management platform with intelligence and insight capabilities for technology intensive industries. It integrates a variety of next generation knowledge management (NGKM) technologies in order to observe information resources automatically on the internet, and notify users about changes occurring in their domain of interest. There are various new technologies developed in this research:

- a tool/model for the development of ontologies, which can be used to describe concepts and trends in the user's domain of interest;
- a tool/model for the development of generic and targeted search agents which can use these ontologies to search for business intelligence from diverse web-based sources;
- a platform for integrating information from various sources and consolidating, analysing and publishing this information.

There are also new competences in the form of knowledge about porting the tools and methodologies into any industry/technology, and about localising support services throughout Europe.

[⋆] This work is partially supported by the EU-funded Knowledge Web network of excellence ((IST-2004-507482) and SEKT project (IST-2004-506826)

1.1 Technology watch in the employment domain

Employment is a generic domain into which a great deal of effort in terms of knowledge management has been placed, because every company, organization and business unit must encounter it. Human Resources departments often have an eye open for knowledge management in order to monitor their environment in the best way, and many recruitment consultant companies have watchdogs to monitor and alert them to changes. There exist a variety of job search engines (portals) which use knowledge management extensively to link employees and employers, e.g. JobSearch[1] and Job Portals[2].

The employment domain is also chosen for h-TechSight because it contains many generic kinds of concepts. First this means that an existing IE system can more easily be ported to this domain (because it does not require too much adaptation), and second, it does not require a domain expert to understand the terms and concepts involved, so the system can easily be created by a developer without special domain skills. These two considerations are very important in the fast development of a system to be used as an example application [9].

The employment application in the KMP aims to alert users to technological changes, since job advertisements are a very good indicator of moving trends in the field. By monitoring these advertisements over a period of months or even years, we can examine, for example, changes in the requirements for particular skills and kinds of expertise required, how salaries fluctuate, what kinds of qualifications are being demanded, and benefits awarded to employees.

1.2 Monitoring of the news domain

The news domain is another clear area where it is important for companies to keep a close eye on technological developments in their field. Primary market players for this are the pharmaceutical industry and the oil and gas industry. Pharmaceutical companies need to extract knowledge from diverse sources in order to predict pharmacological and toxicological effects, for example integrating knowledge from newly acquired organisations and keeping a close watch on news of and reports from their competitors. The oil and gas industry is currently faced with increasing pressures to create higher quality and more environmentally friendly products, and therefore such companies need up-to-the-minute access to news, reports, and experiences of colleagues around the world in order to leverage such information and respond to critical information requests from government agencies. Our application for the news domain is aimed at helping companies to access and monitor such information quickly and accurately, bringing new products, processes and technologies to their attention, as well as tracking the progress of rival companies in the field.

1.3 The h-Techsight Knowledge Management Platform

In this paper we shall focus on the application mode of the KMP, which is used for analysing and enhancing previously discovered information. The Targeted Search Module (Application Mode) can either be used as standalone, if the user already has access to the information sources required, or combined with the other tools in the platform such as the Generic Search module in order to discover such sources. In the following sections, we shall describe the tools for the data-driven analysis of terminology in the portal. These aim at creating semantic metadata automatically from web-mined documents, and monitoring concepts and instances (domain-specific terms) extracted over time. We have developed sample applications in the employment and news domains in the field of chemical engineering.

[1] http://www.job-search.com/

[2] http://www.aspanet.org/solutionstemp/jobport.html

2 Ontology-based Information Extraction

The advent of tools and resources for the semantic web brings new challenges to the field of Information Extraction (IE), and in particular with respect to Ontology-Based IE (OBIE). Such tools are being developed within the context of projects such as SEKT[3] and others (see Section 5). One of the important differences between traditional IE and OBIE is the use of a formal ontology rather than a flat lexicon or gazetteer structure. This may also involve reasoning. Another difference is that OBIE not only finds the (most specific) type of the extracted entity, but it also identifies it, by linking it to its semantic description in the ontology. This allows entities to be traced across documents and their descriptions to be enriched through the IE process.

If the ontology is already populated with appropriate instances, the task of an OBIE system may be simply to identify instances from the ontology in the text. Similar methodologies can be used for this as for traditional IE systems, but using an ontology rather than a flat gazetteer. For rule-based systems, this is relatively straightforward, other than in the case of ambiguity. For learning-based systems, however, this is more problematic because training data is required and collecting such training data is likely to be a large bottleneck. Unlike traditional IE systems for which training data exists in domains like news texts in plentiful form, there is a dearth of material currently available for semantic web applications. New training data needs to be created manually or semi-automatically, which is a time-consuming and onerous task, although systems to aid such metadata creation are currently being developed (see Section 5).

The advantage of OBIE over traditional IE is that the output (semantic metadata about the text) is linked to an ontology, so this enables us to extract much more meaningful information about the text, for example making use of relational information or performing reasoning. We therefore can get a much better "snapshot" of the text and draw more meaningful and useful conclusions from it. For example, in the employment domain, identifying the locations where there are job vacancies is handy (as can be done with traditional IE), but linking towns and cities to areas and countries provides us with much more useful information, because we can then perform analyses about specific areas (for example, that the computer industry is growing in the North of England, or that London-based jobs are providing better benefits packages than those in the rest of the UK).

3 GATE

GATE is an architecture for language engineering developed at the University of Sheffield [1], containing a suite of tools for language processing, and in particular, a vanilla IE system ANNIE. In traditional IE applications, GATE is run over a corpus of texts to produce a set of annotated texts. In h-TechSight, the input to GATE takes the form of a set of URLs of target webpages, and an ontology of the domain. Its output comprises annotated instances of the concepts from the ontology. The ontology sets the domain structure and priorities with respect to relevant concepts with which the application is concerned.

GATE's IE system is rule-based, which means that unlike machine-learning based approaches, it requires no training data [8]. On the other hand, it requires a developer to create rules manually, so it is not totally dynamic. The architecture consists of a pipeline of processing resources which run in series. Many of these processing resources are language and domain-independent, so that they do not need to be adapted to new applications [6]. Pre-processing stages include word tokenisation, sentence splitting, and part-of-speech tagging, while the main processing is carried out by a gazetteer and a set of grammar rules. These generally need to be modified for each domain and application, though the extent to which this is necessary depends on the complexity and generality of the domain. The gazetteer contains a set of lists which help identify instances in the text. Traditionally, this is a flat structure, but in an OBIE application, these lists can be linked directly to an ontology, such that instances found in the text are then related back to the ontology.

[3] http://www.sekt.semanticweb.org

3.1 GATE in h-TechSight

The GATE application performs targeted information extraction relative to a domain and ontology, enabling statistical information to be gathered about the data collected. Inferences drawn from this information pave the way for the monitoring of trends of new and existing concepts and instances. For example, companies can track information about their rivals over time, and check for the emergence of new companies, products and technologies.

The GATE application consists of 5 basic stages:

1. web mining application to find relevant documents (or manual input of relevant documents);
2. selection of concepts in which the user is interested;
3. information extraction;
4. visual presentation of results (annotation of instances) and statistical analysis
5. ontology modification (an ontology editor is used to enrich the existing ontology from the results of the analysis)

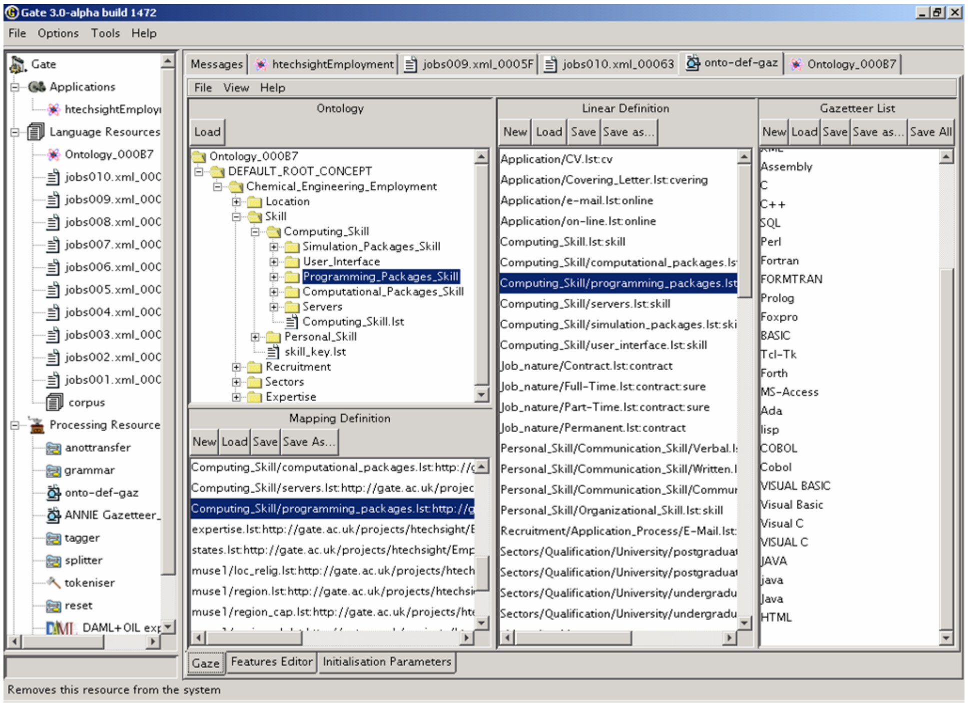
The application uses two main inputs: a web mining application which feeds relevant URLs to GATE based on the user's query, and a domain ontology. Alternatively, the user can input their own relevant documents to GATE. The texts are automatically annotated with semantic information based on the concepts in the ontology. Instances in the text can not only be visualised (through colour-coding) but can also be output in two forms: into a database for further processing, and in the form of a new ontology (DAML+OIL or RDF).

h-TechSight proceeds a stage further than traditional IE systems and other systems performing OBIE (see Section 5), by not only performing metadata generation and ontology population (by adding new instances to the ontology), but also by enabling the process of ontology evolution. By this we mean that the IE application serves not only to populate the ontology with instances, but also to modify and improve the ontology itself on the conceptual level. Statistical analysis of the data generated can be used to determine how and where this should take place. For example, a set of instances will be linked to a concept in the ontology, but this concept may be too general. A clustering algorithm can be used to group such instances into more fine-grained sets, and thereby lead to the addition of new subconcepts in the hierarchy. h-TechSight is unique in performing monitoring of the data over time, which can also lead to suggested changes in the ontology.

3.2 Application for the employment domain

For the employment domain in h-TechSight, a domain-specific application has been created, which searches for instances of concepts present in a sample employment ontology. The ontology has 9 main concepts: Location, Organisation, Sectors, JobTitle, Salary, Expertise, Person and Skill. Each concept in the ontology has a set of gazetteer lists associated with it. Some of these (generic lists) are reused from previous applications, while others (domain-specific lists) need to be created from scratch. In total there are around 60 domain-specific lists, and 50 generic lists. The generic lists are quite large (around 29,000 entries) and contain common entities such as first names of persons, locations, abbreviations etc. Collection of lists is done through corpus analysis (examining the texts manually and/or performing statistical analysis to spot important instances and concepts), unless a set of texts has been manually annotated by a user, in which case, the list collection process can be automatic [5]. For the employment domain, we used a combination of methods. We annotated around 20 documents manually and used this to collect lists automatically. This enabled us to bootstrap the development of the system and then complete the lists through further text analysis methods.

Grammar rules for recognition of new types of entities mainly use the gazetteer lists. However, not all entities can be recognised just from gazetteer lists. Some entities require more complex rules based on contextual



Gate
Applications
htechsightEmployı
Language Resources
Ontology_000B7
jobs010.xml_00C
jobs009.xml_00C
jobs008.xml_00C
jobs007.xml_00C
jobs006.xml_00C
jobs005.xml_00C
jobs004.xml_00C
jobs003.xml_00C
jobs002.xml_00C
jobs001.xml_00C
corpus
Processing Resource
anottransfer
grammar
onto-def-gaz
ANNIE Gazetteer_
tagger
splitter
tokeniser
reset
DAML+OIL exp

Ontology_000B7
DEFAULT_ROOT_CONCEPT
Chemical_Engineering_Employment
Location
Skill
Computing_Skill
Simulation_Packages_Skill
User_Interface
Programming_Packages_Skill
Computational_Packages_Skill
Servers
Computing_Skill.lst
Personal_Skill
skill_key.lst
Recruitment
Sectors
Expertise

Computing_Skill/computational_packages.lst:http://c
Computing_Skill/servers.lst:http://gate.ac.uk/projec
Computing_Skill/programming_packages.lst:http://g
expertise.lst:http://gate.ac.uk/projects/htechsight/E
states.lst:http://gate.ac.uk/projects/htechsight/Emp
muse1/loc_relig.lst:http://gate.ac.uk/projects/htech
muse1/region.lst:http://gate.ac.uk/projects/htechsi
muse1/region_cap.lst:http://gate.ac.uk/projects/hte

Application/CV.lst:cv
Application/Covering_Letter.lst:cvering
Application/e-mail.lst:online
Application/on-line.lst:online
Computing_Skill.lst:skill
Computing_Skill/computational_packages.ls
Computing_Skill/programming_packages.lst
Computing_Skill/servers.lst:skill
Computing_Skill/simulation_packages.lst:ski
Computing_Skill/user_interface.lst:skill
Job_nature/Contract.lst:contract
Job_nature/Full-Time.lst:contract:sure
Job_nature/Part-Time.lst:contract:sure
Job_nature/Permanent.lst:contract
Personal_Skill/Communication_Skill/Verbal.l
Personal_Skill/Communication_Skill/Written.l
Personal_Skill/Communication_Skill/Commur
Personal_Skill/Organizational_Skill.lst:skill
Recruitment/Application_Process/E-Mail.lst:
Sectors/Qualification/University/postgraduat
Sectors/Qualification/University/postgraduat
Sectors/Qualification/University/undergradu
Sectors/Qualification/University/undergradu

Assembly
C
C++
SQL
Perl
Fortran
FORMTRAN
Prolog
Foxpro
BASIC
Tcl-Tk
Forth
MS-Access
Ada
lisp
COBOL
Cobol
VISUAL BASIC
Visual Basic
Visual C
VISUAL C
JAVA
java
Java
HTML

Removes this resource from the system

Fig. 1. Screenshot of Populated Employment Ontology in GATE

information. These may also use special lists that contain keywords and are used to assist such contextually-based rules. Some of the keyword lists are also attached to the ontology, because they clearly show the class to which the identified entity belongs. All lists that correspond to the ontology are ordered in a hierarchy similar to the class hierarchy in the ontology. A section of the ontology, the mappings from the lists to the ontology, and the contents of a list is shown in Figure 1.

The concepts in which we are interested can be separated into 3 groups. The first consists of classic named entities which are general kinds of concepts such as Person, Location, Organisation. The second is more specific to the chosen domain of employment, and consists of the following types:

- JobId - shows the ID of posted job advertisements;
- Reference - shows the reference code of the job position;
- Status - shows the employment/position type;
- Application - shows the documents necessary and the method of job application (e.g. by email, letter, whether a CV should be sent, etc.);
- Salary - shows the information available in the text about salary rates, bonus packages, compensations, benefits etc.;
- Qualification - shows the qualifications required for the advertised position, mainly a University degree;
- Citizenship - shows restrictions about the applicant's citizenship, eligibility, etc.;
- Expertise - shows the required expertise / skills for the job.

For both groups, the grammar rules check if instances found in the text belong to a class in the ontology and if so, they link the recognised instance to that same class and add the following features:

```
EntityType.ontology = ontology url,
EntityType.class = class name
```

The third group presents instances already annotated with HTML or XML tags (if such exist), and consists of the following:

- Company - contains the name of the organisation advertising the job;
- Date_Posted - shows the date when the job advertisement was posted;
- Title - shows the job title;
- Sector - shows the sector of the job that is advertised.

If these are not already annotated in the texts, they are identified using further rules.

The grammar rules for creating annotations are written in a language called JAPE [2]. The rules are implemented in a set of finite-state transducers, each transducer usually containing rules of a different type, and are based on pattern-matching. In traditional IE applications, the rules find a pattern on the LHS, in the form of annotations, and on the RHS an action such as creating a new annotation for the pattern. In OBIE applications such as this, the rules also add information about the class and ontology on the RHS of the rule. So for example the string "PhD" found in the text might be annotated with the features:

```
{class = Postgraduate}
{ontology = http://gate.ac.uk/projects/htechsight/Employment}
```

This information is taken from the gazetteer, which is mapped to an ontology, as described earlier. In total the application contains 33 grammars, which run sequentially over the text. Each grammar contains anything from 1 to about 20 rules, depending on the complexity of the annotation type.

3.3 Adaptation to the news domain

The GATE application for the news domain is focused on the area of chemical technologies. For this, a new domain-specific ontology and set of texts is required as input to the system. We have constructed a sample ontology consisting of 13 concepts related to the technologies domain, such as Corrosion, Thermodynamics, Optimization, Reaction, Equipment, etc. Some gazetteer lists were reused from the employment domain, while others needed to be created from scratch and mapped to the ontology in the correct place. In total there are 181 lists.

Some of the grammar rules used for the employment domain were directly reused for the news domain, while others had to be created from scratch. The aim was to minimise the amount of adaptation necessary; however, the nature of technical terminology makes generic kinds of rules very difficult to implement, because of its specialised nature and the fact that not only are the terms different, but the syntax and structure of more technical texts can be very different. A discussion of the problems in adapting an IE system to different genres and domains can be found in [10,7]. For this domain, the help of a chemical engineering expert was required, since it was impossible for a non-domain expert to understand correctly which instances should be linked with which concepts, and therefore to construct appropriate rules.

Unlike the employment domain, the news application also finds relations between entities in the text. This is accomplished by using JAPE grammar rules to search for instances belonging to two different concepts in the hierarchy, and analysing the syntax of the text between the two instances using a Noun Phrase and Verb Phrase chunker (also developed using JAPE grammars), to extract relevant relations based on verbal groups. So for example, we use patterns such as $< instance >< Verb >< Instance >$ to extract triples like $< TIPunit >< upgrades >< octane >$ (where "upgrades" is a relation between the two terms TIP unit and octane) from the sentence "The TIP unit upgrades the octane of the feed to achieve research octanes of close to 90 in the product". We can then form clusters between related concepts, using information collected about such relations and also infer other important knowledge. This is another example of how ontologies can help us to extract more useful information, because instead of simply linking instances from certain annotation types (e.g. finding relations between Person and Organisation), we can progress up or down the hierarchy in order to obtain more or less fine-grained information.

4 Presentation and analysis of results

The GATE application for the employment domain has been implemented in the h-TechSight portal as a web service. The user may select a URL and choose the concepts for the ontology. Then by invoking the service, a new web page is created with highlighted colour-coded annotations of the web page selected. The results are collected by dynamically populating a Microsoft Access database, and their statistical analysis is presented inside the KMP. The database has the following structure:

- Concepts: the concept which the record set of the database is about;
- Annotations: the instance of the record set annotated inside a document;
- Document _ID: a unique ID for the document;
- Time_Stamp: a time stamp found inside the document.

4.1 Monitoring instance-based dynamics

One of the most primitive dimensions of ontologies is the display of data as concrete representations of abstract concepts, i.e. as instances. GATE leads the data-driven analysis in h-TechSight, as it is responsible for extracting from the text instances represented in the ontology. Statistical analysis is then invoked to present instance-based dynamics.

In the h-TechSight platform, we try to monitor the dynamics of ontologies using two approaches: dynamics of concepts and dynamics of instances. Users may not only annotate their own websites according to their ontology, but may also see the results of a dynamic analysis of the respective domain. They may see tabular results of statistical data about how many annotations each concept had in the previous months, as well as seeing the progress of each instance in previous time intervals (months). Following this analysis, end users may also see the dynamics of instances by means of an elasticity metric that indicates the trend of each individual instance. Developments in the GATE results analysis have eliminated human intervention, as the results are created automatically in a dynamic way. The two approaches to the monitoring of dynamics are described in more detail below.

Dynamic metrics of concepts are calculated by counting the total occurrences of annotated instances over time intervals (per month). By clicking on the concepts, a user may see the instances related to a concept. Instances are presented in a time series where the total occurrences per month and a calculation of an elasticity metric of instances are presented in tabular form. The elasticity metric (Dynamic Factor) counts the differences between the total occurrences of every instance over time intervals (per month) taking into consideration the volume of data of each time period. The mathematical type that calculates the DF takes into consideration the differences of volume of data (documents annotated by GATE) of each time period (months).

4.2 Analysis of results

From Table 1 we can examine how particular kinds of expertise are being sought over a period of time. Clearly, looking at just 3 months of data is not sufficient to make an informed analysis about trends, but looking at data over a longer period of time will be a useful indicator. Instances with a negative Dynamic Factor (DF) show an overall downward trend. The higher the dynamic factor, the greater the upward trend.

Instances	Dynamic Factor	Jan	Feb	Mar
1 year as a J2EE designer	-1	0	1	0
1 year JSP experience	48	0	0	2
2 years banking	23	0	0	1
2EE	145	0	15	6

Table 1. Dynamics of Instances for the Concept "Expertise"

Instances	Dynamic Factor	Jan	Feb	Mar
ARC	145	0	12	6
Archimedia SA	-1	0	1	0
Army	23	0	2	1
AT&T	-1	0	2	0
AT&T Wireless	-1	0	3	0
BA	23	0	3	1
BMI British Midland	-335	1	3	0
British Airways	-163	1	11	7

Table 2. Dynamics of Instances for the Concept "Organisation"

From Table 2 we can see how frequently different companies are placing job advertisements on the portals under scrutiny. One important fact to notice is that at the moment, if the same company is referred to in two (or more) different ways, the results will be stored individually, thus skewing the figures. For example, the counts for BA and British Airways are stored separately, because the system does not recognise that these refer to the same company We are currently implementing a coreference mechanism to cluster such term variants together, so that we only calculate one overall score rather than two separate ones. This is also extended to cluster more loosely connected variants, so for example the term C and C++ might be grouped together. In this way we can show two separate views of such clusters an overall count and DF for the cluster, and a table showing details of the individual instances that form the cluster.

4.3 Evaluation of the IE technology

We conducted an initial evaluation of the IE application to see how well the system found relevant instances of the concepts. We tested the system on a small set of 38 documents containing job advertisements in the Chemical Engineering domain, mined from the website http://www.jobserve.com. The web portal is mined dynamically using a web content agent written in WebQL, a commercial web crawling software[4]. We manually annotated these documents with the concepts used in the application, and used the evaluation tools provided in GATE to compare the system results with the gold standard. Overall, the system achieved 97% Precision and 91.5% Recall, with an F-Measure of 94.2%.

4.4 User Feedback

The KMP has been tested by real users in industry, such as Bayer Technology Services and IChemE. Users found that it was very helpful in increasing the efficiency of acquiring knowledge and supporting project work in industry, by helping to automatically scan, filter, structure and store the wealth of information available on the web related to their needs. For Bayer, the potential areas of application spanned from research and development, engineering and production, to marketing and management.

[4] http://www.webql.com

Users at IChemE, a leading international body which provides services for chemical engineers world-wide, claimed that the employment application was a very sound idea, and that it "would be a very valuable means of graduates gaining a fresh insight into their jobs and related training which may be narrower than ideally it should be due to company constraints (i.e. time and money for development!)".

One important fact to note is that due to the complexity of the underlying system, it is not really feasible for non-IE experts to adapt the system to new domains. However, since the system runs as a web service, the end user need have no knowledge of the underlying technology in order to use the system, so this is not necessarily a problem.

5 Related Work

There currently exist several other systems for automatic semantic metadata creation of web-based documents.

Magpie [4] is a suite of tools which supports the interpretation of webpages and "collaborative sense-making", by annotating a text with instances from a known ontology. These instances can be used as a confidence measure for carrying out some services. The principle behind it is that it uses an ontology to provide a very specific and personalised viewpoint of the webpages the user wishes to browse. This is important because different users often have different degrees of knowledge and/or familiarity with the information presented, and have different browsing needs and objectives.

KIM [11] is an architecture for automatic semantic annotation developed within a platform for semantic-based indexing and retrieval from large document collections. KIM contains an instance base which has been pre-populated with 200,000 entities (mostly locations), and performs information extraction based on GATE. Essentially, KIM recognises entities in the text with respect to the KIM ontology, and adds new instances where they do not already exist.

The SemTag system [3] performs large-scale semantic annotation with respect to the TAP ontology. It first performs a lookup phase annotating all possible mentions of instances from the TAP ontology, and then performs disambiguation, using a vector-space model to assign the correct ontological class or determine that this mention does not correspond to a class in the ontology.

h-TechSight and KIM both use the same core IE system, although KIM uses a general IE application while h-TechSight uses one tuned to the specific domain and ontology being used. KIM supports ontology modification in that it identifies new instances and adds them to the ontology. h-TechSight also supports ontology evolution, whereby the actual structure of the ontology can be modified as a result of the instances discovered, in a semi-automatic way (making suggestions to the user).

h-TechSight also has a slightly different goal from systems such as SemTag and KIM, in that these are domain-independent, large-scale approaches, while in h-TechSight the IE algorithms have been specifically created for particular domains and therefore can offer the extended functionality. Finding the balance between sophisticated functionality and good IE performance and domain independence is always difficult. The approaches used in KIM and SemTag are more appropriate for large-scale automatic annotation systems, while user involvement in the process of adding new instances is more beneficial for domain-specific applications which can afford to be semi-automatic and which, by their nature, are more suitable for user involvement.

There are also many other research efforts in the area of NGKM. Two major projects in this area, which are frequently referred to as grounding initiatives, are On-To-Knowledge[5] and Vision[6]. Related research on

[5] Content-driven Knowledge Management Tools through Evolving Ontologies IST-1999-10132

[6] http://km-aifb.uni-karlsruhe.de/fzi/vision/

Knowledge Management System development is discussed in detail in [12], but is not so relevant to this work, where we focus on the application-specific tools in the KMP.

6 Conclusions

In this paper we have presented an application for automatic knowledge extraction, management and monitoring in the Chemical Engineering domain, integrated in a dynamic knowledge management portal. Combined with the other tools and applications for knowledge engineering found within the portal, it forms the basis of a system for information retrieval, terminology acquisition and technology watch. GATE makes use of terminological processing and domain-specific IE to evolve existing ontologies automatically and to enable thte monitoring of domain-specific information relevant to the user. The application has been tested in the Employment sector with excellent results, and has been successfully ported to other genres of text such as news items and company reports.

References

1. H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan. GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. In *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02)*, 2002.
2. H. Cunningham, D. Maynard, and V. Tablan. JAPE: a Java Annotation Patterns Engine (Second Edition). Research Memorandum CS–00–10, Department of Computer Science, University of Sheffield, November 2000.
3. S. Dill, N. Eiron, D. Gibson, D. Gruhl, R. Guha, A. Jhingran, T. Kanungo, S. Rajagopalan, A. Tomkins, J. A. Tomlin, and J. Y. Zien. SemTag and Seeker: Bootstrapping the semantic web via automated semantic annotation. In *Proceedings of WWW'03*, 2003.
4. J. Domingue, M. Dzbor, and E. Motta. Magpie: Supporting Browsing and Navigation on the Semantic Web. In N. Nunes and C. Rich, editors, *Proceedings ACM Conference on Intelligent User Interfaces (IUI)*, pages 191–197, 2004.
5. D. Maynard, K. Bontcheva, and H. Cunningham. Automatic Language-Independent Induction of Gazetteer Lists. In *Proceedings of 4th Language Resources and Evaluation Conference (LREC'04)*, 2004. http://gate.ac.uk/sale/lrec2004/gazcollector.pdf.
6. D. Maynard and H. Cunningham. Multilingual Adaptations of a Reusable Information Extraction Tool. In *Proceedings of the Demo Sessions of EACL'03*, Budapest, Hungary, 2003. http://gate.ac.uk/sale/eacl03/demo.pdf.
7. D. Maynard, V. Tablan, K. Bontcheva, H. Cunningham, and Y.Wilks. Multi-source entity recognition – an information extraction system for diverse text types. Research Memorandum CS–03–02, Department of Computer Science, University of Sheffield, April 2003.
8. D. Maynard, V. Tablan, and H. Cunningham. NE recognition without training data on a language you don't speak. In *ACL Workshop on Multilingual and Mixed-language Named Entity Recognition: Combining Statistical and Symbolic Models*, Sapporo, Japan, 2003. http://gate.ac.uk/sale/acl03/surprise.pdf.
9. D. Maynard, V. Tablan, H. Cunningham, C. Ursu, H. Saggion, K. Bontcheva, and Y. Wilks. Architectural Elements of Language Engineering Robustness. *Journal of Natural Language Engineering – Special Issue on Robust Methods in Analysis of Natural Language Data*, 8(2/3):257–274, 2002. http://gate.ac.uk/sale/robust/robust.pdf.
10. D. Maynard, V. Tablan, C. Ursu, H. Cunningham, and Y. Wilks. Named Entity Recognition from Diverse Text Types. In *Recent Advances in Natural Language Processing 2001 Conference*, pages 257–274, Tzigov Chark, Bulgaria, 2001. http://gate.ac.uk/sale/ranlp2001/maynard-etal.pdf.
11. B. Popov, A. Kiryakov, A. Kirilov, D. Manov, D. Ognyanoff, and M. Goranov. KIM ?Semantic Annotation Platform. In *2nd International Semantic Web Conference (ISWC2003)*, pages 484–499, Berlin, 2003. Springer.
12. M. Stollberg, A. Zhdanova, and D. Fensel. h-TechSight – A Next Generation Knowledge Management Platform. *Journal of Information and Knowledge Management*, 3 (1):1–22, 2004.

Building Semantic Web Applications as Information/Knowledge Sharing Systems

Hideaki Takeda[1][2] and Ikki Ohmukai[1]

[1] National Institute of Informatics (NII)
2-1-2, Hitotsubashi, Chiyoda-ku, Tokyo, Japan
[2] The Graduate University of Advanced Studies (Sokendai), Japan

Abstract. In this paper, we propose the methodology to design Semantic Web applications that can be acceptable widely by ordinary people. We first analyze "miracle of web" as an information sharing tool that is basically difficult for people to accept. In order to overcome this point, Semantic Web applications should have two types of gratification simultaneously, i.e., instant gratification that can be obtained even without information/knowledge sharing, and delayed gratification that can be obtained through information/knowledge sharing. The gap between two types of gratification can be bridged by the *translucence strategy* that lures people into information/knowledge sharing by showing delayed gratification within kissing distance. We then show our experience to build information/knowledge sharing tools with the above methodology. One is Community Navigator that helps participants for a conference to share knowledge like topics and related people. The other is Semblog systems that helps weblog people to exhibit and exchange their information more.

1 Introduction

Researchers tend to be obsessed with technical details when solving problems, i.e., they tend to forget the purpose or mission of the problem itself. It likely happens more when technical problems looks complicated and difficult. Semantic Web is probably the case. There is a nice technical road map like "the layer cake" and each step looks challenging technologically. According to the road map, many technologies have been developed like RDFS and OWL. But we are not sure how these technologies would contribute the purpose of the original problem. We start with this viewpoint.

Web is no doubt an information sharing tool. Scientists have been eager to exchange data and information among their organizations and communities quickly and easily Web have firstly spread out to people in universities. Then ordinary people find out that web is also useful for them, and expand web for their use. We have so accustomed to life with web, but wide dissemination of web is probably "miracle of web", because "**people are basically reluctant to exhibit their information.**"

Recall our daily life. We are even reluctant to put a free ad paper on wall in a supermarket nearby. Before web, there exist information publishing tools like ftp, but only limited people used such tools. Without intention to exhibit information to others, information sharing can not work. Information sharing is basically a difficult task to involve people.

As successor of web, Semantic Web should be an information sharing tool. Of course Semantic Web is going one step beyond web, i.e., aims to be a knowledge sharing tool. So it is reasonable that Semantic Web will be more difficult than web for dissemination.

We should develop Semantic Web applications carefully to involve people to use them as information/knowledge sharing.

The paper is organized as follows; In the following section, we show our methodology to build information/knowledge sharing systems. Then we show two systems we built in the following two sections (Section 3 and 4). In Section 5, we dicuss other information sharing systems and conclude the pager in Section 6.

2 Double-loop Gratification

We can enumerate many benefits for information/knowledge sharing, while there exist also hurdles for dissemination of information/knowledge sharing. One of the hurdles is the privacy and security issue that is related to sociological point of view.

Another is the feedback issue that is related to cognitive point of view. The feedback on contribution to information sharing is rarely visible. One of the reason why people do not wish to use information sharing tools is that their effort looks in vain because of lack of feedback. McDowell et al. [1] pointed out this issue as *instant gratification*. They said that instant gratification is needed to involve people in Semantic Web applications, and their application called Mangrove have succeeded because of realization of instant gratification.

We agree to importance of instant gratification, but instant gratification should be different in information/knowledge sharing applications. In Mangrove, users' contribution is quickly reflected to information sharing results by collecting and revising revise them as fast as it can. It is a nice feature but it sacrifices variety and scalability of information/knowledge sharing, because information/knowledge sharing takes time naturally.

We think that information/knowledge sharing applications should have two types of gratification simultaneously, i.e., instant gratification that can be obtained even without information/knowledge sharing, and delayed gratification that can be obtained through information/knowledge sharing. It always takes efforts for users to be accustomed with new applications. Instant gratification can be an anchor to keep users to use the applications. While users keeping to use them, delayed gratification that are real benefits of information/knowledge sharing arrives in them. The balance of two types of gratification is important rather than quantity of them. As I mentioned above, benefits from information/knowledge sharing tends to take time, it is too strict restriction to require instant gratification by information/knowledge sharing.

Web has both types of gratification. Authoring hypertexts gives people instant gratification. It is a new fascinating method for people to organize own information that is difficult to write down as stable well-organized form like word processing documents. Since authoring hypertexts and publishing them are so closely connected in Web, people are publishing their information with almost no extra efforts. Then they will receive delayed gratification as feedback from users who read their published information.

The problem is how to design such systems with two types of gratification. Through our observation on other systems and our experience on information/knowledge sharing, we propose *translucence strategy* to make people to shift instant gratification receivers to delayed gratification receivers. The strategy is simple: just put people in a situation where they can feel possible delayed gratification within kissing distance. Then they shift to the next step where they can receive delayed gratification. The step should be minimum, i.e., it should be a very small amount of extra efforts to join information/knowledge sharing in addition to ordinary efforts to obtain instant gratification.

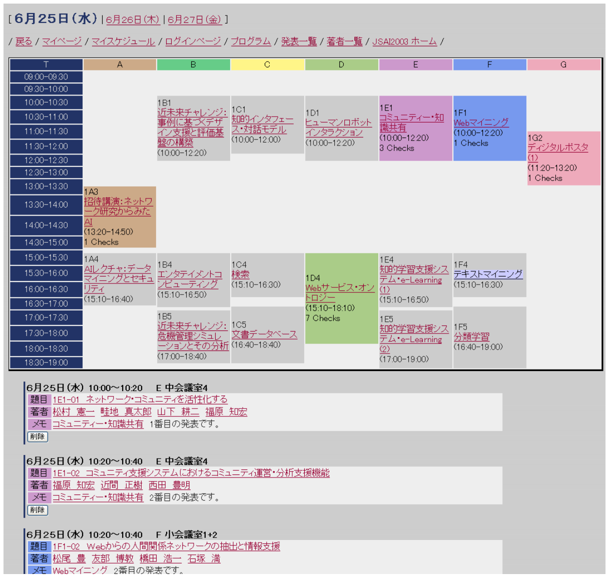
In the following section, we explain two systems we built and how the above strategy works on them. The same methodology was applied to other systems like Ba-log[2] that stimulates communication based on location with location-embedded weblog and Social Scheduler[3] that assists people to determine schedules of shared tasks by analyzing personal network.

3 Community Navigator: Collaborative Scheduling Support System for Conferences

We built a system called *Community Navigator* that supports conference participants by helping their own scheduling *and* communication among them[4].

The first look of the system is just a personal scheduling system for a conference (see Figure 1). Users can browse the timetable of the conference and detail of each session and paper, and click bottoms to slot in papers they like to listen. Then the system shows their personal schedule both as a timetable and a list of papers.

But the system has another function, i.e., information sharing and recommendation by interpersonal network. When users browse paper pages, they can also click authors' names to register them as their acquaintance. The system also shows a list of "know" people and "is known by" people in the personal scheduling pages (see Figure 2). The former means a person whom the owner of the personal scheduling page actually registers as acquaintance, and the latter means person who registers the owner of the page as acquaintance. After people register acquaintances, they can access detail information of their acquaintances and receive recommendation of papers and people from the system that calculates the degree of importance among their acquaintances.



[**6月25日（水）** | 6月26日（木） | 6月27日（金）]

/ 戻る / マイページ / マイスケジュール / ログインページ / プログラム / 発表一覧 / 著者一覧 / JSAI2003 ホーム /

T	A	B	C	D	E	F	G
09:00-09:30							
09:30-10:00							
10:00-10:30		1B1 近未来チャレンジ:事例に基づくデザイン支援と評価基盤の構築 (10:00-12:20)	1C1 知的インタフェース・対話モデル (10:00-12:00)	1D1 ヒューマンロボットインタラクション (10:00-12:20)	1E1 コミュニティー・知識共有 (10:00-12:20) 3 Checks	1F1 Webマイニング (10:00-12:20) 1 Checks	
10:30-11:00							
11:00-11:30							1G2 ディジタルポスタ(1) (11:20-13:20) 1 Checks
11:30-12:00							
12:00-12:30							
12:30-13:00							
13:00-13:30	1A3 招待講演:ネットワーク研究からみたAI (13:20-14:50) 1 Checks						
13:30-14:00							
14:00-14:30							
14:30-15:00							
15:00-15:30	1A4 AIレクチャ:データマイニングとセキュリティ (15:10-16:40)	1B4 エンタテイメントコンピューティング (15:10-16:50)	1C4 検索 (15:10-16:30)	1D4 Webサービス・オントロジー (15:10-18:10) 7 Checks	1E4 知的学習支援システム・e-Learning(1) (15:10-16:50)	1F4 テキストマイニング (15:10-16:30)	
15:30-16:00							
16:00-16:30							
16:30-17:00			1C5 文書データベース (16:40-18:40)			1F5 分類学習 (16:40-19:00)	
17:00-17:30		1B5 近未来チャレンジ:危機管理シミュレーションとその分析 (17:00-18:40)			1E5 知的学習支援システム・e-Learning(2) (17:00-19:00)		
17:30-18:00							
18:00-18:30							
18:30-19:00							

6月25日（水） 10:00～10:20　E 中会議室4
題目 1E1-01　ネットワーク・コミュニティを活性化する
著者 松村　憲一　畦地　真太郎　山下　耕二　福原　知宏
メモ コミュニティー・知識共有　1番目の発表です。
削除

6月25日（水） 10:20～10:40　E 中会議室4
題目 1E1-02　コミュニティ支援システムにおけるコミュニティ運営・分析支援機能
著者 福原　知宏　近間　正樹　西田　豊明
メモ コミュニティー・知識共有　2番目の発表です。
削除

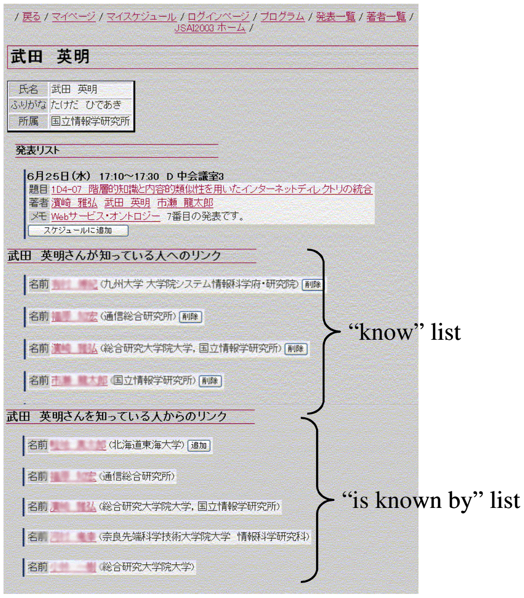
6月25日（水） 10:20～10:40　F 小会議室1+2
題目 1F1-02　Webからの人間関係ネットワークの抽出と情報支援
著者 松尾　豊　友部　博教　橋田　浩一　石塚　満
メモ Webマイニング　2番目の発表です。

Fig. 1. Community Navigator: The first look is just a scheduler

We applied the system to an academic conference called *JSAI2003* and the result is v ery successful.

The conference held three days in 2003. 259 papers were presented and about 400 people were participated in the conference. The system was used by 276 users, and among them 160 users added 1840 papers in their schedules and 99 users registered 840 persons as acquaintance. These are significant numbers as acquirement of users, because there are no obligation to use the system. About 40% of participants actually used the system, and about 60% among them stepped forward to sharing information stage.

In this system, instant gratification corresponds to personal scheduling function, and delay ed gratification to information sharing via interpersonal network. P ersonal scheduling function successfully attracted people to use the system. Our *translucence strategy* here is that we require minimum action like clicking their acquaintance and the system then starts information sharing with information already registered as personal scheduling. It is noted that the rate to enter the information sharing stage is 60%. We think the n umber is v ery successful but ev en with such a strategy, about a half out of initial users are inv olved in the information sharing stage. It suggests that involving people in the first stage as many as possible is important.



/ 戻る / マイページ / マイスケジュール / ログインページ / プログラム / 発表一覧 / 著者一覧 / JSAI2003 ホーム /

武田　英明

氏名	武田　英明
ふりがな	たけだ　ひであき
所属	国立情報学研究所

発表リスト

6月25日(水)　17:10〜17:30　D 中会議室3
題目 1D4-07　階層的知識と内容的類似性を用いたインターネットディレクトリの統合
著者 濱崎　雅弘　武田　英明　市瀬　龍太郎
メモ Webサービス・オントロジー　7番目の発表です。
スケジュールに追加

武田　英明さんが知っている人へのリンク

名前 [redacted] (九州大学 大学院システム情報科学府・研究院) 削除

名前 [redacted] (通信総合研究所) 削除

名前 [redacted] (総合研究大学院大学, 国立情報学研究所) 削除

名前 [redacted] (国立情報学研究所) 削除

武田　英明さんを知っている人からのリンク

名前 [redacted] (北海道東海大学) 追加

名前 [redacted] (通信総合研究所)

名前 [redacted] (総合研究大学院大学, 国立情報学研究所)

名前 [redacted] (奈良先端科学技術大学院大学　情報科学研究科)

名前 [redacted] (総合研究大学院大学)

Fig. 2. Community Navigator: It can work as a navigator of community

4 Semblog: Metadata-driven Personal Publishing System

In this section, we introduce a personal knowledge publishing system called *Semblog* that provides an integrated environment for distributing small contents and making human relationship seamlessly [5]. It enables people to exchange information and knowledge with easy and casual fashion in degrees of personal interest, e.g. checking, clipping, and posting with various metadata and Weblog tools.

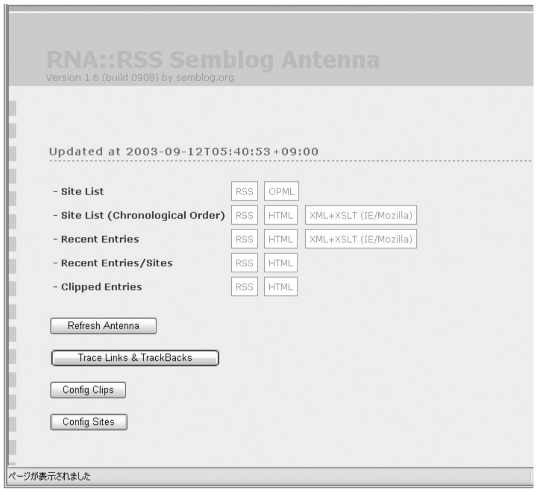
We developed two types of RSS aggregator called "RNA" and "glucose".

4.1 RNA: Web-based RSS Aggregator

RNA is a Web-based RSS aggregator written with Perl. A user can operate RNA through her/his Web server. Figure 3 shows a snapshot of RNA.

Firstly the user should register URIs of RSS in configuration page of RNA shown in Figure 4. The user can categorize these RSSs. List of sites checked by the user are converted into an RSS that can be used by other RSS-based applications again. RNA can also import and export OPML that is a standard metadata set for Web bookmark.

RNA produces site/entry list ordered by updated time of each element. After getting RSS files from various sources, RNA parses these RSSs and merges into single a "global" RSS tree. RNA converts this global tree to several forms by ordering chronologically. These partial trees are published as RSS and rendered into HTML.



RNA::RSS Semblog Antenna
Version 1.6 (build 0908) by semblog.org

Updated at 2003-09-12T05:40:53+09:00

- Site List RSS OPML
- Site List (Chronological Order) RSS HTML XML+XSLT (IE/Mozilla)
- Recent Entries RSS HTML XML+XSLT (IE/Mozilla)
- Recent Entries/Sites RSS HTML
- Clipped Entries RSS HTML

Fig. 3. Snapshot of RNA

Figure 5 shows a site list with HTML. User can browse description element of RSS in each channel (site).

RNA supports various output formats such like HTML, RSS and other forms i.e., JavaScript with server/client-side XSLT engines and original template engine. The user can create customized partial tree using plug-in and template script.

The user can save a favorite content in RNA to a clip list with one click. Clipped contents are stored in the "clipped" RSS tree and it is published like other RSS. RNA can post clips to social bookmark service such like del.isio.us[3].

RNA extracts TrackBack links from each entry in registered sites, and embeds TrackBack metadata in RSS and renders it.

The current version of RNA cooperates with RSS-based search engine such as Technorati[4] and Bulkfeeds[5]. By setting some keywords, the user can obtain new contents from non-registered sites.

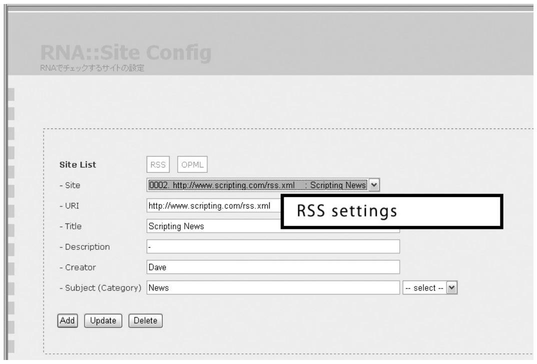
Most of RSS generated by Weblog and news sites includes categorical information with `<dc:subject>` vocabulary. RNA can aggregates only specific contents by selecting certain categories and re-distribute categorical RSS.

It is necessary to get RSS and build trees occasionally since those contents are changeable with update of information sources. RNA can update periodically by cron interface of the server. Update interface can be called both manually and remotely by XML-RPC message that is generated automatically by Weblog tools.

[3] http://del.isio.us/

[4] http://www.technorati.com/

[5] http://bulkfeeds.net/



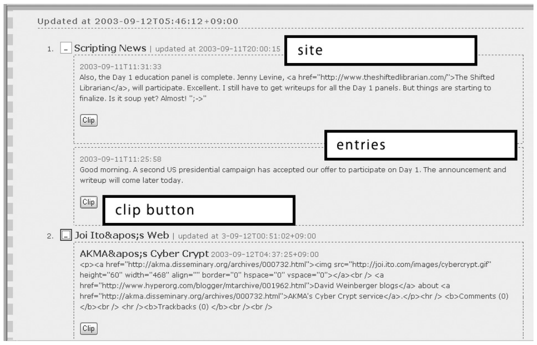
RNA::Site Config

RNAでチェックするサイトの設定

Site List

- Site: 0002. http://www.scripting.com/rss.xml : Scripting News
- URI: http://www.scripting.com/rss.xml
- Title: Scripting News
- Description: -
- Creator: Dave
- Subject (Category): News

Fig. 4. RSS Registration



Updated at 2003-09-12T05:46:12+09:00

1. Scripting News | updated at 2003-09-11T20:00:15

2003-09-11T11:31:33
Also, the Day 1 education panel is complete. Jenny Levine, <a href="http://www.theshiftedlibrarian.com/">The Shifted Librarian</a>, will participate. Excellent. I still have to get writeups for all the Day 1 panels. But things are starting to finalize. Is it soup yet? Almost! ";->"

2003-09-11T11:25:58
Good morning. A second US presidential campaign has accepted our offer to participate on Day 1. The announcement and writeup will come later today.

2. Joi Ito&apos;s Web | updated at 3-09-12T00:51:02+09:00

AKMA&apos;s Cyber Crypt 2003-09-12T04:37:25+09:00
<p><a href="http://akma.disseminary.org/archives/000732.html"><img src="http://joi.ito.com/images/cybercrypt.gif" height="60" width="468" align="" border="0" hspace="0" vspace="0"></a><br /> <a href="http://www.hyperorg.com/blogger/mtarchive/001962.html">David Weinberger blogs</a> about <a href="http://akma.disseminary.org/archives/000732.html">AKMA's Cyber Crypt service</a>.</p><hr /> <b>Comments (0)</b><br /> <hr /><b>Trackbacks (0) </b><br /><br />

Fig. 5. Site List in HTML

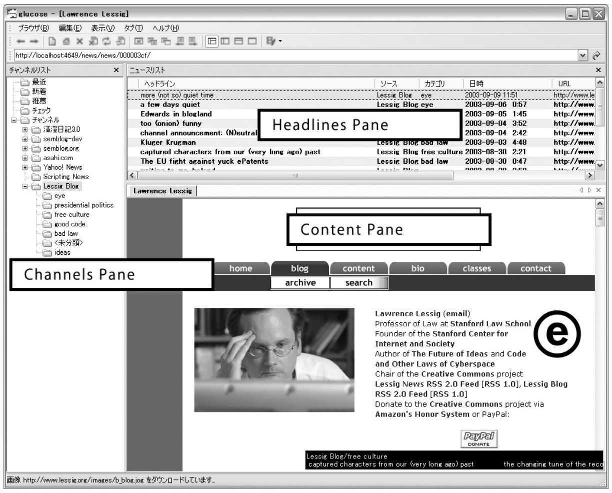
RNA checks syntax of acquired RSS and corrects them if they are not valid. RNA converts all versions of RSS into 1.0, which is based on RDF model.

4.2 Glucose: Stand-alone RSS Aggregator

Glucose is also an extended RSS aggregator for Windows. Figure 6 shows a snapshot of Glucose. Unlike orthodox aggregators, Glucose is developed to support information distribution process in cooperation with coordinating with RNA. Main functions and interfaces of Glucose are shown below.

Like in RNA, the user registers URIs of RSS or OPML site list. Glucose can access several news sites without RSS by ”sensor” script which extracts articles and converts them into RSS.

Glucose has three panes interface. The left pane shows ”RSS Channels” which is subscribed by the user. The upper right pane indicates the headline list of contents including title, updated time, source and category. The lowerright pane shows original contents.



http://localhost:4649/news/news/000003cf/

ヘッドライン	ソース	カテゴリ	日時	URL
more (not so) quiet time	Lessig Blog	eye	2003-09-09 11:51	http://www.le
a few days quiet	Lessig Blog	eye	2003-09-06 0:57	http://www.
Edwards in blogland			2003-09-05 1:45	http://www.
too (onion) funny			2003-09-04 3:52	http://www.
channel announcement: (N)eutral			2003-09-04 2:42	http://www.
Kluger Krugman	Lessig Blog	bad law	2003-09-03 4:48	http://www.
captured characters from our (very long ago) past	Lessig Blog	free culture	2003-08-30 2:21	http://www.
The EU fight against yuck ePatents	Lessig Blog	bad law	2003-08-30 0:47	http://www.

Lawrence Lessig (email)
Professor of Law at Stanford Law School
Founder of the Stanford Center for Internet and Society
Author of The Future of Ideas and Code and Other Laws of Cyberspace
Chair of the Creative Commons project
Lessig News RSS 2.0 Feed [RSS 1.0], Lessig Blog RSS 2.0 Feed [RSS 1.0]
Donate to the Creative Commons project via Amazon's Honor System or PayPal:

Lessig Blog/free culture
captured characters from our (very long ago) past the changing tune of the reco

画像 http://www.lessig.org/images/b_blog.jpg をダウンロードしています...

Fig. 6. Snapshot of Glucose

Glucose can extract TrackBack links from each conten t. Obtained links are shown below the corresponding entry in headline pane with "Re:..." like a mailer.

With Weblog editor in terface (Figure 7) in Glucose, the user can post an en try to her/his Weblog if she/he has strong interest for conten t. This interface uses XML-RPC protocol. The user can clip contents to the clip list of own RNA using XML-RPC, then clipped conten ts are published via RNA.

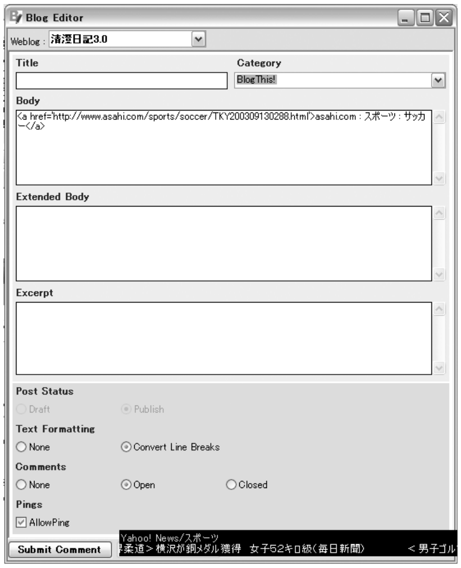
4.3 Double-loop Gratification with Semblog

In our system, instant gratification is realized b y the basic functions of RNA and Glucose. People simply read RSS-based conten ts from various information sources with our aggregators. These functions can make benefit to individual users in reading and writing Weblog contents.

Clipping is one of instance of translucence strategy in Semblog. For herself/himself, clipped content works as reminder that means "what I was thinking about?" instantly. On the other hand, someone who bro wses her/his clipscan understand "what she/he was interested in?" because all clips are published on the Web.

Our Semblog system can be used as an information sharing platform. It is based on simple metadata so that it can be extended easily. W e develop a new type of recommendation and retrieval systems to support delay ed gratification as follows.

F OAF T rackBack Each RNA has XML-RPC interface that can send and receiv e its data dynamically RNA alliance is a conten t recommendation system based on cooperation of multiple RNAs.


Blog Editor

Weblog : 清澄日記3.0

Title

Category: BlogThis!

Body:
```
<a href='http://www.asahi.com/sports/soccer/TKY200309130288.html'>asahi.com : スポーツ : サッカー</a>
```

Extended Body

Excerpt

Post Status: ○ Draft ● Publish

Text Formatting: ○ None ● Convert Line Breaks

Comments: ○ None ● Open ○ Closed

Pings: ☑ AllowPing

Yahoo! News/スポーツ
柔道＞横沢が銅メダル獲得　女子52キロ級（毎日新聞）　＜男子ゴル

Fig. 7. Weblog Editor

We use FOAF metadata to identify each RNA. FOAF is RDF-based metadata format for describing human relationship. Besides the basic elements such as name, email and URI of the user, FOAF provides a statement that user X knows user Y.

The current version of RNA can generate FOAF data. RNA also has an interface for FOAF management to extend social network easily. We call this method as "FOAF TrackBack".

First the user X enters an RNA URI of the user Y in her/his own FOAF manager. The manager X asks the manager Y to acquire the FOAF data of Y, and writes "X knows Y" link in its FOAF. The manager Y records "Y isKnown by X" link in its FOAF and notifies to the user Y. If the user Y agrees, her/his manager registers "Y knows X" link. Repeating this process, a personal network of the user is constructed. Following recommendation methods are performed in the network.

RNA Alliance RNA Alliance is collaborative recommendation based on difference of registered sites or clips among multiple RNAs. At first it calculates similarities: S_i between the user's RNA:R_0 and a RNA on the personal network: R_i $(1 < i < n)$. Each RNA has a list of URIs: $R_i = \{u_0, \ldots, u_k\}$.

$$S_i = \frac{|R_0 \cap R_i|}{|R_0| + |R_i|}$$

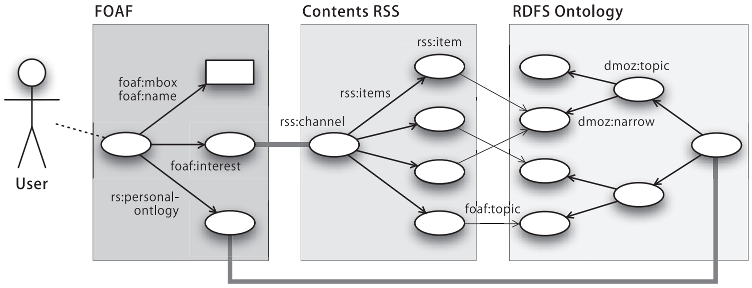


Fig. 8. Personal Ontology Framework

The system gives recommendation score: $V(u)$ to each URI by the following formula:

$$V_i(u) = \begin{cases} S_i & \text{if } u \in R_i \\ 0 & \text{if } u \notin R_i \ (i = 1, \ldots, n) \end{cases}$$

$$V(u) = \frac{\sum_{i=i}^{n} V_i(u)}{n}$$

This score is used for recommendation to R_0's user if URI u is not included in R_0.

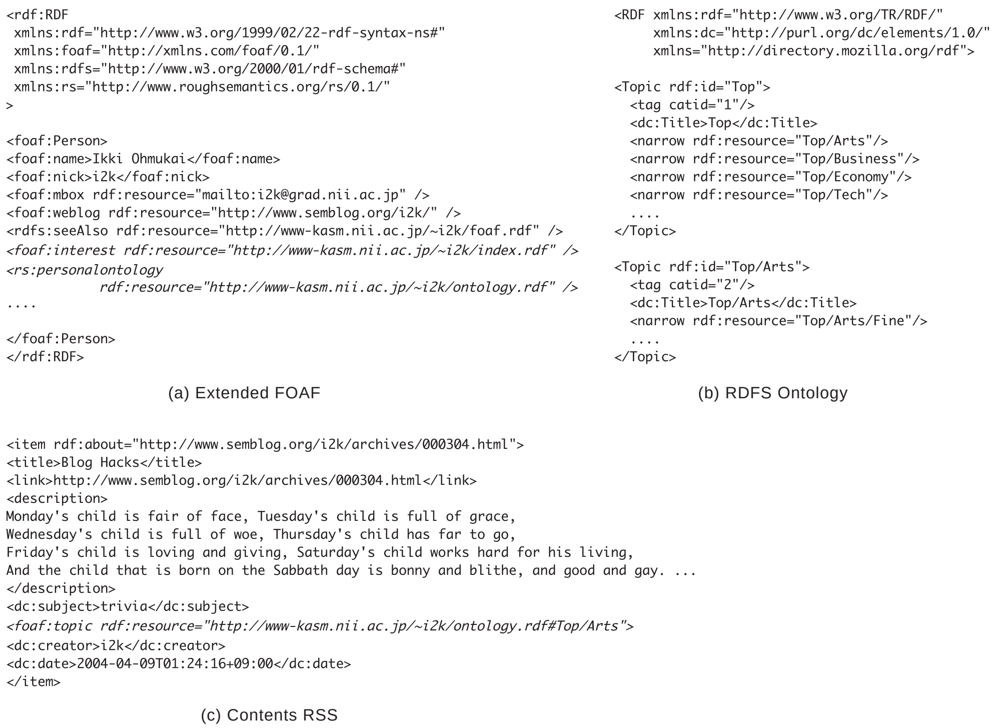
The system shows the list of recommended URIs sorted by the score. The user can add these URI to her/his own "check" list.

Personal Ontology We propose a bottom-up personal ontology framework using RSS and FOAF metadata. To process small contents in various forms, we have to annotate a semantic markup with an ontology language to those contents. It is difficult to organize practical ontology hierarchy with top-down approach because building and maintaining such well-organized large ontology takes a lot of efforts. We aim to develop loose and bottom-up ontology system by combining personal classification, because we consider that personal knowledge will be represented with a routine work such as categorization and arrangement of information. Figure 8 indicates a conceptual architecture of the personal ontology system.

At first we define a personal ontology as a hierarchical system of categories. Everyone has those categories, and they routinely classify described and collected contents to the categoryA label of a category can be named arbitrarily by user.

Unlike the conventional ontology, the personal ontology has to be related to the person who produces it. Therefore we apply FOAF metadata to link between the ontology and the person.

Personal ontology metadata consists of FOAF, RDFS Ontology and Contents RSS. The FOAF describes personal information, and the RDFS ontology shows



```
<rdf:RDF
 xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xmlns:foaf="http://xmlns.com/foaf/0.1/"
 xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
 xmlns:rs="http://www.roughsemantics.org/rs/0.1/"
>

<foaf:Person>
<foaf:name>Ikki Ohmukai</foaf:name>
<foaf:nick>i2k</foaf:nick>
<foaf:mbox rdf:resource="mailto:i2k@grad.nii.ac.jp" />
<foaf:weblog rdf:resource="http://www.semblog.org/i2k/" />
<rdfs:seeAlso rdf:resource="http://www-kasm.nii.ac.jp/~i2k/foaf.rdf" />
<foaf:interest rdf:resource="http://www-kasm.nii.ac.jp/~i2k/index.rdf" />
<rs:personalontology
            rdf:resource="http://www-kasm.nii.ac.jp/~i2k/ontology.rdf" />
....

</foaf:Person>
</rdf:RDF>
```

(a) Extended FOAF

```
<RDF xmlns:rdf="http://www.w3.org/TR/RDF/"
     xmlns:dc="http://purl.org/dc/elements/1.0/"
     xmlns="http://directory.mozilla.org/rdf">

<Topic rdf:id="Top">
  <tag catid="1"/>
  <dc:Title>Top</dc:Title>
  <narrow rdf:resource="Top/Arts"/>
  <narrow rdf:resource="Top/Business"/>
  <narrow rdf:resource="Top/Economy"/>
  <narrow rdf:resource="Top/Tech"/>
  ....
</Topic>

<Topic rdf:id="Top/Arts">
  <tag catid="2"/>
  <dc:Title>Top/Arts</dc:Title>
  <narrow rdf:resource="Top/Arts/Fine"/>
  ....
</Topic>
```

(b) RDFS Ontology

```
<item rdf:about="http://www.semblog.org/i2k/archives/000304.html">
<title>Blog Hacks</title>
<link>http://www.semblog.org/i2k/archives/000304.html</link>
<description>
Monday's child is fair of face, Tuesday's child is full of grace,
Wednesday's child is full of woe, Thursday's child has far to go,
Friday's child is loving and giving, Saturday's child works hard for his living,
And the child that is born on the Sabbath day is bonny and blithe, and good and gay. ...
</description>
<dc:subject>trivia</dc:subject>
<foaf:topic rdf:resource="http://www-kasm.nii.ac.jp/~i2k/ontology.rdf#Top/Arts">
<dc:creator>i2k</dc:creator>
<dc:date>2004-04-09T01:24:16+09:00</dc:date>
</item>
```

(c) Contents RSS

Fig. 9. Personal Ontology Metadata

a structure of the categories, and the contents RSS shows written and collected contents by the user.

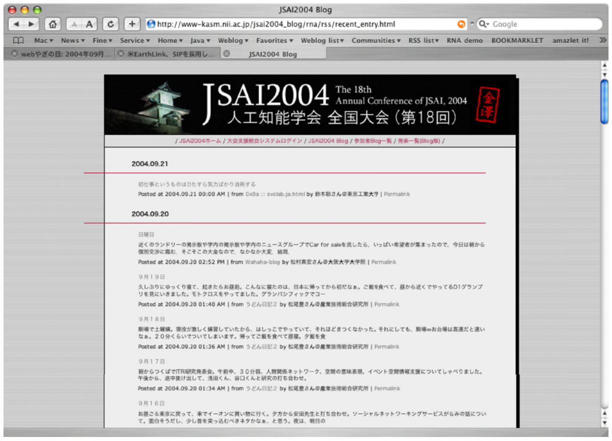
We add two elements to basic FOAF model shown in Figure 9 (a). One is `<foaf:interest>` which is to point the contents RSS, and the other is `<rs:personalontology>` that is originally defined by our Rough Semantics project [6] to indicate the RDFS ontology .

The RDFS ontology is described with the form of Open Directory RDFS format shown in Figure 9 (b). Each node has a fragment ID.

The contents RSS is similar to a conventional RSS. Our RSS uses `<foaf:topic>` to point a category on the RDFS ontology , while the conventional model applies `<dc:subject>` to express a thesis of a content. This makes our RSS to have backward compatibility. Example of this RSS is shown in Figure 9 (c). It should be noted that topics pointed by this tag are not restricted to those in their own ontology , rather any topics in others' and some global ontology . Separating ontology and instances enables such flexible management.

FOAF, RDFS ontology and RSS are described in separate files so that we can keep compatibility with existing applications on these formats. This is a

[6] http://www.roughsemantics.org/


JSAI2004 Blog
http://www-kasm.nii.ac.jp/jsai2004_blog/rna/rss/recent_entry.html
Google
Mac ▾ News ▾ Fine ▾ Service ▾ Home ▾ Java ▾ Weblog ▾ Favorites ▾ Weblog list ▾ Communities ▾ RSS list ▾ RNA demo BOOKMARKLET amazlet it!
webやぎの目: 2004年09月...
米EarthLink、SIPを採用し...
JSAI2004 Blog
JSAI2004 The 18th Annual Conference of JSAI, 2004
人工知能学会 全国大会（第18回）
金澤
/ JSAI2004ホーム / 大会支援統合システムログイン / JSAI2004 Blog / 参加者Blog一覧 / 発表一覧(Blog版) /

2004.09.21

初仕事というものはひたすら気力ばかり消耗する

Posted at 2004.09.21 00:00 AM | from 0x0a :: svslab.ja.html by 鈴木聡さん@東京工業大学 | Permalink

2004.09.20

日曜日

近くのランドリーの掲示板や学内の掲示板や学内のニュースグループでCar for saleを流したら、いっぱい希望者が集まったので、今日は朝から個別交渉に臨む。そこそこの大金なので、なかなか大変。結局、

Posted at 2004.09.20 02:52 PM | from Wahaha-blog by 松村真宏さん@大阪大学大学院 | Permalink

９月１９日

久しぶりにゆっくり寝て、起きたらお昼前。こんなに寝たのは、日本に帰ってから初だなぁ。ご飯を食べて、昼から近くでやってるD1グランプリを見にいきました。モトクロスをやってました。グランパシフィックでコー

Posted at 2004.09.20 01:40 AM | from うどん日記２ by 松尾豊さん@産業技術総合研究所 | Permalink

９月１８日

駒場で土曜練。現役が激しく練習していたから、はしっこでやっていて、それほどきつくなかった。それにしても、駒場⇔お台場は高速だと速いなぁ。２０分くらいでついてしまいます。帰ってご飯を食べて昼寝。夕飯を食

Posted at 2004.09.20 01:36 AM | from うどん日記２ by 松尾豊さん@産業技術総合研究所 | Permalink

９月１７日

朝からつくばでITR研究発表会。午前中、３０分弱、人間関係ネットワーク、空間の意味表現、イベント空間情報支援についてしゃべりました。午後から、途中抜け出して、浅田くん、谷口くんと研究の打ち合わせ。

Posted at 2004.09.20 01:34 AM | from うどん日記２ by 松尾豊さん@産業技術総合研究所 | Permalink

９月１６日

お昼ごろ東京に戻って、車でイーオンに買い物に行く。夕方から安田先生と打ち合わせ。ソーシャルネットワーキングサービスがらみの話について。面白そうだし、少し首を突っ込むべきネタかなぁ、と思う。夜は、明日の

Fig. 10. RNA in an academic conference

great benefit that our system can cope with such existing applications via these files.

Our framework enables applications and services to produce new types of searc h or recommendation. For example, mapping methods betw een t wo directories or bookmarks are applicable to the personal ontology . Egocentric searc h[6] is also able to realize easily by building a social network with `<foaf:knows>` in the users' FO AF.

Unlike these peer-to-peer model, we can calculate a similarity among a personal ontology and the global ontologies such like WordNet and ODP in advance. Multiple personal ontology can be matched with each other via the global ontology and this method needs less computation cost. In addition, it is not necessary to modify that algorithm in P2P model and personal-global model because both on tology has the same structure.

4.4 Use Case

We applied our system to some communities.

One is for an academic conference called JSAI2004 (Figure 10). Participants registered URI of her/his Weblog to RNA so that other attendees can browse various opinion for the conference and papers. Unlike con v en tional closed system, RNA pro vides that an author of an opinion keeps her/his authorship permanently.

Other example is education support. Senshu University developed class support system based on RNA. In this system, all students and teaching staff should have Weblog and all conten ts will be aggregated with each class or project respectively. The user post her/his conten t using original editor interface which communicates multiple Weblog tools and RNA. RNA aggregates and shows recently updated contents of member so that the user can access newest topics in the class and project in the university.

RNA is used as person-based contents management system. Research institute of economy, trade and industry (RIETI)[7] publishes Weblog of its research associates with RNA. Official contents should be managed in single policy but it may restrict their contribution since it is so messy to follow. On the other hand, it does not seem to official contents when each member just publishes her/his Weblog freely.

Thus RIETI introduces RNA to aggregate all contents from their Weblog and embed composite contents into official Web site. This model may decreases management cost.

We distribute RNA and Glucose in our web site[8]. About 3,000 users downloaded RNA and over 150,000 users downloaded Glucose from September 2003.

5 Discussion

We have seen a lot of failure of information sharing. But some exceptionally seem successful.

In one sense, one of the most successful information sharing is Amazon.com[9]. In Amazon.com, users' actions searching products are stored without any extra operations and used to recommend products similar to current actions. It is passive but powerful information sharing since users' actions are shared and used.

Another new trend for information sharing is *social tagging* like del.icio.us[10] and flickr[11]. We can say that it is another kind of bookmark sharing that could not be so popular. The difference is that it is more convenient in making and posting bookmarks and more powerful in using collected information.

They indicate that improving instant gratification is effective to involve people. On the other hand, improving delayed gratification is not easy to realize. One of the reason is that delayed gratification varies in communities. For example, density of information sharing also varies. Some community welcomes distributed style of information sharing like weblog with TrackBack and comments, and some embraces centered style information sharing like wiki. We should develop information sharing systems that fit types of information sharing they need.

We designed Semblog systems to allow users to select levels of information sharing in order to deal with variety of information sharing. We provide various levels of information sharing, i.e., posting clips, FOAF relationship, and personal-network based recommendation. We expect that users can find their appropriate level of information sharing by using Semblog systems.

[7] http://www.rieti.go.jp/en/index.html
[8] http://www.semblog.org/wiki/?en
[9] http://www.amazon.com
[10] http://del.icio.us/
[11] http://www.flickr.com/

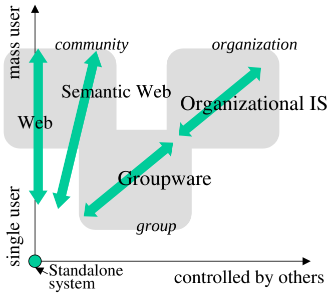


Fig. 11. Systems for community, group and organization

6 Related Work

Groupware has faced the similar problem from its beginning since acceptance by people is crucial. Grudin[7] summarized the eight challenges for Groupware, including "disparity in work and benefit", "critical mass and Prisoner's dilemma problems", and "disruption of social processes". Some of them are also challenges for Semantic Web.

The difference between Groupware and Semantic Web is ballance between control and mass of users (see Figure 11). Grudin placed Groupware between standalone systems and organizational information systems like management information systems. Standalone systems are single-user and no control by other people, while organizational information systems are multi-user and controlled by authority or other people. Groupware is middle-scale in mass of users and partially controlled. Web is, on the other hand, on the different line. Web is large-scale in mass of users but no control by authority or other people. The difference is what kind of people is target for systems. Web is used in community in which people are loosely connected and just share interest, while Groupware and organizational information systems aim to support group and organization in which people are tightly connected and share common goal.

Semantic Web is slightly controlled because maintenance of metadata is needed, but still by far free from control in comparison with Groupware. Since such difference and similarity exists between Groupware and Semantic Web, some solutions for Groupware are applicable but others not.

We can pick up some lessons from Groupware research. One is supporting of social awareness that tells people what other people do vaguely. The early example of realization of social awareness is Babble system that shows interaction among people by location of points and circles. Another example is "Gleams of People"[8] that illustrates how people are active or communicate each other by blinking of balls. This approach can be used to realize translucence strategy since it may make people aware of importance of participation to community gradually.

7 Summary

In this paper, we discuss how we can build Semantic Web applications appealing to ordinary people. We showed that two types of gratification should be needed, i.e., instant gratification that can be obtained even without information/knowledge sharing, and delayed gratification that can be obtained through information/knowledge sharing. The gap between two types of gratification can be bridged by the *translucence strategy* that lures people into information/knowledge sharing by showing delayed gratification within kissing distance. We also showed that the above methodology seems to work in building our applications.

I focus on information sharing rather than knowledge sharing because even information sharing is still a difficult task. More discussion is needed especially for delayed gratification that people can receive only after they are involved in rich knowledge sharing systems. I think that there is no royal road, but at least ballance between formality and familiarity in knowledge representation is crucial to realize widely acceptable knowledge sharing systems.

References

1. McDowell, L., Etzioni, O., Gribble, S.D., Halevy, A.Y., Levy, H.M., Pentney, W., Verma, D., Vlasseva, S.: Mangrove: Enticing ordinary people onto the semantic web via instant gratification. In: International Semantic Web Conference. (2003) 754–770
2. Uematsu, H., Numa, K., Tokunaga, T., Ohmukai, I., Takeda, H.: Balog: Location-based information aggregation system. In: Poster Proceedings of Third International Semantic Web Conference (ISWC2004). (2004)
3. Ohmukai, I., Takeda, H.: Collaborative task scheduling method based on social network analysis for cellphone application. In: Proceedings of the IADIS International Conference of WWW/Internet (ICWI2004), Madrid, Spain (2004)
4. Hamsaki, M., Takeda, H., Ohmukai, I., Ichise, R.: Scheduling support system for academic conferences based on interpersonal networks. In: Poster, HyperText 2004. (2004)
5. Ohmukai, I., Takeda, H., Hamasaki, M., Numa, K., Adachi, S.: Metadata-driven personal knowledge publishing. In McIlraith, S.A., Plexousakis, D., van Harmelen, F., eds.: The Semantic Web - ISWC 2004: Third International Semantic Web Conference, Hiroshima, Japan, November 7-11, 2004. Volume 3298 of Lecture Notes in Computer Science (LNCS). (2004) 591–604
6. Numa, K., Ohmukai, I., Hamasaki, M., Takeda, H.: Egocentric search based on RSS. In: Poster Proceedings of Third International Semantic Web Conference (ISWC2004). (2004)
7. Grudin, J.: Groupware and social dynamics: eight challenges for developers. Communications of ACM **37** (1994) 92–105
8. Ohguro, T., Yoshida, S., Kuwabara, K.: Gleams of people: Monitoring the presence of people with multi-agent architecture. In: Approaches to Intelligent Agents (PRIMA'99 proceedings). Volume 1733 of Lecture Nots for Artificial Intelligence. Springer-Verlag (1999) 170–182

Semantic Navigation with VIeW*s*

Paul Buitelaar, Thomas Eigner, Stefania Racioppa

DFKI GmbH, Language Technology Lab
Stuhlsatzenhausweg 3
66123 Saarbruecken, Germany

paulb@dfki.de

The paper describes VIeWs, a system that combines ontologies, web-based information extraction, and automatic hyperlinking to enrich web documents with additional relevant background information. The central idea behind VIeWs is to demonstrate how web portals can be dynamically tailored to special interest groups by use of corresponding ontologies. As a particular use case we developed an application for the "saarland.de" web portal of the Saarland region in Germany, which we present here in some detail. The paper describes the ideas behind the system and the Saarland.de application and provides an overview of the system architecture and components. Additionally, next to a comparison with related work, also some discussion on end user aspects of the application and its connection to the Semantic Web is given. It is argued that VIeWs is a typical end user application that depends on ontologies as semantic models for different scenarios, but that the need for Semantic Web technology beyond this has not been proven yet.

1 Introduction

The central idea behind VieWs is to demonstrate how web portals can be dynamically tailored to special interest groups by use of corresponding ontologies. For this purpose, the VieWs system combines ontologies, web-based information extraction, and automatic hyperlinking to enrich web documents with additional relevant background information, relative to particular ontologies selected by individual users.

The automatically generated hyperlinks are based on specific ontological "views" on the web portal information, which allow for a high level definition of specific interest topics. As a particular use case we developed an application for the "saarland.de" web portal of the Saarland region in Germany, which we present here in some detail.

The paper is organized as follows: first in section 2 an overview of the VieWs saarland.de application will be described, followed by a brief description of the system architecture and individual components in section 3, and in section 4 by a discussion of end-user issues of the application described here as well as of related work.

2 VieW*s* on saarland.de

The “saarland.de” web portal[1] provides general information on events concerning the local government and institutions. Additionally, sub-sections of the portal include information on various broader topics, such as tourism (“tourismus.saarland.de”), business (“wirtschaft.saarland.de”), etc.

The VIeWs saarland.de application automatically provides users with additional information that is specific to their interests (e.g. hotel information with indication of price and location for tourists or information on the city council, representations of political parties or similar for a local citizen) as derived from the saarland.de portal itself (interlinking portal web pages) or from the web in general (interlinking portal web pages with external information).

2.1 Scenarios

Two application scenarios have been defined and represented in ontologies reflecting the profiles of user groups that correspond to these scenarios:

The **Tourism** scenario reflects a visit of the saarland.de web portal by someone who is interested in tourism options of the Saarland region. The “Tourist” will be interested to know about hotels, restaurants in any city mentioned on the pages of the web portal. In the Tourism ontology this ‘view’ on saarland.de has been defined as follows: a city has Cultural Institutes (Theatre, Cinema), Accommodations (Hotel, “Gasthof”), and Gastronomy (Restaurant, “Konditorei”). These topics are defined in the ontology as classes that are connected over attributes with the class “Stadt” (City). Additionally, every class has attributes such as Location, Number of Rooms, Name, Address and Homepage for the Accommodations class and its subclasses (Hotel, “Gasthof”).

The **Administration (“Verwaltung”)** scenario reflects a visit of the saarland.de web portal by a local citizen who knows the cities in the region but may be interested in specifics, such as administrative offices, political parties, etc. In the Administration ontology this ‘view’ on saarland.de has been defined as follows: a city has a City Administration, Organizations (Political Party, “Wirtschaftsverband”), and Council Offices (“Arbeitsagentur”, ”Standesamt”) In the ontology these topics again are defined as classes that are connected with the class “Stadt”. Additionally, every class has attributes such as Name, Address and Homepage for the Organizations class and its subclasses (Political Party, “Wirtschaftsverband”).

[1] http://www.saarland.de (see http://www.english.saarland.de/ for an English version - only partial)

2.2 Demonstrator

VieWs is a server side application that can be used with a standard web browser[2], which makes it transparent to the normal web user. The user simply browses the saarland.de web portal as normal, but is now being supported by the VieWs system that adds additional information on the basis of a web-based search and from an automatically extracted knowledge base and shows this over generated hyperlink structures. The user can simply decide to follow the regular links or the generated links with added information.

The new links include information from within the saarland.de domain, or also from outside. Depending on the application scenario, this should be set by the user or could be fixed by the system administrator. For instance, in the case of tourism it does make sense to include also external web sites, e.g. hotel home pages. On the other hand, in the case of information for the citizen it may be better to include only 'controlled' information, i.e. only web pages from within the saarland.de domain. The current demonstrator leaves this decision up to the user.

The VieWs entry page[3] for saarland.de enables the user to select their specific interest, currently either "Tourismus" (Tourism) or "Verwaltung" (Administration). By selecting a preference the user automatically enters the VieWs system. From this point on all navigation will be supported by the system according to the selected ontology and, dependent on how the user entered (as a tourist or as a citizen), identified city names will be hyperlinked with additional tourist- or administration-related topics and web-based information.

2.3 VIeWs on Tourism in saarland.de

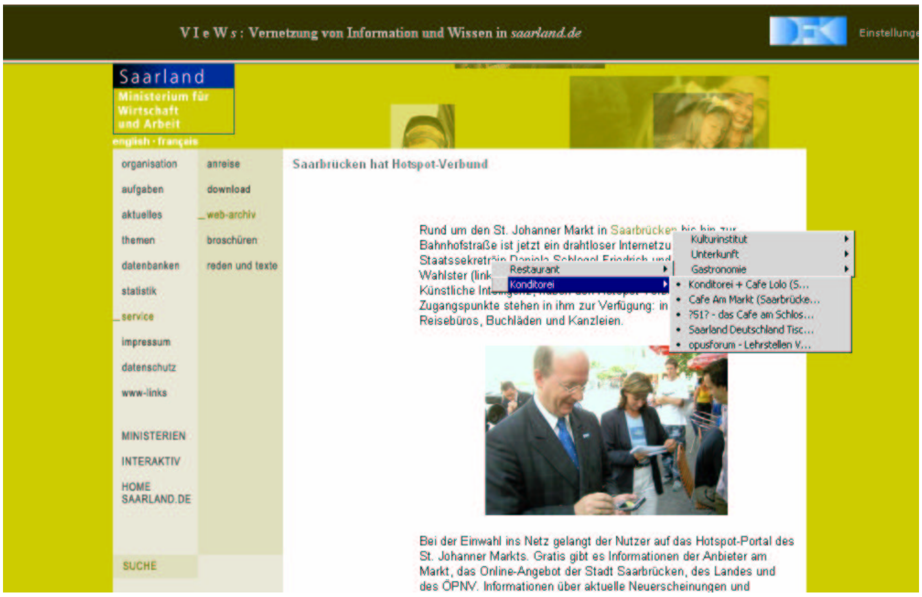
For example, if the user entered as a "Tourist", as shown in Figure 1 below, the generated hyperlink structure shows web links to accommodation (e.g. hotels), dinner options (e.g. restaurants) and cultural institutions (e.g. cinema, theatre) for each identified city name (of the Saarland region) on the page. The added information is included through a Google-based web search for each recognized city name in combination with keywords ("Hotel", "Restaurant", etc.) derived from the ontology class label names.

For selected classes (e.g. hotels) additional information (e.g. address, indication of size, location) is added as shown in Figure 2. This additional information has been previously extracted from retrieved web pages. For this purpose each time a web search has been executed, all retrieved URLs are checked for existence in the knowledge base. If the URL is not in the knowledge base, it will be send to the information extraction component for further extraction of relevant, class-specific information.

The hyperlink structure is generated out of the corresponding ontology, i.e. from the underlying RDF/S file. Over a separate window this structure can be inspected by the user as shown in Figure 3.

[2] The demonstrator has been optimized for Internet Explorer 6.x.

[3] http://views.dfki.de

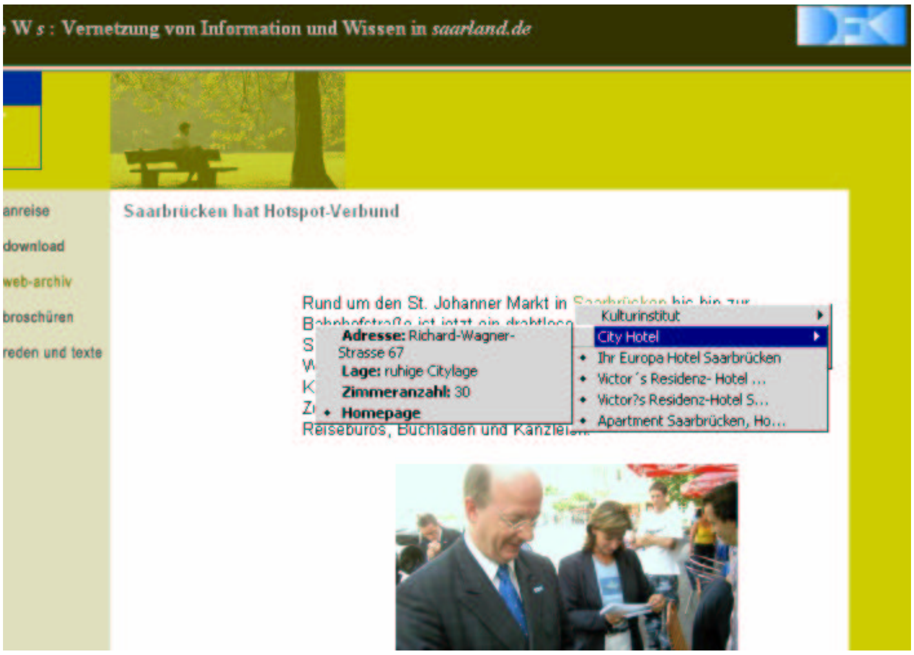


Saarbrücken hat Hotspot-Verbund

Rund um den St. Johanner Markt in Saarbrücken bis hin zur
Bahnhofstraße ist jetzt ein drahtloser Internetzu
Staatssekreträin Daniela Schlegel-Friedrich und
Wahlster (link
Künstliche Int
Zugangspunkte stehen in ihm zur Verfügung: in
Reisebüros, Buchläden und Kanzleien.

Bei der Einwahl ins Netz gelangt der Nutzer auf das Hotspot-Portal des St. Johanner Markts. Gratis gibt es Informationen der Anbieter am Markt, das Online-Angebot der Stadt Saarbrücken, des Landes und des ÖPNV. Informationen über aktuelle Neuerscheinungen und

Figure 1: VIeWs with the Tourism Ontology


W s : Vernetzung von Information und Wissen in saarland.de
anreise
download
web-archiv
broschüren
reden und texte
Kulturinstitut
City Hotel
Ihr Europa Hotel Saarbrücken
Victor´s Residenz- Hotel ...
Victor?s Residenz-Hotel S...
Apartment Saarbrücken, Ho...
Homepage

Saarbrücken hat Hotspot-Verbund

Rund um den St. Johanner Markt in Saarbrücken bis hin zur
B
S
W
K
Z
Reisebüros, Buchläden und Kanzleien.

Adresse: Richard-Wagner-Strasse 67
Lage: ruhige Citylage
Zimmeranzahl: 30

Figure 2: Detailed information on hotels from the Knowledge Base

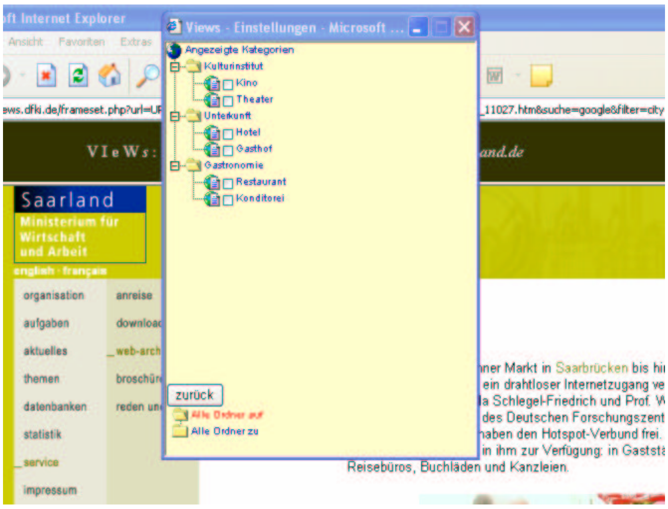

ft Internet Explorer
Ansicht
Favoriten
Extras
ews.dfki.de/frameset.php?url=UF
_11027.htm&suche=google&filter=city
VIeWs:
and.de
Views - Einstellungen - Microsoft ...
Angezeigte Kategorien
Kulturinstitut
□ Kino
□ Theater
Unterkunft
□ Hotel
□ Gasthof
Gastronomie
□ Restaurant
□ Konditorei
zurück
Alle Ordner auf
Alle Ordner zu
Saarland
Ministerium für Wirtschaft und Arbeit
english · français
organisation
aufgaben
aktuelles
themen
datenbanken
statistik
service
impressum
anreise
download
web-arch
broschür
reden un
ner Markt in Saarbrücken bis hi
ein drahtloser Internetzugang ve
la Schlegel-Friedrich und Prof. W
des Deutschen Forschungszent
haben den Hotspot-Verbund frei.
in ihm zur Verfügung: in Gaststä
Reisebüros, Buchläden und Kanzleien.

Figure 3: User interaction with the Tourism Ontology

3 The VIeW*s* System

VieWs is implemented as a web-based system and consists of several components as shown in Figure 4 below. The user activates the system over the VieWs web interface as discussed in section 2. The accessed web page is processed by extracting text segments and sending these to the named-entity recognition component for the identification and markup of relevant hyperlink anchors (e.g. city names). For each combination of city name and keyword (“hotel”, “restaurant”, etc.) derived from the ontology, a Google-based web search is started. The results of the web search and information already in the knowledge base is shown in the form of generated hyperlink menus on each of the identified city names. Additionally, an information extraction process is started in the background over the retrieved documents to extract additional relevant information that will be stored in the knowledge base for future access.

The online part of the VIeWs system is written entirely in Java and consists of a hyperlinking component (for generating hyperlink menus in JavaScript), the Google API (for web search with ontology-based keywords), a web service interface with the named-entity recognition component, a database connection with the knowledge base and a crawling component (for downloading the web pages that were retrieved by the web Google API).

The offline part of VIeWs consists of an independently developed information extraction system (the same as used for the online named-entity recognition) and the knowledge base.

The ontologies are an additional static resource that are used online (in building up the hyperlinking menus) and offline (in information extraction).

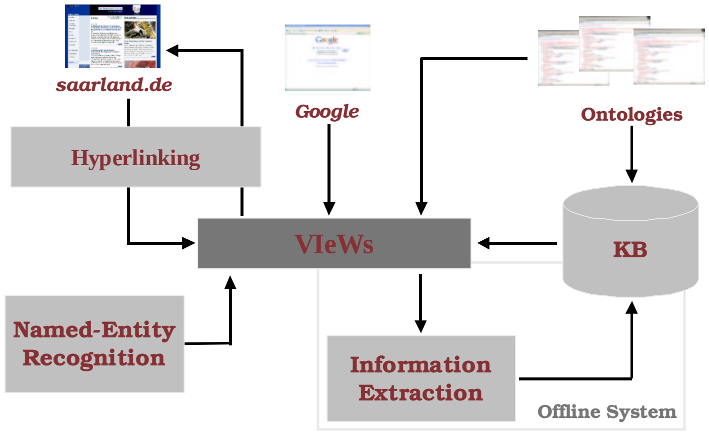


Figure 4: VIeW*s* System Overview

3.1 Hyperlinking

The hyperlinking component takes the accessed web page and regenerates it with the addition of JavaScript hyperlink menus for all identified anchors. The hyperlink structure shows the five best results from Google for each ontology-based keyword (i.e. ontology class name) with stored facts if available.

In this process the following information is integrated:

- Identified hyperlink anchors – named-entity recognition with SProUT
- Ontology structure – ontologies are parsed with Jena
- Results of web search with Google – accessed with Google API
- Stored facts from the knowledge base

3.2 Named-Entity Recognition

The named-entity component is based on SProUT[4], a type-driven information extraction tool that was developed at DFKI [Drozdzynski et al., 2004]. Anchors, e.g. city names, are recognized on the basis of gazetteers and extraction rules over shallow linguistic information (part-of-speech, morphological analysis). A rule in SproUT consists of a regular expression over typed feature structures representing the recognition pattern, and a typed feature structure on the right-hand side that specifies positions and attributes of identified entities in an XML format (see also [Busemann et al., 2003]).

[4] More information on SProUT is available at http://sprout.dfki.de/

3.3 Ontologies

Ontologies are defined using Protégé with export in RDF/S, which is accessed and processed by the VIeWs system to generate a corresponding hyperlink menu in Javascript. As described in section 2 above, each ontology defines a particular user scenario that is organized around a central object class (e.g. cities), over which more specific information objects are defined (e.g. city institutes or organizations). The information structure that is defined in an ontology also guides the information extraction process for filling out the corresponding knowledge base (see also below).

3.4 Web Search

The VieWs system is a hyperlinking application that integrates information on one web page with information from other web pages. For this purpose, a web crawler is included that searches for relevant web pages, given a set of keywords that can be derived from the ontology. The web crawler that we currently use is the Google API, but as it is rather slow and not always reliable in terms of precision we are considering the integration of other search engines (such as Yahoo) or the implementation of a dedicated crawler for the Saarland region.

3.5 Information Extraction

The information extraction component is also based on SProUT and is used offline to derive class-specific information from web pages. For instance, the address, location description (e.g. “central”, “no traffic”, “near railway station”) or the number of rooms for a hotel could be extracted from the hotel home page. The extracted information is stored in the knowledge base and accessed if the corresponding URL of the web page has been retrieved by the web search component. In this way, stored information is only shown if the corresponding web page is still regarded as ‘relevant’ by the web search component (i.e. Google currently).

3.6 Adapting VIeW*s* to Other Domains and Applications

The VieWs system has been designed to be adaptable to other scenarios, either within the saarland.de application or in a completely new application context[5]. For this purpose the following components should be adapted: an ontology should be defined for the new scenario; a corresponding information extraction grammar should be defined; additionally, if the ontology is defined around a different central object class (i.e. different from “cities” in the current implementation) then also the named-entity recognition component should be adapted accordingly.

[5] For instance, we are currently working on an application of VieWs for http://www.dfki.de

4 End User Issues and the Semantic Web

As shown by the examples in this paper, the VieWs system is a typical end user application, in which any level of technological complexity should be kept fully transparent. In this respect it is also irrelevant if the technology used in VieWs is based on Semantic Web technology or not. The main goal is to satisfy user needs in accessing relevant information at the right moment and in the right context.

Nevertheless, exactly this context is the central aspect of the VieWs application that can be expressed by use of available Semantic Web standards and technology. The user context, i.e. a user group profile such as “those web portal visitors interested in tourism”, can be captured in an ontology defined for instance in RDF/S or OWL. Extracted information can be stored in and accessed from a corresponding knowledge base that can be based on Semantic Web technology, such as SESAME, Jena, etc. Reasoning facilities can then also be easily added to the application, e.g. to integrate class-specific semantic web services [Dzbor et al., 2004] or to derive further knowledge by use of rules or axioms.

On the other hand, it is also true that VIeWs in its current form can be implemented without a complete use of Semantic Web standards and tools. Relational databases and other standard technology are equally capable of providing the current functionality of the VIeWs application. Although reasoning capabilities cannot be offered, the use case for these has not been established yet. At the same time, web-based search is central to VIeWs which obviously is also not Semantic Web based.

In summary, semantic context models such as user profiles and associated knowledge bases seem to provide an application scenario for Semantic Web standards and technologies in the VIeWs context, but the use case for this needs still to be proven.

5 Related Work

Related work to VIeWs exists in various respects, i.e. on the level of semantic-based indexing and hyperlinking (e.g. [Pustejovsky et al., 1997], [Carr et al., 2001], [Dill et al., 2003]), information extraction and hyperlinking (e.g. [Busemann et al., 2003], [Popov et al., 2003], [Basili et al., 2004]), and ontologies as user models - in hyperlinking (e.g. [Maedche et al., 2002]).

In general however, VIeWs is most similar to Magpie [Dzbor et al., 2003] although it seems also complementary in some respect. In particular, VieWs integrates an online web search functionality, which makes it very flexible in the kind of information it is able to show. Magpie on the other hand has access only to an underlying static knowledge base. Secondly, VieWs includes an information extraction component that is fully integrated in the automatic processing of retrieved web pages and knowledge base extension and updating. It is not clear if information extraction has been similarly completely integrated with Magpie. Finally, VIeWs can handle most web page formats and seems therefore more robust in real-life applications than Magpie.

6 Conclusions

We presented the VieWs system and its application in the context of the saarland.de web portal. The system consists of clearly defined and efficiently integrated components for web search, information extraction and hyperlinking and has been designed in such a way that it can be readily adapted to other application scenarios and domains.

VieWs can be seen as a Semantic Web application as it uses related standards such as RDF/S and tools such as Jena. On the other hand, the core functionality of Semantic Web applications, reasoning and inference, has not been integrated as the use case for this functionality has not been proven yet. In future work, we will concentrate on identifying the use case for reasoning and inference in the context of real-life applications, such as the saarland.de scenarios described here.

Acknowledgements

This research has been supported by a grant for the project VIeWs by the Saarland Ministry of Economic Affairs.

References

Roberto Basili, Maria Teresa Pazienza, Fabio Massimo Zanzotto *Inducing hyperlinking rules in text collections* In: Proceedings of RANLP2004, John Benjamins, Amsterdam/Philadelphia, 2004.

Stephan Busemann, Witold Drozdzynski Hans-Ulrich Krieger, Jakub Piskorski, Ulrich Schäfer, Hans Uszkoreit, Feiyu Xu *Integrating Information Extraction and Automatic Hyperlinking*. In Proceedings of the ACL-2003 demo session, Sapporo, Japan, 2003.

Leslie Carr, Sean Bechhofer, Carole Goble, Wendy Hall. *Conceptual Linking: Ontology-based Open Hypermedia*. WWW10, Tenth World Wide Web Conference, Hong Kong, May 2001.

Dill S., N. Eiron, D. Gibson, D. Gruhl, R. Guha, A. Jhingran, T. Kanungo, S. Rajagopalan, A. Tomkins, J. A. Tomlin, and J. Y. Zien, *SemTag and Seeker: Bootstrapping the semantic Web via automated semantic annotation*, The Twelfth International World Wide Web Conference Budapest, Hungary, 2003.

W. Drozdzynski, H.-U. Krieger, J. Piskorski, U. Schäfer, F. Xu. *Shallow Processing with Unification and Typed Feature Structures - Foundations and Applications.* In Künstliche Intelligenz, 1/2004.

M. Dzbor, J.B. Domingue, E. Motta *Magpie - towards a semantic web browser*. 2nd International Semantic Web Conference, October 2003, Florida, USA.

M. Dzbor, E. Motta, J.B. Domingue *Opening Up Magpie via Semantic Services*. Proc. of 3rd International Semantic Web Conference (ISWC04). November 2004. Japan.

A. Maedche, S. Staab, R. Studer, Y. Sure and R. Volz. *SEAL - Tying Up Information Integration and Web Site Management by Ontologies*. In: IEEE Computer Society Data Engineering Bulletin, Special issue on "Organizing and Discovering the Semantic Web", Vol. 25, No. 1, pp. 10-17, March 2002.

B. Popov, A. Kiryakov, D. Ognyanoff, D. Manov, A. Kirilov, M. Goranov *Towards Semantic Web Information Extraction.* Human Language Technologies Workshop at the 2nd International Semantic Web Conference (ISWC2003), 20 October 2003, Florida, USA.

J. Pustejovsky, B. Boguraev, M. Verhagen, P.P. Buitelaar and M. Johnston *Semantic Indexing and Typed Hyperlinking* In: Proceedings of AAAI Spring 1997 Workshop on Natural Language Processing for the World Wibe Web, Stanford University, March 1997.

Ontology Mapping with domain specific agents in the AQUA Question Answering system

Miklos Nagy, Maria Vargas-Vera and Enrico Motta

Knowledge Media Institute (KMi),
The Open University,
Walton Hall, Milton Keynes, MK7 6AA, United Kingdom
miklos.nagy@jrc.nl; {m.vargas-vera, e.motta}@open.ac.uk

Abstract. This paper describes a domain specific multi-agent ontology-mapping solution in the AQUA query answering system. In order to incorporate uncertainty inherent to the mapping process, the system uses the Dempster-Shafer model for dealing with incomplete and uncertain information produced during the mapping. A novel approach is presented how specialized agents with partial local knowledge of the particular domain achieve ontology mapping without creating global or reference ontology. Our approach is particularly fit for a query-answering scenario, where answer needs to be created in real time that satisfies the query posed by the user.

1 Introduction

An important aspect of ontology mapping is how the incomplete and uncertain results of the different similarity algorithms can be interpreted during the mapping process started to become a well-acknowledged research direction. As the latest research started moving towards a more automated mapping process it has been recognized that current approaches do not fully investigate the nature of the produced similarity information and mainly rely on a human domain expert to make a judgment about the correctness of the established mapping. However in the context of question answering like the AQUA [1,2] system the dynamic nature of the source information (e.g. web enabled databases) does not make it possible that a domain expert help is necessary every time the source changes to follow up the modifications in the existing mapping. Our novel approach to address this problem utilizes a multi agent framework where the different mapping agents possess local sub-domain specific knowledge about particular entities (e.g. material, specimen, etc.). From the end user perspective our system addresses the problem of data integration of scientific databases containing vast number of experimental Semantic Web enabled data in order to facilitate better knowledge sharing and reuse between the scientific communities. Although these databases are accessible, the seamless data exchange between different databases is still an unsolved problem in spite of the fact that different XML based languages were defined by the different scientific communities e.g. MatML(Materials Markup Language)[3] on the field of material science to facilitate a standardized XML based data exchange. This solution solved a number of interoperability issues but makes the assumption that both parties agreed the syntax of the data exchange. This assumption fails when one would search for existing experimental data available on the WWW since neither the syntax nor the semantics of the requested data is known before the submission of the query. The problem is that different research institutions, companies use different standards and naming conventions in their logical data model for the same data, additionally these data model is not always even accessible on the WWW. Hence a vast number of experimental data are remaining inaccessible, or unanalyzed that probably hides the undiscovered correlations of science disciplines. The mapping agents use the Dempster-Shafer theory of evidence [4] to assess and combine the belief in the correctness of the different similarity algorithms. Our approach also does not assume the existence of global or reference ontology that is the superset of the different source ontologies and contains the existing mappings a priory. This approach makes it possible to perform query answering effectively with multiply source ontologies. In our first experimental system we consider query answering over Web enabled S&T (Scientific and Technical) or engineering databases those are described with their own domain specific ontologies.

The paper is organized as follows:
Section 2 presents the architecture of the mapping framework and describes how mapping agents on the different levels are carrying out the mapping. Section 3 introduces the similarity algorithm used by the framework to assess syntactic and semantic similarities between the posed query and the local ontologies. Section 4 describes how the problem of uncertain information created by the similarity mapping process is resolved and handled by the mapping framework. Section 5 presents a working example. Section 6 presents implantation details. Sections 7 discuss the related work and Section 8 gives conclusions as well as the future research directions.

2 Architectural overview of the mapping framework

The high-level system architecture figure 1 shows how the functional parts of the system are related with each other. In the mediator layer the agents are organized in different levels. Agents on the broker level responsible for decomposing the query into sub queries, based on the global descriptor. The decomposed query parts are sent into the mapping agents located in the mapping layer. Mapping agents obtain the relevant information from the sources through the source agents. When only one source corresponds to the query the scenario is pretty straightforward and there is no need for any mapping between the sources, the query can be answered from the source. In a real case scenario this possibility is not so likely and this is why the mapping between local ontologies is a justified scenario in our case.

The idea that has been investigated in our research is that mapping agents can build up mappings simultaneously, utilizing different similarity measures Based on their belief agents need to harmonize their beliefs based on trust that is formed during the mapping process.

This is a two-step process:

1. Mapping agent based on evidences that is available to them built up belief about the mapping.
2. Group of mapping agents need to harmonize their beliefs over the solution space.

The key components of the prototype are grouped by the different functional levels and from bottom to up as follows.

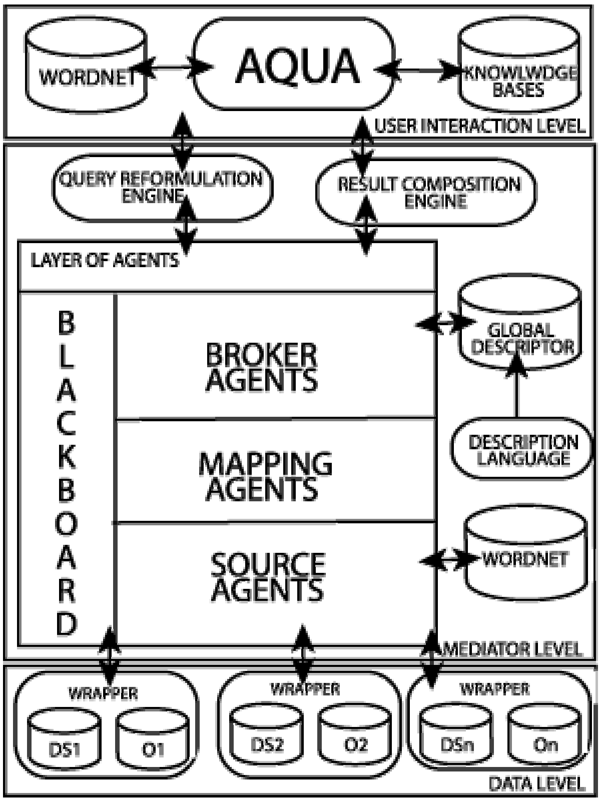


Figure 1. Architecture of the Multi-ontology mapping framework.

Data Level
On the data level the heterogeneous data sources are represented by their ontologies. The format of these sources varies from relational databases to simple files.

- Data source (DS): actual data represented in the database, file etc.
- Ontology (O) Semantic metadata that describes the particular data source.
- Wrapper creates a unified XML representation of the source that is queried by the particular resource agents.

Mediator level
Layer of agents: Typically three kind of agents: broker that receives a FOL (First Order Logic) query and decomposes it into sub queries based on the global descriptor, mapping that has knowledge of a particular domain specific area and cooperatively map up the source concept with the concepts contained by the query string, source that access a particular data source and it's ontology and passes it to the mapping agents on a request basis.
Global descriptor and description language: Key component of the system that describes what kind of information can be found in the different sources, and which agent is able to answer the query posed by the user based on the entities in the query. Practically FOL knowledge base that contains information about the agents and entities as well as the resources
Query reformulation and result composition engine: Query that is raised by the user needs to be reformulated and decomposed before entered into the system, which is the purpose of the query reformulation engine. Information flow stems from the mapping process needs to be composed into a single coherent answer, which is done by result composition engine. These subsystems are out of the scope of our research.

User Interaction level
The AQUA query answering system itself, which provides precise answers to specific questions raised by the user. It integrates Natural Language Processing (NLP), Logic, Ontologies and Information retrieval techniques

3. Similarity algorithms

The similarity-mapping algorithm takes one entity from O1 and tries to find similar entity in O2 . The similarity mapping process has different levels as follows:

- Concept-name similarity with Character-based Jaccard measure [5].

$$sim_{x_a x_b} = \frac{x_a * x_b}{\|x_a\|\|x_b\| - x_a * x_b} \qquad \textbf{(1)}$$

where $x_a * x_b$ is the inner product of x_a, x_b and $\|x\|$ is the Euclidean norm for the vectors.

- Property set similarity with token based Jaccard distance: As first approach the property names are flattened into a bag of words per each node so similarity algorithms from the information retrieval field can be considered when two graph like structure are compared.
- Instance values similarity based on string similarity
- Concept-property similarity graph assessment

In order to increase the correctness of our similarity measures the obtained similarity coefficients need to be combined. Establishing this combination method is the primary objective that needs to be delivered with the with our outlined system. Further once the combined similarity has been calculated we need to develop a methodology to derive a belief mass function that is the fundamental property of Dempster-Shafer evidence theory.

In our prototype it is necessary to assess not only the syntactic but also the semantic similarity between concept, relations and the properties. The main reason why semantic heterogeneity occurs in the different ontology structures is the fact that different institutions developed their data sets individually, which contains mainly overlapping concepts. Assessing the above-mentioned similarities in our multi agent framework we adapted and extended the SimilarityBase and SimilarityTop algorithms [6,7] used in the current AQUA system for multiply ontologies. The goal of our approach is that the specialized agents simulate the way in which a human designer would describe its own domain based on a well-established dictionary. What also needs to be considered when the two graph structures obtained from both the user query fragment and the representation of the subset of the source ontology is that there can be a generalization or specialization of a specific concepts present in the graph which was obtained from the local source and this needs to be handled correctly. In our multi agent framework the extended and combined SimilarityBase and SimilarityTop algorithms can be described as follows:

1. Based on the WordNet reflexive lexical morphosemantic relation a directed graph is constructed from the FOL query fragment where there are bi-directional edges between the nodes representing the concepts and there are directed edges from the concepts to the property nodes. In this step the specialized agents try determine all possible alternatives for the meaning of the query fragment that it can be aware of. Figure 2 depicts the graph representation of the hasName(material, 10 CrMo 9 10) FOL query fragment.

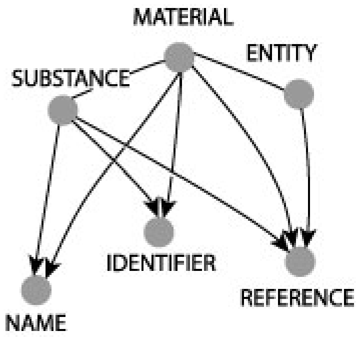


Figure 2. G0 query fragment graph

2. Based on the before mentioned character and token based Jaccard distance similarity measure the specialized agent builds up a directed graphs from the local ontology structures that supposedly answers the query fragment. Figure 3 depicts two graphs obtained from two different sources.

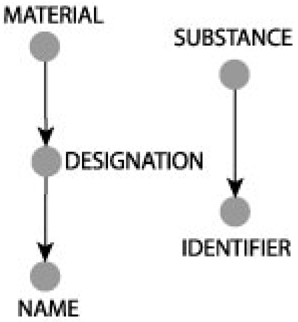


Figure 3. G1 and G2 graph representation of the local ontology fragment.

3. Top-down sub-graph (isomorphism) similarity assessment[8] is applied on the graph G0 in order to find the subgraph G1 and G2 respectively. The aim is to find identical subgraphs to G1 and G2 in order to assess the similarity of the concepts and properties that can answer the query fragment. We call this method a top-down assessment because the search for the sub graphs starts from the concept nodes towards property nodes through the directed edges. Once we reached the property node the search stops. If along the path we walked through the graph we found a sub graph identical (isomorph) to G1 and G2 that agent can deduce that the query fragment can be answered from the sources that belong to the particular ontology and the concepts or proper-

ties identified in the different sources are similar to both each other and to the query fragment and a basic mass function can be calculated that express the extent of belief in the existence of the similarity mapping between them. In case G1 or G2 contains nodes that could not be found in the G0, because of the nature of the top down assessment the agent can deduce that the particular concept node is a specialization of the concept that was identified by the algorithm.

4. Uncertainty handling

In our framework we use the Dempster-Shafer theory of evidence, which provides a mechanism for modeling and reasoning with uncertain information in a numerical way especially when it is not possible to assign a belief to a single element of a set of values. The main advantage of the Dempster-Shafer (D-S) theory over the classical probabilistic theories that the evidence of different levels of abstraction can be represented in a way that clear discrimination can be made between uncertainty and ignorance. Further advantage is that the theory provides a method for combining the effect of different learned evidences to a new belief by the means of the Dempster's combination rule. Let's first describe the basic concepts of the Dempster-Shafer theory and how it corresponds to our system.

Frame of Discernment (Θ): finite set representing the space of hypothesizes. It contains all possible mutually exclusive context events of the same kind. In our system this corresponds to the possible properties, those of the base entities that describes the concepts of the domain e.g. Material Name, Test Control, Specimen Identifier etc.

Evidence: available certain fact and is usually a result of observation. Used during the reasoning process to choose the best hypothesis in Θ. In our system this can be a certain observation that e.g. in the case of material entity the production details has been observed or not.

Belief mass function (m): is a finite amount of support assigned to the subset of Θ. It represents a strength of some evidence and

$$\sum\nolimits_{A \subseteq \Theta} m(A) = 1 \qquad (2)$$

where m(A) is our exact belief in a proposition represented by A. The similarity algorithms itself produce these assignment based on the before mentioned (Section 3) similarities e.g. between name and identifier property the assigned value is 0.7.

Once the belief mass functions have been assigned the following additional measures can be derived from the available information.

Belief: amount of justified support to A that is the lower probability function of Dempster, which accounts for all evidence E_k that supports the given proposition A.

$$belief_i(A) = \sum\nolimits_{E_k \subseteq A} m_i(E_k) \qquad (3)$$

Plausibility: amount of potential support on A that is the upper probability function of Dempster, which accounts for all the observations that do not rule out the given proposition.

$$plausibility_i(A) = 1 - \sum_{E_k \cap A = \varnothing} m_i(E_k) \tag{4}$$

Ignorance: the lack of information.

$$ignorance(A) = plausibility(A) - belief(A) \tag{5}$$

Once all the necessary variables have been assigned to a qualitative value we need to combine the belief mass functions that was created by the different agents for the particular query fragment.

Dempster's rule of combination:
Suppose we have two mass functions $m_i(E_k)$ and $m_j(E_k')$ and we want to combine them into a global $m_{ij}(A)$. Following Dempster's combination rule

$$m_{ij}(A) = m_i \oplus m_j = \sum_{E_k \cap E_{k'}} m_i(E_k) * m_j(E_{k'}) \tag{6}$$

However when $E_k \cap E_{k^{='}} = \varnothing$ the mass $m_i(E_k) * m_j(E_{k'})$ would go to $\varnothing$, it is necessary to normalize the mass function with the lost mass so

$$m_{ij}(A) = \frac{\sum_{E_k \cap E_{k'}} m_i(E_k) * m_j(E_{k'})}{1 - \sum_{E_k \cap E_{k'} = \varnothing} m_i(E_k) * m_j(E_{k'})} \tag{7}$$

An important part of the system is how the similarity measures are applied in the concrete scenario and how the particular agent assesses the belief mass functions and belief functions. In our experimental system we consider basic probability assessment over the following entities:

1. Class: The most basic concepts in the domain that correspond to classes that are the root of the various taxonomies
2. Object properties: Relation between the instances of two classes
3. Data type properties: Relation between instances of classes and RDF literals and XML Schema data types therefore it describes that the particular class e.g. material has a data type property called name that which is a string.

5. Working example

In this chapter we describe the main functionality of our system with a rather simple example. This example serves as a first test bed of this complex problem. The global descriptor describes what kind of information can be found in the different local ontologies/sources.

$$GD = DO_1 \cup DO_n \cup DO_{n+1} \qquad \textbf{(8)}$$

where GD is the global descriptor and DO_n is one of the particular domain ontology and

$$DO_i = \{R_{i1} ... R_{ij}\} \qquad \textbf{(9)}$$

where R_{ij} means the relation j in the ontology i.
As discussed the global descriptor can be best represented by FOL since the AQUA system also creates the query in FOL.

The global descriptor contains information about:

- Agents: MaterialAgent, SpecimenAgent, SourceAgent, TestConditionAgent and TestAgent as constant symbols

- Query and property information: (canAnswer(x,Test),hasInformation(x,MaximumStress)) as predicate symbols.

In the following example the system uses two ontologies O_1 and O_2 and creates similarity mapping between the query fragment and the concepts in the ontologies respectively. Both O_1 and O_2 ontology describes mechanical material test information from different institutes. Extracts from the two ontologies can be found in section 6.1.
To illustrate the mapping process in our system the following steps are taken before the query can be answered:

1. At system startup the Global Descriptor contains only the pre defined concept-mapping agent pairs that describe which agent knows the particular concept:
   ∀x Materialagent(x) and canAnswer(x,Material)
   ∀x Specimenagent(x) and canAnswer(x,Specimen)
   ∀x Testagent(x) and canAnswer(x,Test)
   ∀x Sourceagent(x) and canAnswer(x,Source)
   ∀x TestConditionagent(x) and canAnswer(x, TestCondition)
2. FOL Query passed to the broker agent:
   Which test has been carried out on a bar shaped specimen?

(∀x, ∃y) (Test(x) and Specimen(y) and form(y,bar) and carriedOutOn(x,y))

3. Broker agent decomposes the query based on the information present in the Global Descriptor and forwards it to the particular agents:
   - TestAgent→ Test(x) and carriedOutOn(x,y)
   - SpecimenAgent → Specimen(y) and form(y,bar) and carriedOutOn(x,y)

   Both agent received part of the query that corresponds to multiply entities. Since this is a relation between the two concepts, agents need to share the meaning of this expression. Agents place this into a blackboard, which is visible for all agents.
   - Blackboard→ carriedOutOn(x,y)
4. Test and Specimen agents retrieve fragments of two ontologies. Test Agent identifies two similar concepts:
   - O_1→TestResult and O_2→Test

   Specimen Agent identifies two similar properties:
   - O_1→Form and O_2→SpecimenForm

   a) Dempster-Shafer belief mass function is evaluated based on the node name
   similarities

TestAgent	SpecimenAgent
Test-TestResult=0.1	Specimen-Specimen=1.0
Control-TestControl=0.3	Form-SpecimenForm=0.3
Temperature-TestTemperature=0.4	Name-SpecimenName=0.25
Standard-TestStandard=0.2	Characterisation- SpecimenCharactesisation=.25

Table 1. Assigned belief function for the different entities.

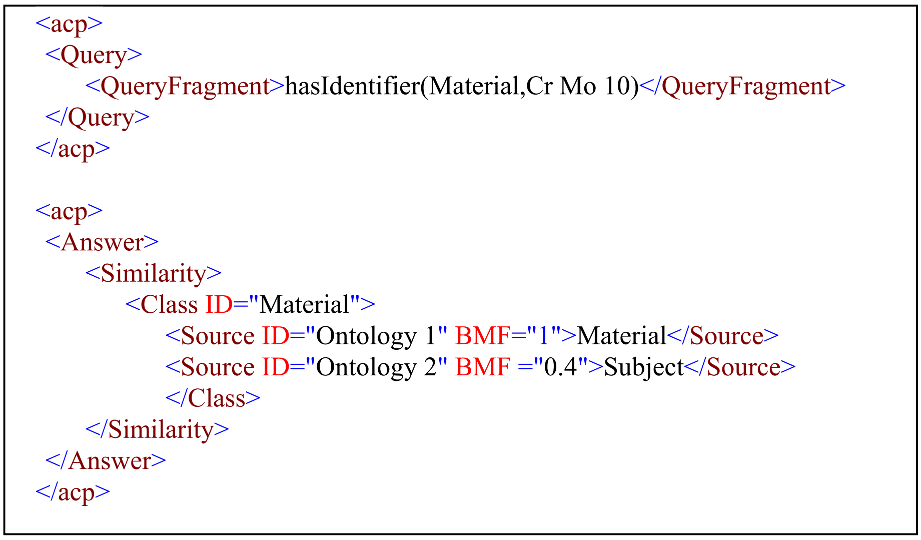
b) Dempster-Shafer belief mass function is evaluated (Table 1) based on the node structure similarities
Test(Control,Temperature,Standard)- TestResult(TestControl, TestTemperature, TestStandard)= 0.5
Specimen(Name,Form,Characterization) and Geometry(SpecimenForm,SpecimenName,
SpecimenChar)=0.6

c) Combined similarity, belief function can be calculated cooperatively by the two agents.
TestResult in O_1 is similar concept to Test in O_2 with belief function 0.8
Geometry in O_1 is similar concept to Specimen in O_2 and Form in O_1 is similar property in SpecimenForm in O_2

5. New findings can be added to the global descriptor:
   ∀x Testagent(x) and canAnswer(x,TestResult)
   ∀x Specimenagent(x) and canAnswer(x,Geometry)

6. Implementation

Our framework is implemented with JADE [9] agents using SWI prolog [10] engine to achieve reasoning capabilities. Because of the original ACL (Agent Communication language) implemented by JADE assumes that every used ontology is a subset of the domain ontology or there exists a map between it and the domain ontology; we defined our own agent



```
<acp>
 <Query>
     <QueryFragment>hasIdentifier(Material,Cr Mo 10)</QueryFragment>
 </Query>
</acp>

<acp>
 <Answer>
     <Similarity>
         <Class ID="Material">
             <Source ID="Ontology 1" BMF="1">Material</Source>
             <Source ID="Ontology 2" BMF ="0.4">Subject</Source>
             </Class>
     </Similarity>
 </Answer>
</acp>
```

Figure 4. Agent Communication Protocol

communication protocol(Figure 4) that sits atop of the standard ACL messages and describes not only the similarity information but the quantitative measure of the uncertainty inherent to the mapping process. This protocol is a simple XML based communication protocol called ACP (Agent Communication Protocol) that is tightly integrated with the AQUA FOL formula representation and the specific nature of the question answering. The two main entities are the query and the answer. The sub elements in each node depend on which agent communicates with whom e.g. the query and answer structure between the broker and the mapping agents is depicted before.

6.1 Source ontologies

Our ontology O is defined by its set of concepts C (instances of “owl:Class”) with a corresponding relations R (instances of “owl:ObjectProperty or owl:”DataTypeProperty”) exist between single concepts. Ontologies that describe the entities in the different databases cover the main domain specific concepts like test result, source, material, specimen, test condition, etc. We assume that different institutions create their own domain specific ontology and since these domains describe the same information in different domains their designers have a different conceptualization, which leads to a different definitions of concepts and relationships for same objects even if it is expressed in the same ontology language. The following example ontology fragments describe two data source where in ontology 1 there is a relation explicitly described between the TEST and the SPECIMEN whereas in the second example it is expressed through one unique property of the SPECIMEN namely the identifier.

Our examples are represented in OWL ontology language(Figure 5,6)

```
                              ONTOLOGY 1

<owl:Class rdf:ID="Test"/>
<owl:Class rdf:ID="Specimen"/>
<owl:DatatypeProperty rdf:ID="Control">
   <rdfs:domain rdf:resource="# Test"/>
   <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:ObjectProperty rdf:ID="hasSpecimen">
   <rdfs:range rdf:resource="#Test"/>
   <rdfs:domain rdf:resource="#Specimen"/>
</owl:ObjectProperty>
```

Figure 5. Sample ontology fragment

```
                              ONTOLOGY 2

<owl:Class rdf:ID="TestResult"/>
<owl:Class rdf:ID="Specimen"/>
<owl:DatatypeProperty rdf:ID="Control">
   <rdfs:domain rdf:resource="# Test"/>
   <rdfs:range
rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="SpecimenIdentifier">
   <rdfs:domain rdf:resource="#TestResult"/>
   <rdfs:range
rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
```

Figure 6. Sample ontology fragment

7. Related work

Ontology mapping is widely investigated area and a numerous approaches led to different solutions.

Derived from the data engineering community several solutions have been proposed that based on a mediator architecture where logical database schemas are used as shared mediated views over the queried schemas. A number of systems have been proposed e.g TSIMMIS[11], Information Manifold [12], InfoSleuth [13], MOMIS [14] that shows the flexibility and the scalability of these approaches.

Derived from the knowledge engineering community solutions the use of ontologies (conceptual domain knowledge schemas) is the main approach for resolving semantic differences in heterogeneous data sources.

To date uncertainty handling during the mapping process was not in the focus of the research community since initially only different logic(FOL, Description Logics) based approaches has been utilized. As practical application of ontologies emerged on the web it has been acknowledged that considering the dynamic nature of the Web the problem of inconsistencies, controversies and lack of information needs to be handled. First systems that used probabilistic information like LSD, GLUE [15] proved that combining different similarity measures based on their probability could significantly improve the accuracy of the mapping process. It is worth to note that the Baysian networks and different variants dominate current research addressing the qualitative reasoning and decision-making problem under uncertainty. Although these approaches successfully lead to numerous real world applications there are several situations where the problem cannot be represented properly within the classical probability framework. The most related research for ontology mapping framework under uncertainty using Bayesian networks [16] to tackle this problem.

8. Conclusion and future research

In our prototype we successfully addressed the problem of a single agent or application that is limited by its knowledge, perspective, and its computational resources. It is clear that if we try to move towards a fully automated ontology mapping in order to provide a better integration of the heterogeneous sources we need to investigate the limitations of multi agent systems such as our prototype. In this complex environment different scientific disciplines need to be utilized together to achieve better results to the users' query within an acceptable time frame. We think that in our implementation we have made a encouraging step towards a theoretical solution but the different key system components such as similarity measure or the uncertainty handling part needs to be investigated further. In our future research we are planning to establish a qualitative comparison of the similarity algorithms that fulfill all the requirements of our examined domain and our tasks.

We believe that probability theory and distribution does not have enough expressive power to tackle certain aspects of the uncertainty e.g. total ignorance.

As a consequence we expect that evidence (Dempster-Shafer) theory is the most suitable approach and needs to be investigated in ontology mapping context thought this has not been done so far. The reason is that Dempster Shafer combination rule can easily be unfeasible in case of domains with large number of variables. Different optimalisations methods have been developed but to date we could not find approaches that considered distributed environment. Local computation and valuation networks uses joint tree structure to narrow down the number of focal elements and different architectures has been proposed based on message passing schemes to carry our inference and resolve the problem of the Dempser's rule of combination. In our scenario we assume a dynamic multi agent environment where different agents has partial knowledge of the domain.

References

[1] Vargas-Vera M. and Motta E. (2004) AQUA - Ontology-based Question Answering System. Third International Mexican Conference on Artificial Intelligence (MICAI-2004), Lecture Notes in Computer Science 2972 Springer Verlag, (eds R. Monroy et al.), April 26-30, 2004.

[2] Vargas-Vera M., Motta E. and Domingue J. (2003) AQUA: An Ontology-Driven Question Answering System. AAAI Spring Symposium, New Directions in Question Answering, Stanford University, March 24-26, 2003.

[3] MatML Materials Markup language, http://www.matml.org

[4] Shafer Glenn,(1976) A Mathematical Theory of Evidence. Princeton University Press.

[5] Haveliwala T, Gionis A., Klein D., and Indyk P (2002). Evaluating strategies for similarity search on the web. Proceedings of WWW, Hawai, USA, May 2002.

[6] Vargas-Vera M. and Motta E. (2004) A Knowledge-Based Approach to Ontologies Data Integration. KMi-TR-152, The Open University, July 2004.

[7] Vargas-Vera M. and Motta E. (2004) An Ontology-driven Similarity Algorithm. KMI-TR-151, Knowledge Media Institute, The Open University, July 2004.

[8] Atallah Mikhail J.,(1999) Algorithms and Theory of Computation Handbook, ed., CRC Press LLC, 1999.

[9] JADE Web Site, http://jade.tilab.com/

[10] Wielemaker J . SWI-Prolog 5.1: Reference Manual. SWI, University of Amsterdam, Roetersstraat 15, 1018 WB Amsterdam, The Netherlands, 1997-2003. E-mail: jan@swi.psy.uva.nl.

[11] Garcia-Molina, H.; Papakonstantinou, Y.; Quass, D.; Rajararnan, A.; Sagiv, Y.; Ullman, J.;Vassalos, V.; Widom, J (1997) The TSIMMIS Approach to Mediation: Data Models and Languages, Journal of Intelligent Information Systems, 8(2):117-132.

[12] A. Halevy (1998) The Information manifold approach to data integration.

[13] Bayardo, R.; et al.. Infosleuth (1997) Agent-based Semantic Integration of Information in Open and Dynamic Environments. In Proceedings of ACM SIGMOD Conference on Management of Data, 195-206. Tucson, Arizona.

[14] Beneventano, D.; Bergamaschi, S.; Guerra, F.; Vincini, M..(2001) The MOMIS Approach to Information Integration. In ICEIS(1), 194-198.

[15] Doan, A. H.; Madhavan, J.; Domingos, P.; Halevy (2002) A. Learning to Map between Ontologies on the Semantic Web. In WWW 2002.

[16] Zhongli Ding, Yun Peng, Rong Pan. A Bayesian Approach to Uncertainty Modeling in OWL Ontology. In Proceedings of 2004 International Conference on Advances in Intelligent Systems - Theory and Applications (AISTA2004). November 15-18, 2004, Luxembourg-Kirchberg, Luxembourg. in Intelligent Systems - Theory and Applications (AISTA2004). November 15-18, 2004, Luxembourg-Kirchberg, Luxembourg.

Semantically correct Visio

Christian Fillies[1], Frauke Weichhardt[1], and Bob Smith[2]

[1] Semtation GmbH
Potsdam, Germany
[2] Tall Tree Labs,
Huntington Beach, CA

Abstract. This paper presents some use cases of ontologies outside of the OWL community. We see these fields as initializers for non specialized users that are not familiar with the Semantic Web. Regarding these fields we experience a need for easy ontology editing, on different levels of ontology complexity. To fulfil these needs, we show how users make use of Semantic Web technology while modelling with Microsoft Visio. We explain how ontologies are used to ensure semantic consistency while flowcharting, which is the most important use case for Visio. We also present a graphical notation for authoring OWL in Visio and discuss which part of Description Logic can be expected to be used frequently.

1 SemTalk and Visio

SemTalk is a graphical editor for various modeling solutions based on Microsoft Visio. Visio was chosen as a platform because of its great graphical flexibility and its extendable design. It has a large installed base in the information worker community. SemTalk basically adds a Meta model layer to Visio, which allows specifying syntax for modeling methods on top of Visio shapes. Custom data, reporting and navigation are realized using an internal xml database.

It provides business process modeling, product modeling as well as a graphical notation for authoring and visualizing OWL. In respect to ontologies the focus of the tool is not on being a data store for large ontologies rather than an easy to use front end for manual editing of ontologies in a distributed environment of OWL aware systems. SemTalk provides consistency checking inside one Visio document and basic consistency checking between multiple models. While modeling SemTalk compares each text, which is entered by the user with a given list of ontologies.

2 Modeling with a corporate Semantic Web

Only a small section of our users create OWL models for its own sake. In most of the current models ontologies are used to normalize names of items in models made for a different purpose than authoring OWL. Examples are the process steps in a business process or the names of dimensions in a data warehouse. Another field is product modelling or knowledge management oriented models for portal building or EAI subjects. The intention of using ontologies for that is to create content which is semantically consistent with other content created in the same community. This means trying to check one model against the other semantically. People have models they use as checking or reference models. Often these models are glossaries or data dictionaries from other applications like Enterprise Resource Planning systems (ERP) or portals.

Any existing OWL or RDFS source can be used as a repository or glossary to ensure consistency. People often use lists of business objects provided by ERP vendors like SAP. For more general purposes web services like “WordNet” can be applied. In a corporate environment company or department specific ontologies are used. The resulting models are published in two ways: For end users graphical representations of the models are published on the intranet as HTML, MS Word or PowerPoint. For other modellers the model itself is available as a reusable component, e.g. a process model to be refined with subprocesses or reused as a process component. This makes all models a distributed web of knowledge.

Modelling of business processes and products in the context of a distributed web of knowledge differs significantly from the way those models have been created before. Before new terms are introduced, the user has to investigate if the term or fact already exists in the community semantic web. If the term already exists, the user model will reference that term by using the same URN and providing an URL to obtain its definition. Existing terms may be extended by subclasses or existing definitions of properties are added. If the concept is identical but the current domain requires a different name, a synonym can be added. For example a *customer* will be called *patient* in a medical domain.

The ontology contained, e.g. in a process is available for reuse in different processes in the same domain. The most common use case of ontologies in process modelling is to localize content to multiple languages. This is done by translating objects in the ontology which will automatically generate translated business processes. Sometimes ontologies created for one specific purpose can be reused for a new modelling problem in the same domain. For example a product catalogue made for the web shop can be reused in a process modelling project.

3 Ontologies for Business Processes as an example of light weight ontologies

The specification of business processes is a task executed by end users or consultants who are often specialized on process optimization or ERP systems. Those people are usually not educated in Description Logic and we do not experience a lot of enthusiasm to learn about it in order to make “better” ontologies.

For our purpose, which is ensuring consistency of other models, it is sufficient to build taxonomies, sometimes enriched with properties in order to make them more readable. Users have to learn about process modelling languages and a minimum of object-oriented thinking in order to apply the ontologies to their process models. We use subclassing, DataProperties and ObjectProperties. For process models we also add the list of valid verbs to the classes.

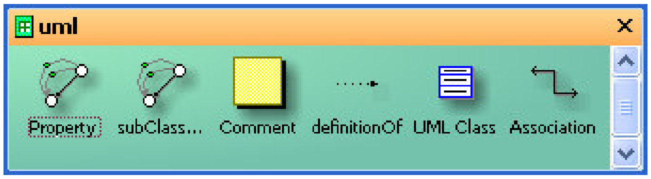


Figure 1: Class Shapes

UML-style symbols (Figure 1) are used to represent classes and connectors for “Property” and “subClassOf”. There is also a specific connector named “Association” which can display cardinalities on ObjectProperties in a UML like style. DataProperties and methods (verbs) are displayed within the UML class shape. A lot of users are familiar with UML class shapes. The language which is used and supported by the SemTalk internal inference engine is similar to RDFS.

4. OWL-Ontologies as an example of heavy weight ontologies

In order to be able to express the complete language set of OWL within Visio we extended the UML shapeset with OWL specific connectors and shapes[1] (see Figure 2).

On classes we have added the constraints “disjointWith” and “equivalentClass”. Different from standard SemTalk, instances are allowed in class diagrams and are allowed to be instance of multiple classes.

[1] The OWL Shapeset was jointly developed with Network Inference in order to use it as a graphical front end for their Cerebra reasoner.

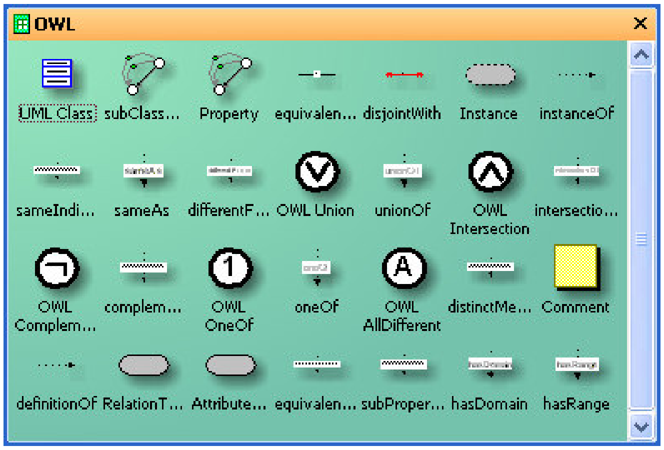


Figure 2: OWL Shapes

Anonymous classes such as *unionOf* are being expressed by a mastershape (“OWL Union”) for the class and a connector (“unionOf”) for the membership. Analogue combinations of master and connectors have been chosen for *intersectionOf*, *complementOf* and *oneOf.* These connectors can also be used on ordinary, named classes.

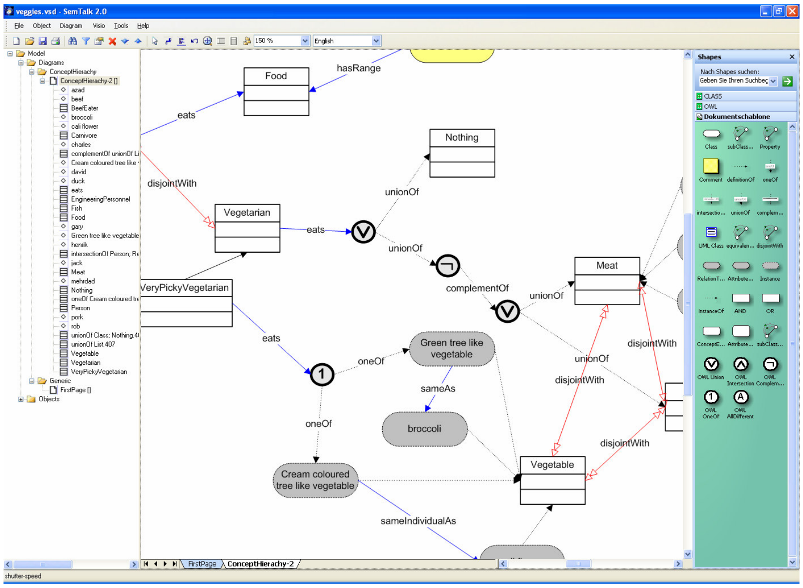


Figure 3: Anon Classes

As an addition to standard SemTalk class diagrams we have special Visio shapes to represent ObjectProperties (“RelationType”) and DataProperties (“AttributeType”) as objects in the diagram which can have graphical links “hasDomain” and “hasRange” to other objects (Figure 4).

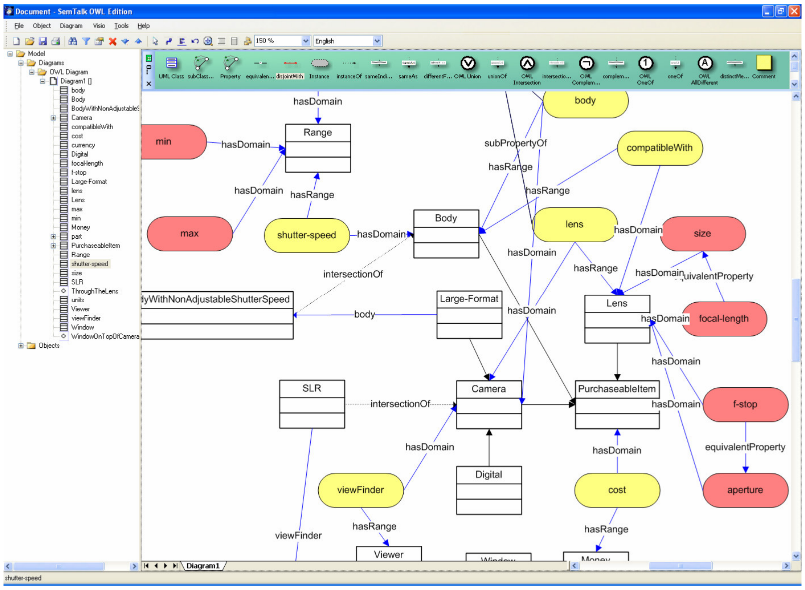


Figure 4: Properties in an OWL Diagram

Using these expressions new OWL files can be created and existing OWL files can be presented in a manually or automatically arranged way. Because predefined Visio shapes can be used to represent classes and objects, OWL models designed with SemTalk are often better understandable for non-technical end users than models created with other tools. Even if the graphical notation makes authoring OWL simpler than entering the same OWL data with other tools, it does not educate people in Description Logic. Compared to the amount of users entering knowledge using MS PowerPoint and MS Visio, the number of users specifying knowledge with OWL will be small and limited to technical experts integrating IT-Systems in EAI or Portal scenarios. We do not expect people to annotate their documents manually by modelling the contents of documents in a way inference engines can "understand" the documents. Resulting from complexity of the DL-modelling paradigm in full OWL even for stand-alone models an inference engine is needed to prove their correctness.

For some of the constraints it also makes sense to enforce consistency in a distributed environment even for taxonomies. This is especially true for disjointness, which can be violated without using any other OWL constructs other than subclassOf. A major challenge we see for inference engines is to support the distributed modelling of business processes including support for finding homonyms. Homonyms are different words having the same meaning.

5. Tools for Semantic Web Authoring

In the early nineties business process modelling has started from revolutionary ideas of Michael Hammer, who proposed business process reengineering. Pushed by the success story of ERP systems, 10 years later process modelling made its way from an academic discipline using research prototypes to a commercial component integrated in Microsoft Office used for any serious system integration.

Ontology modelling is still in its early stage. Most ontologies are made by academics using non-commercial tools which have their roots in research often in Artificial Intelligence. The “Semantic Web” in its original sense seems to be far away from reaching the critical mass required for a takeoff. But semantic technology is one of the very few technologies of the last decade which seems to ignore Gartner’s “Hype Cycle” [Eric Miller, STC05]. There has been slow but continuous growth on semantic technology and an end is not foreseeable.

Ontologies offer great value to common modelling problems especially to process modelling. Specification of procedural knowledge in processes is very common. Specification of static knowledge and rules in ontologies can be seen as an extension. Support for maintenance of static and dynamic knowledge will become part of knowledge worker’s workplace. Building end user tools for static knowledge can and will benefit from experiences made with process modelling tools.

5. Conclusion

We believe in ontology modelling as a great way of enhancing current possibilities of writing computer programs on one hand and of closing the gap between users and IT specialists on the other hand. For this we see the need to enable everybody to develop, document and maintain his or her ontology, be it in a conscious manner using DL metaphors or be it unconsciously while modelling a business process or a product. In order to fulfil this need we present a graphical way of editing ontologies, that enables all kinds of users to participate in the great vision of the semantic web.

References

[BHL01] Berners-Lee, T. Hendler, J., and Lassila, O.: published an article about the Semantic Web in Scientific American. http://www.scientificamerican.com/2001/0501issue/0501berners-lee.html

[GRU95] Gruber, T. (1995). Towards principles for the design of ontologies used for knowledge sharing. International Journal of Human-Computer Studies, (43):907–928.

[HF03] van Hoof, A, Fillies, C: Das semantische Unternehmensprozessweb, Künstliche Intelligenz 4/03

[FWW02] Fillies, C., Wood-Albrecht, G., Weichhardt, F.: A Pragmatic Application of the Semantic Web Using SemTalk, WWW2002, May 7-11, 2002, Honolulu, Hawaii, USA ACM 1-5811-449-5/02/0005

[OWL02] OWL Web Ontology Language 1.0 Reference: W3C Working Draft 29 July 2002, 12 November 2002. Mike Dean, Dan Connolly, Frank van Harmelen, James Hendler, Ian Horrocks, Deborah L. McGuinness, Peter F. Patel-Schneider, and Lynn Andrea Stein eds. Latest version is available at http://www.w3.org/TR/owl-ref/

[STC05] Semantic Technology Conference 2005, http://www.semantic-conference.com

The table metaphor: A representation of a class and its instances

Jan Henke

Digital Enterprise Research Institute (DERI)
University of Innsbruck, Austria
jan.henke@deri.org

Abstract

This paper describes an approach on how to visualize instances and the relation to their classes. The approach is motivated and described in the context of ontologies and the Semantic Web but is general enough to be utilized for any object model visualization.

1 Introduction

For the Semantic Web ontologies are the basic building block. On the one hand they allow for a common vocabulary and thus for communication and interoperation. On the other hand they can be used for logical reasoning and therefore statements can be verified / falsified and knowledge can be inferred.

For ontologies in turn, classes and instances are the building blocks. As in object orientation they are used to model abstract definitions on the one hand and concrete examples of these on the other hand. How the visual editing of ontologies can be gained by a new representation of the class instance relation – called table metaphor – will be described in the following.

In section 2 the requirements of a class - instance visualization are listed before section 3 checks which of these are fulfilled by current approaches. Section 4 will address the table metaphor and section 5 concludes this paper.

2 Requirements

A good visualization should provide a correct transformation as well as a high usability. Leaving out the first would make it useless; leaving out the second would make it unused. Based on this belief some specific requirements shall be described below.

2.1 Correctness

Based on a classification by Shneiderman [5] visualization can be divided into the seven subtasks “Overview”, “Zooming”, “Filtering”, “Details on demand”, “Relations”, “History” and “Extraction”. The requirement of correctness to be described below is derived from the “Relations” task which is about emphasizing which item belongs to which other one.

Class membership
An instance cannot be used if its class membership - and thus its definition - isn’t clear. The other way around, a class without instances isn’t very useful (except for the case of abstract classes) because the definition has never been applied. Therefore it is crucial that a visualization unambiguously reveals this relation in both directions.

Attribute value mapping
The relation between attributes and their values can be compared to the one between classes and instances: While an attribute defines a range (a class) this requirement is fulfilled by the respective value (an instance). For this reason also the visualization of the attribute-value-relation has to be bidirectional and unambiguous.

Level matching
An instance is a concretization of a class. It reduces the abstractness by filling slots with values. Nevertheless an instance is neither a sub object of a class nor the other way around. Despite the different levels of abstraction they belong onto the same level of definition, i.e. an instance is as special as its class – not more special as a subclass (see Figure 1).

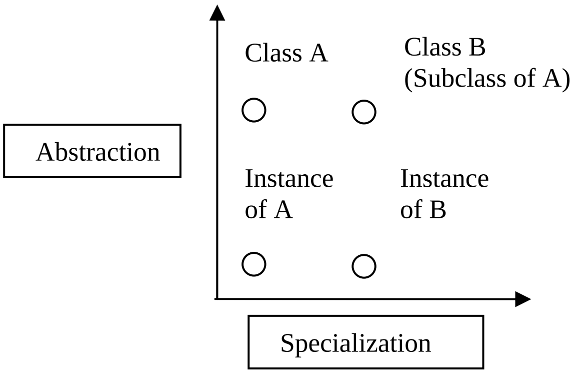


Figure 1 Concreteness vs. Specialization

Because of this, a visualization has to make sure that a class and its instances are positioned on one and the same level.

2.2 Usability

Usability can be described using the five criterions learnability, efficiency, memorability, errors and subjective pleasure [2]. In the following these requirements shall be applied to the special case of visualization.

Learnability
Learnability stands for a minimization of the learning phase duration. This is also required for a visualization so that it can be utilized as fast as possible.

Efficiency
Efficiency describes how much work can be done in a certain amount of time. An increased efficiency is a clear indicator for an improved visualization.

Memorability
After a longer break between two usage sessions it should not be necessary to restart the learning phase. Therefore a visualization should be simple enough to be memorized.

Errors
Errors should appear as seldom as possible and if they appear they should be fixable as easily as possible. For the visualization case this can be translated to the requirement of high clarity.

Subjective pleasure
The less formal criterion of subjective pleasure should not be underestimated. The user should have a good “feeling” using the visualization in order to improve his / her working results.

3 Current approaches

In order to show the lack in current instance visualizations and thus to motivate the table metaphor two current approaches will be described in the following.

3.1 Class tree duplication

One approach in current ontology editors – like for instance Protégé [4] – is to display instances connected to a copy of the class tree.

Protégé provides different tabs for classes and instances. If the instance tab is selected the class tree is displayed again – and once a class is selected, its instances will be displayed. Thus the class membership can easily be recognized.

Also the relation between attributes and their values is revealed unambiguously which fulfills the second criterion of the correctness requirement.

The requirement of level matching cannot really be decided. As mentioned above when a class is selected its instances are displayed. But it is not really clear whether they are still on the same level as the class.

Beyond this, the usability requirement of subjective pleasure can hardly be fulfilled by an approach that repeats the class tree in two different views, as it is done in Protégé.

3.2 Purely textual visualization

Another approach – as can be found in Oiled [1] e.g. – is to offer no graphical representation of the class instance relation but to display it purely textually.

In Oiled there can be found both a tab for classes and one for instances – comparably with Protégé. The difference consists in the fact that there's no class tree available in the instance tab. Instead of this the class membership can only be found in a respective combo box – thus purely textually.

The visualization of attributes and their values utilizes a table and is nicely usable.

The not very smooth class tree repetition of Protégé is not used – but unfortunately, it has not been replaced by any other feature. Because of the missing graphical connection between classes and their instances the level matching criterion cannot be applied.

4 The table metaphor

As shown above current visualization approaches hardly fulfill the mentioned requirements. Therefore a new idea – called table metaphor – shall be introduced.

The table metaphor represents a class and its instances by a table. More precisely this means a class is represented by a table header while each table row stands for an instance (see Table 1 – three instances of the class "City" with the attributes "Name" and "Inhabitants" are displayed). Thus an ontology – consisting of a schema and a knowledge base – can be seen as a tree of tables.

Table 1 Instance table example

Name	**Inhabitants**
Berlin	3 420 000
Hamburg	1 640 000
Munich	1 220 000

4.1 Correctness

Using the criterions described in the requirements section the correctness of the table metaphor will be shown below.

Class membership
The table metaphor allows for an easy recognition of class membership. Whenever a certain class is selected the respective table can be displayed.

Attribute value mapping
Using the table metaphor a certain attribute value is always displayed underneath the respective header cell. Thus it is always clear which attribute a value belongs to and which values have been assigned to a certain attribute.

Level matching
The header and the body of a table can be clearly distinguished thus the class and the instance part are explicitly separated. On the other hand a table body is not a sub object of a table header which fulfills the level matching criterion.

4.2 Usability

In the following it will be shown that a high usability can be expected of object model editors that apply the table metaphor.

Learnability
The presented approach is very simple. Tables are applied manifold in everyday life. Thus the leaning phase can be expected to be minimal.

Efficiency
Because of the dissemination of tables they can be read and understood quickly. Beyond this available widgets that allow for column wise sorting e.g. should also improve efficiency by far.

Memorability
Because hardly anything has to be learned in order to understand the table metaphor the problem of insufficient Memorability cannot appear.

Errors
Also the error minimization is guaranteed by the simplicity of the approach. If error correction strategies should be necessary nevertheless this has to be solved at implementation level.

Subjective pleasure
This criterion can hardly be predicted. But if the approach is well understood and thus increases efficiency also the subjective pleasure should be influenced in a positive way.

5 Conclusions

In this paper a visualization approach – called table metaphor – was introduced. A real world implementation of it can be inspected in the Distributed Ontology Management Environment (DOME) [2] – to be found at http://www.omwg.org.

References

1. Bechhofer, Sean; Horrocks, Ian; Goble, Carole; Stevens, Robert. OilEd: a Reason-able Ontology Editor for the Semantic Web. Proceedings of KI2001, Joint German/Austrian conference on Artificial Intelligence, September 19-21, Vienna. Springer-Verlag LNAI Vol. 2174, pp 396--408. 2001.
2. Henke, Jan. Architecture Design of an Editing & Browsing tool, 2004
3. Nielsen, Jakob. Usability Engineering. Morgan Kaufmann - An imprint of Academic Press, 1993
4. Noy, N. F.; Sintek, M.; Decker, S.; Crubezy, M.; Fergerson, R. W.; & Musen, M. A.. Creating Semantic Web Contents with Protege-2000. IEEE Intelligent Systems 16(2):60-71, 2001.
5. Shneiderman, Ben. Designing the User Interface – Strategies for effective Human-Computer Interaction. Addison-Wesley-Longman, 1998

User Profiling for Interest-focused Browsing History

Miha Grčar, Dunja Mladenič, Marko Grobelnik

Jozef Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia
{Miha.Grcar, Dunja.Mladenic, Marko.Grobelnik}@ijs.si
http://kt.ijs.si

Abstract. User profiling is an important part of the Semantic Web as it integrates the user into the concept of Web data with machine-readable semantics. In this paper, user profiling is presented as a way of providing the user with his/her interest-focused browsing history. We present a system that is incorporated into the Internet Explorer and maintains a dynamic user profile in a form of automatically constructed topic ontology. A subset of previously visited Web pages is associated with each topic in the ontology. By selecting a topic, the user can view the set of associated pages and choose to navigate to the page of his/her interest. Each topic can be seen as an interest of the user (hence the term *interest-focused* browsing history). The ontology is constructed by transforming the textual contents of the pages into sparse word-vectors and applying bisecting k-means clustering (i.e. a form of hierarchical clustering) on the set of sparse vectors. The most recently visited pages are used to identify the user's current interest and map it to the ontology. The user can clearly see which topics, and their corresponding pages, are related (or are not related, for that matter) to his/her current interest. We see this as a useful way of organizing the user's browsing history. To illustrate the functioning of the system, we demonstrate its behavior in one particular real-life scenario.

1 Introduction

In this paper, user profiling is presented as a way of providing the user with his/her interest-focused browsing history. We present a system that is incorporated into the Internet Explorer and maintains a dynamic user profile in a form of automatically constructed topic ontology.

Let us begin by briefly summarizing some of the related work in the field of user profile construction. The most related work is that of (KIM AND CHAN, 2003). They propose a tree-like hierarchy of interests, the root being the user's general interest (i.e. long-term interest) and leaves representing domains the user is – was ever – interested in (i.e. short-term interests). User interest hierarchies are built using a form of hierarchical clustering on a set of Web pages visited by a user.

Another less related way of constructing a user profile is to analyze the user's browsing history and apply modified collaborative filtering techniques (SUGIYAMA ET AL., 2004). Here, the user profile is also a combination of both (i) user's persistent preferences (long-term preferences) and (ii) user's ephemeral preferences (short-term preferences – "today's" preferences) and is represented as a vector of term weights. Modified collaborative filtering is then applied to a user-term matrix (in contrast to being applied to a user-item matrix as is the case with the original collaborative filtering approach – hence the word "modified") to predict the missing term weights in each user profile. Clustering is used (in one of their approaches) to determine user communities. Cluster centroids are compared to the active user's term vector to find the user's neighborhood (a threshold is used to discard less relevant communities). The latter approach, according to (SUGIYAMA ET AL., 2004), achieves the best results.

In Foxtrot recommender system (MIDDLETON ET AL., 2003), an ontology (taxonomy) based on CORA digital library is used – new documents are classified into the taxonomy by using a variant of the nearest neighbor algorithm. A user profile holds a set of topics and their corresponding interest values. Each topic adds 50% of its interest value to its super-class. They also used "static knowledge" ontologies to alleviate the cold-start problem. Visualization of profiles is used to encourage immediate users' feedbacks. For evaluation, collaborative filtering is performed on a user-topic matrix (they term this technique "collaborative and content-based recommendations").

The rest of the paper is arranged as follows. In Section 2, user profiling is viewed from the perspective of the Semantic Web. Architecture of our system is presented in Section 3. In Section 4, we demonstrate the functioning of the system in a real-life scenario. The paper concludes with the discussion and some ideas for future work in Section 5.

2 User Profiling from the Perspective of the Semantic Web

When thinking of the Semantic Web we can say that the Semantic Web is a Web focused on the exchange of information between computers that does not explicitly involve human users. Although computers could be quite busy communicating to each other, there still needs to be some space left for human users in the whole process – there is where user profiling comes into the play.

Technically speaking, the Semantic Web is mainly about the data that are self-explanatory, or in other words, about the data which are annotated in some standard fashion that enables efficient computer-to-computer communication. The main purpose of the Semantic Web is to enable better services for the end-users. Since in general the data can be understood in more than one way – especially when talking about the more abstract categories which cannot be annotated explicitly – one of the possible sources of annotations (i.e. meta-data) may also be the information about the user. This information can be represented in several ways. Typically, if we talk about more abstract and aggregated information, we talk about *user profiles* or *user models*. Their main characteristic is the ability to generalize the collected data about the user's behavior (such as click-stream data of the user's browsing behavior). Such *user-models* are then used to annotate the data in such a way that Web services are able to deliver personalized information, aiming at increasing the user's efficiency when he/she is communicating with the computer.

We can conclude this short description of user profiling from the perspective of the Semantic Web by saying that user profiling is an important source of meta-data on the user's understanding of the data semantics. In particular, this compensates for the differences in users' understanding of the data by using an alternative annotation, which is more of the *soft* nature (the *softness* comes from the fact that the data are annotated implicitly and dynamically by taking a user profile into the account). The main goal of user modeling is increasing the efficiency of user activities by delivering more personalized information.

3 Architecture of the System

The system provides a dynamic user profile in a form of topic ontology. After a page is viewed, the textual content is extracted and stored as a text file as described in Section 3.1. Pages are represented as word-vectors (also termed *bags of words*) as explained in Section 3.2. To construct the topic ontology, we perform a variant of hierarchical clustering (see Section 3.3). By using the cosine similarity measure, we are able to map the user's current interest to the topic ontology (more details in Section 3.4). The latter identifies the ontology nodes that are in the context of the user's current interest. The whole process is illustrated in the system architecture figure (Figure 1) which also includes the references to Sections 3.1 through 3.4. These sections contain a detailed description of individual phases of the process.

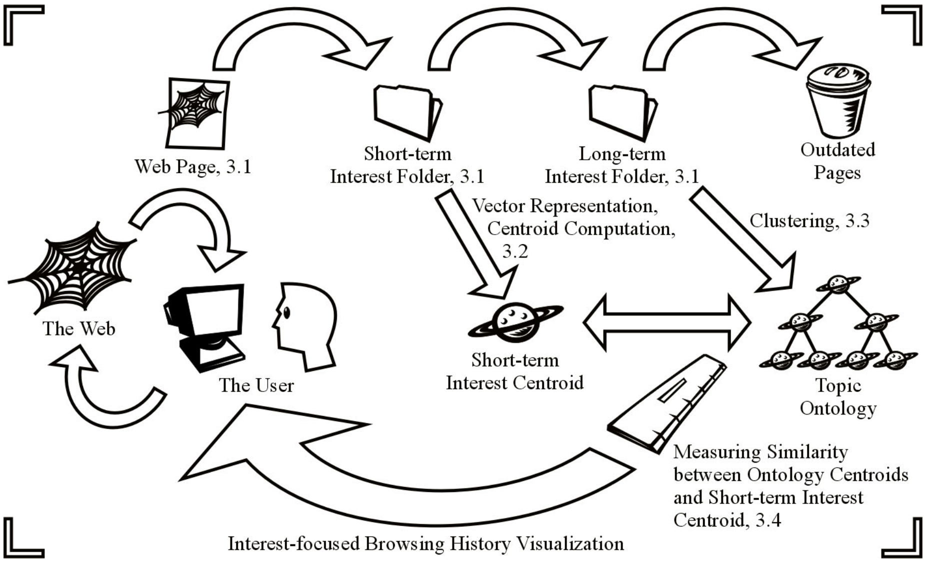


Figure 1. The Interest-focused Browsing History Architecture. The process is described throughout Section 3. The corresponding subsections are noted in the figure (3.1, 3.2, 3.3, 3.4).

3.1 Handling a Page-view

After a page is viewed, the textual content is extracted and stored as a text file. The text extraction is done in two relatively simple steps:

(i) text segments between and including “<script>” and “</script>” or “<style>” and “</style>” are discarded,
(ii) substrings starting with “<” and ending with “>” are removed.

A collection of such text files (from now on simply termed *pages*) is maintained in two folders. The first folder holds *m* most recently viewed pages (*the short-term interest folder*). In our experiments, *m* is set to 5. The second folder contains the last *n* viewed pages, where *n* > *m* (*the long-term interest folder*). In our experiments, *n* is set to 300. When a page is first visited, it is placed into both folders. Eventually it gets pushed out by other pages that are viewed afterwards. A page stays in the long-term interest folder much longer than in the short-term interest folder (hence the terms *long-* and *short-term*), the reason for this being a much higher number of new pages that need to be viewed for the page to be pushed out of the long-term interest folder.

Pages are named after their 128-bit MD5 hash codes. In this way we are able to, at least to some extent, detect a page that was already visited and handle this scenario. Currently we simply update the timestamp of the file (i.e. the page) to mark it as *recently interesting*. This action is carried out in both folders.

3.2 Word-vector Representation of a Page

To build a user profile, we first take the pages from the short-term interest folder and compute their TFIDF vector representations of the textual content, ignoring the order of words (thus such vectors are also termed *bags of words*, see Figure 2 for illustration), as introduced in (SALTON AND BUCKLEY, 1987). Each vector component is calculated as the product of Term Frequency (*TF*) – the number of times a word *W* occurs in the page – and Inverse Document Frequency (*IDF*), as explained by the following equation:

$$d^{(i)} = TF(W_i, d)IDF(W_i),\ \ where\ \ IDF(W_i) = \log\frac{D}{DF(W_i)}$$

where *D* is the number of pages and document frequency *DF*(*W*) is the number of documents in which word *W* occurred at least once.

Prior to transforming pages into vectors, stop-words are removed and stemming is applied. After vectors are obtained, the centroid of short-term interest pages is computed by averaging corresponding TFIDF vectors component-by-component. This process combines the short-term interest pages, regardless of their count, into one single construct – the short-term interest centroid.

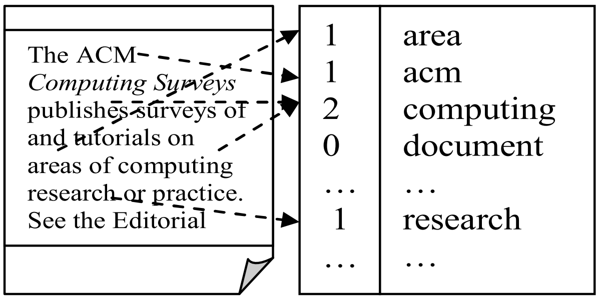


Figure 2. An illustration of a word-vector (term-frequency vector, in this particular case) representation of a document.

3.3 Constructing the Topic Ontology

The long-term interest pages are treated slightly differently from the short-term interest pages. We first perform the bisecting k-means clustering (i.e. a variant of hierarchical clustering) over the long-term interest TFIDF vectors. This clustering method is computationally efficient and was already successfully applied on text documents (STEINBACH ET AL., 2000). At start, all the pages form the root cluster which is first divided into two child clusters (hence the term *bisecting* clustering). The same procedure is repeated for each of the two newly obtained clusters and recursively further down the hierarchy. We perform the splitting until the size of the clusters (i.e. the number of pages the cluster contains) is smaller than the predefined minimum size (usually set to 10% of the initial collection size). During the clustering process, the similarity between two vectors is computed as the cosine of the angle between the two vectors.

The result of the clustering is a binary tree (in this text termed *topic ontology*), with a set of pages at each node. Later on, for each node a centroid is computed in the same way as for the short-term interest pages.

The root of the topic ontology holds the user's general interest while the leaves represent his/her specific interests. By our understanding the term *general interest* is not synonymous with *long-term interest* and in that same perspective the term *specific interest* is not a synonym for *short-term interest*. While the terms *long-term* and *short-term* (i.e. *recent* or *current*) *interest* emphasize the chronological order of page-views, this is not the case with the terms *general* (i.e. *global*) *interest* and *specific interest*. *General interest* stands for all the topics the user is – or ever was – interested in, while the term *specific interest* usually describes one more-or-less isolated topic that is – or ever was – of interest to the user.

3.4 Current Browsing Interest of the User

By using the cosine similarity measure, we are able to compare the centroid at each node to the short-term interest centroid. In other words, we are able to map the user's current interest to the topic ontology. The mapping reveals the extent to which a node (i.e. a set of pages) is related to the user's short-term interest. By highlighting nodes with the intensity proportional to the similarity score, we can clearly expose the topic ontology segments that are (or are not, for that matter) of current interest to the user.

Due to the highlighting the user can clearly see which parts of the topic ontology are relevant to his/her current interest. He/she can also access previously visited pages by selecting a node in the hierarchy which is visualized in the application window. This can be explained as the user's interest-focused Web browsing history, the interest being defined by the selected node.

4 Implementation of the System

The user profile is visualized on the Internet Explorer toolbar that we developed for this purpose. The user can select a node (i.e. his/her more or less general interest) to see its specific keywords and the associated Web pages.

4.1 Toolbar as the User Interface

Generally, an Internet Explorer (IE) toolbar is an extension of the IE's GUI, as well as an application that extends the IE with additional functions. Since it is highly integrated into the IE, a toolbar can also:
(i) receive notifications and information about the user's actions in the IE; most notably the user's requests to "navigate to" (the user's requests can be filtered or preprocessed in some other way),
(ii) access the contents of the currently loaded Web page,
(iii) apply any kind of changes to the content of the currently loaded page (e.g. highlight links to recommended pages, highlight some parts of the text, etc.),
(iv) easily access the Web as well as the local computer.

We have developed an IE toolbar to construct and visualize the user's interest-focused browsing history. The toolbar is placed into the left side of the IE's application window. It is divided into two panels, one showing the user's topic ontology and the other showing the most characteristic keywords and the set of pages corresponding to the selected node (see Figures 4 and 5). The user can select any page from the list and navigate to that page.

The user's current interests are highlighted (see screenshots in Picture 4) in the ontology visualization panel. The color intensity of the highlighting corresponds to the relevance of the node to the user's current interest. The user can thus clearly see which pages that he/she already visited are in the context of his/her current interest.

4.2 Example of the System Usage

We will demonstrate how the system works in a real-life scenario. Let us say that the user is interested in three distinct topics. He/she searches the Web for “whale tooth”, “triumph tr4” and “semantic web”, in this same order. After viewing several pages (55 altogether in our case), his/her topic ontology looks as shown in Figure 3.

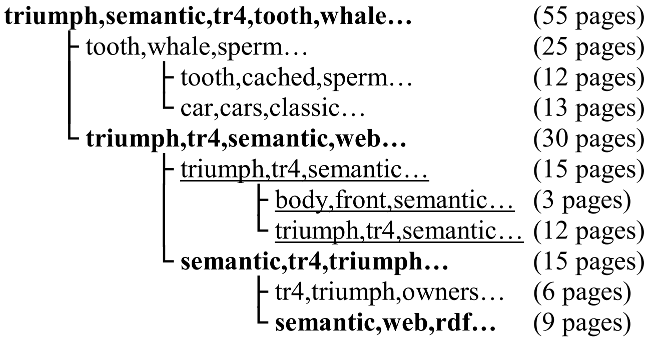
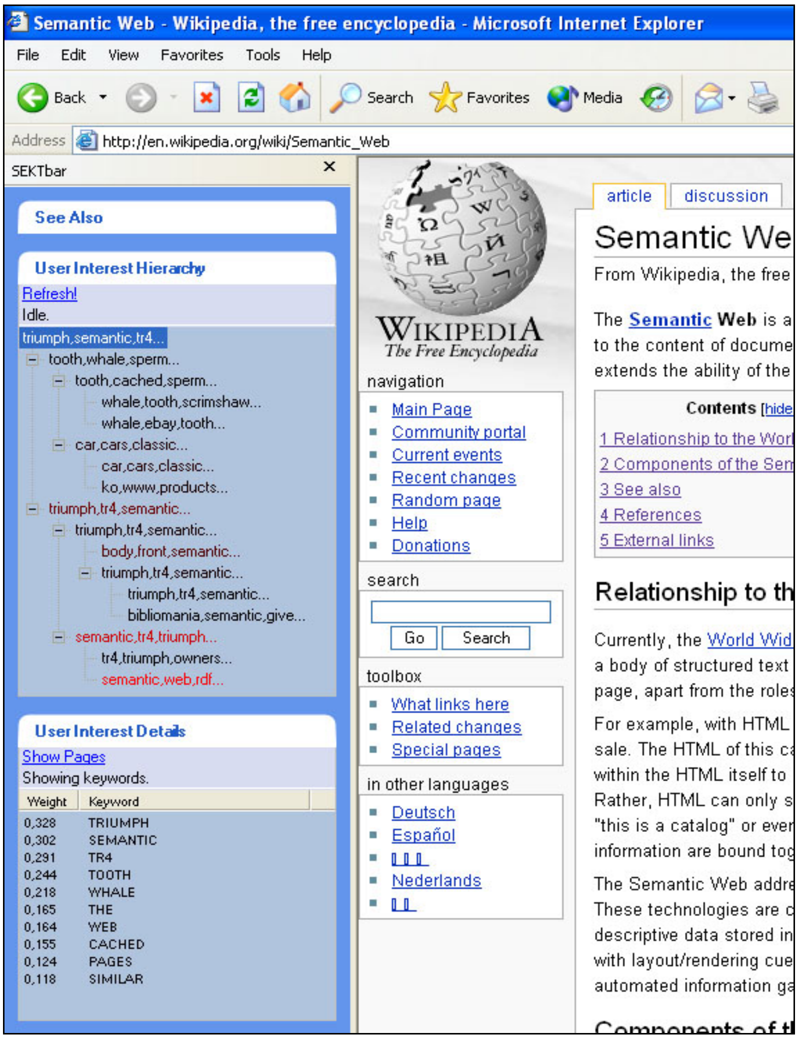

triumph,semantic,tr4,tooth,whale... (55 pages)
tooth,whale,sperm… (25 pages)
tooth,cached,sperm… (12 pages)
car,cars,classic… (13 pages)
triumph,tr4,semantic,web... (30 pages)
triumph,tr4,semantic… (15 pages)
body,front,semantic… (3 pages)
triumph,tr4,semantic… (12 pages)
semantic,tr4,triumph... (15 pages)
tr4,triumph,owners… (6 pages)
semantic,web,rdf... (9 pages)

Figure 3. The topic ontology as automatically constructed after viewing 55 Web pages about “whale tooth”, “triumph tr4” and “semantic web”, in this same order.

Each node is named after the three keywords from the centroid vector that have the highest averaged TFIDF weights. The root node represents the user’s general interest – they appear to be about *tooth*, *whale*, *triumph*, *tr4*, *semantic* and *web*, which is exactly what the user searched for. Note that the user’s search-engine queries are not included in the profiling process and that these keywords were actually reconstructed from the textual contents of the pages that the user visited.

The root node is first partitioned into two clusters – one containing the pages about *whale tooth* and the other containing the pages about *triumph tr4* and *semantic web*. We can see that the partitioning is not perfect. The cluster talking about *classic cars*, for example, is contained within the *whale tooth* cluster. It would make more sense if it was included into the *triumph tr4* cluster. Furthermore, we see that the second cluster (*triumph tr4 & semantic web*) is not clearly partitioned into the *triumph tr4* cluster and the *semantic web* cluster in the next step. However, since we are using fully automated methods, we can say that the results are reasonably good.

Since *semantic web* was the user’s latest interest, the nodes containing mainly pages related to this topic are highlighted (in Figure 3 bolded or underlined). We can see that highlighting works quite well in this particular example. Bolded clusters are highly relevant, underlined clusters are less relevant, and other clusters are irrelevant to the user’s current interest. Two screenshots of the system’s GUI are given in Figure 4 and Figure 5.



Address: http://en.wikipedia.org/wiki/Semantic_Web

Idle.

```
triumph,semantic,tr4...
⊟ tooth,whale,sperm...
   ⊟ tooth,cached,sperm...
      whale,tooth,scrimshaw...
      whale,ebay,tooth...
   ⊟ car,cars,classic...
      car,cars,classic...
      ko,www,products...
⊟ triumph,tr4,semantic...
   ⊟ triumph,tr4,semantic...
      body,front,semantic...
      ⊟ triumph,tr4,semantic...
         triumph,tr4,semantic...
         bibliomania,semantic,give...
   ⊟ semantic,tr4,triumph...
      tr4,triumph,owners...
      semantic,web,rdf...
```

Showing keywords.

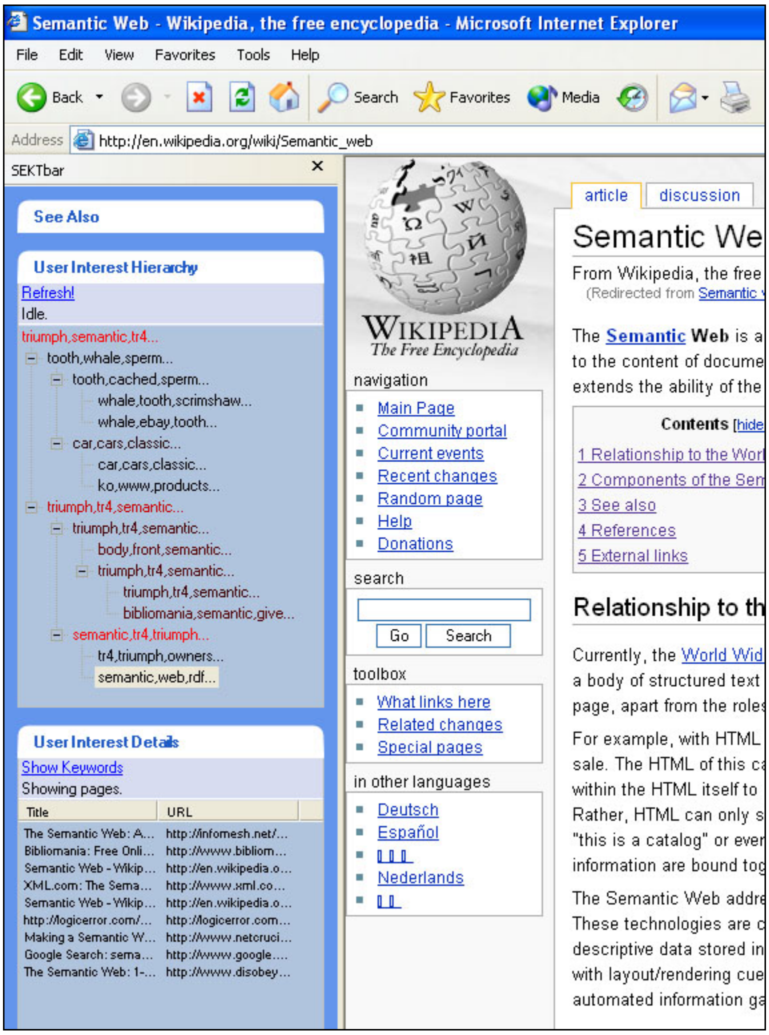
Weight	Keyword
0,328	TRIUMPH
0,302	SEMANTIC
0,291	TR4
0,244	TOOTH
0,218	WHALE
0,165	THE
0,164	WEB
0,155	CACHED
0,124	PAGES
0,118	SIMILAR

Semantic We
From Wikipedia, the free
The **Semantic Web** is a
to the content of docume
extends the ability of the

1 Relationship to the Worl
2 Components of the Sem
3 See also
4 References
5 External links

Relationship to th
Currently, the World Wid
a body of structured text
page, apart from the roles
For example, with HTML
sale. The HTML of this ca
within the HTML itself to
Rather, HTML can only s
"this is a catalog" or even
information are bound tog
The Semantic Web addre
These technologies are c
descriptive data stored in
with layout/rendering cue
automated information ga
Components of th

Figure 4. Screenshot of the system's GUI, captured after the user visited all the pages used for the demonstration in Section 4. Screenshot shows the topic ontology of the user's interests and the most characteristic keywords from the root cluster. The user's most recent interest is highlighted with red color (the brighter the more relevant).



- triumph,semantic,tr4...
  - tooth,whale,sperm...
    - tooth,cached,sperm...
      - whale,tooth,scrimshaw...
      - whale,ebay,tooth...
    - car,cars,classic...
      - car,cars,classic...
      - ko,www,products...
  - triumph,tr4,semantic...
    - triumph,tr4,semantic...
      - body,front,semantic...
      - triumph,tr4,semantic...
        - triumph,tr4,semantic...
        - bibliomania,semantic,give...
    - semantic,tr4,triumph...
      - tr4,triumph,owners...
      - semantic,web,rdf...

Title	URL
The Semantic Web: A...	http://infomesh.net/...
Bibliomania: Free Onli...	http://www.bibliom...
Semantic Web - Wikip...	http://en.wikipedia.o...
XML.com: The Sema...	http://www.xml.co...
Semantic Web - Wikip...	http://en.wikipedia.o...
http://logicerror.com/...	http://logicerror.com...
Making a Semantic W...	http://www.netcruci...
Google Search: sema...	http://www.google....
The Semantic Web: 1-...	http://www.disobey...

Semantic We

From Wikipedia, the free
(Redirected from Semantic w

The **Semantic Web** is a
to the content of docume
extends the ability of the

Relationship to th

Currently, the World Wid
a body of structured text
page, apart from the roles

For example, with HTML
sale. The HTML of this ca
within the HTML itself to
Rather, HTML can only s
"this is a catalog" or even
information are bound tog

The Semantic Web addre
These technologies are c
descriptive data stored in
with layout/rendering cue
automated information ga

Figure 5. Screenshot of the system's GUI, captured after the user visited all the pages used for the demonstration in Section 4. Screenshot shows the topic ontology of the user's interests and the list of Web pages that corresponds to the *semantic-web-rdf* cluster. The user's most recent interest is highlighted with red color (the brighter the more relevant).

5 Discussion

Many research issues and technical details still need to be investigated. We noticed that when extracting the textual content of a Web page, a lot of interest-irrelevant text segments are also processed (e.g. standard navigation menus and adds). A simple heuristic that could be used to alleviate the problem is discarding text segments (i.e.

chunks of text between two HTML tags) that are shorter than some predefined length. This solution has not been applied yet but we are planning to try it out in near future.

Since our software resides on the client side and we are able to track the Web browser's events, we could also efficiently measure the time the user spends on a page and use this information to additionally weight pages that were viewed by the user. In this same context, the pages that were visited more than once should be weighted by the sum of their page-view durations.

Currently we treat all Web pages equally. In the future, we should identify pages that are not suitable for the user profiling process. Such pages may be Web mail pages, search engine results and portal entry pages. To weaken the negative impact of such pages on the user profile construction, we could extend our stop-word collection with most frequent common Internet words. Another approach would be to allow the user to specify URLs (in the form of regular expressions, for instance) that should be excluded from the profiling process.

There is some work on document profiling that extends the vector representation by using word sequences (also termed *n-grams*) instead of single words (MLADENIĆ AND GROBELNIK, 2003). This work suggests that using single words and also word pairs for features in the vector representation of short documents improves the accuracy with which these documents are classified. We should incorporate these findings into our TFIDF vector representation of Web pages.

In our current implementation we are using the *nearest neighbor* approach to map the current interest to the topic ontology. Other more sophisticated machine learning techniques might provide better results in this process (e.g. classification with Naïve Bayes or SVM).

In this implementation, each time a page is viewed, the entire profile is rebuilt from scratch. We need to consider ways to update the topic ontology rather than rebuild it.

The clustering method used was not evaluated. We need to define evaluation methods for the profile generation process and, on the other hand, for the page classification process. This is not a trivial task and needs to be investigated in great detail. Once we are able to evaluate the algorithms, we will also be able to apply other approaches and see how they measure up to the one described in this paper.

The system was not tested in a real-life scenario. We should carry out an experiment involving test-users to see how useful the system really is.

Acknowledgement

This work was supported by the 6FP IP SEKT (2004–2006) (IST-1-506826-IP) and the Slovenian Ministry of Education, Science and Sport. The authors would also like to thank Tanja Brajnik for help in improving the overall clarity of this text.

References

1. KIM, H. R., CHAN, P. K. (2003): Learning Implicit User Interest Hierarchy for Context in Personalization.
2. PAZZANI, M., BILLSUS, D. (1997): Learning And Revising User Profiles: The Identification of Interesting Web Sites.
3. CHAN, P. K. (1999): A Non-Invasive Learning Approach to Building Web User Profiles.
4. BILLSUS, D., PAZZANI, M. J. (1999): A Hybrid User Model for News Story Classification.
5. SUGIYAMA, K., HATANO, K., YOSHIKAWA, M. (2004): Adaptive Web Search Based on User Profile Construction without Any Effort from Users.
6. MIDDLETON, S. E., SHADBOLT, N. R., DE ROURE, D. C. (2003): Capturing Interest through Inference and Visualization: Ontological User Profiling in Recommender Systems.
7. ADOMAVICIUS, G., TUZHILIN, A. (1999): User Profiling in Personalization Applications through Rule Discovery and Validation.
8. WASFI, A. M. (1999): Collecting User Access Patterns for Building User Profiles and Collaborative Filtering.
9. SALTON, G., BUCKLEY, C. (1987): Term Weighting Approaches in Automatic Text Retrieval. *Technical Report, COR-87-881, Department of Computer Science, Cornell University*.
10. STEINBACH, M., KARYPIS, G., KUMAR, V. (2000): A comparison of Document Clustering Techniques. In: *Proceedings of KDD-2000 Workshop on Text Mining, 109–110*.
11. MLADENIĆ, D., GROBELNIK, M. (2003): Feature Selection on Hierarchy of Web Documents. In: *Journal of Decission Support Sysems, 35, 45–87*.

Towards Overcoming Limitations of Community Web Portals: a Classmates' Example

Anna V. Zhdanova

DERI – Digital Enterprise Research Institute,
University of Innsbruck, Austria, and
National University of Ireland at Galway, Ireland

anna.zhdanova@deri.org

Abstract. Current distributed ontology practices are analyzed and illustrated with typical web portals supporting communication, data sharing and activities of former classmates. The inflexibility and restrictions imposed on users of such portals are demonstrated to support the thesis that introduction of community-driven ontology management is crucial for full-fledged satisfaction of the end user needs on the Web.

1. Introduction

An idea of providing a service for reunion of ex-classmates is proved to be a success by resulting in a large number of highly popular web portals with a multitude of users registered at the largest portals. For instance, more than 75 thousands of classmate groups are registered internationally at Yahoo groups[1] and more than 35 millions of users are registered at a national US and Canadian level (portal Classmates.com uniting graduates of the US and Canadian schools). In relation to other commercial services offered on the Web, the service providing a communication environment for ex-classmates also proved to be promising. For instance, the portal Classmates.com has one of the largest subscriber bases on the Web for paid content and is consistently ranked by Nielsen//Net Ratings as one of the most highly trafficked Web channels.

One of the success reasons for social networking activities across ex-classmates and other user groups [10] is that the portals supporting these activities fill in a novel niche of user demand. Specifically, many e-commerce sites offer what people have always been able to find outside their front doors: books, magazines, toys and groceries. Compared to most online businesses, community web portals are privileged to offer a service that only the Web can provide: the power to connect people who would otherwise be out of touch.

[1] Yahoo Groups: http://groups.yahoo.com

We define ***ex-classmates*** as a group of people who once had a common educational experience and used to live in the same area. We use the term ***classmates*** equivalently to the term “ex-classmates”, because people who once studied together and lived in the same area can be identified as belonging to the same “class”. Specifically, from the virtual community point of view, whether the community is united physically by the past or by the present is usually irrespective for modeling community activities. A ***community Semantic Web portal*** is a web portal which is based on Semantic Web technologies [1] and maintained by a community of users. Further, a ***web portal*** is a web site that collects information for a group of users that have common interests [5]. Yahoo is an example of a web portal, however, Yahoo is not a community web portal, since (i) it is resource consuming and anti-collaborative in providing information, (ii) it is maintained by a special department of the host company, but not by a community of users.

Nowadays, with an exception of few cases, existing community web portals are not Semantic Web portals. We demonstrate below that they suffer from a lack of flexibility, missing functionalities, data input overhead and sparse interactivity. These problems are expected to be resolved by employing technologies constituting community Semantic Web portals. In the Semantic Web, information is semantically represented according to ontologies, evolving and shared knowledge structures, allowing advanced usage of the Internet as an information repository [4]. Further, enabling the Semantic Web to be ***community-driven***, i.e., endowing users and developers with a wide access to ontology management, will make the community Semantic Web portals more dynamic and more responsive to the users’ actual needs.

An extensive overview and state-of-art of existing Semantic Web portals is delivered by Lausen et al. [9]. An approach embedding all phases of a community Web portal (i.e., information accessing, information providing, portal development and maintenance) is described in a paper by Staab et al. [15]. Our work is focused on the existing classmates’ portals. Demonstrating the limitations of available solutions, we show the need for development of Semantic Web content and services.

The paper is organized as follows. In Section 2, an overview of existing classmate web portals is provided and usage scenarios are discussed, highlighting a self-assessment scenario. In Section 3, community-driven ontology management is introduced. Observed limitations of the classmates’ community web portals are described in Section 4. Section 5 concludes the paper.

2. Overview and Scenarios

In this section, we provide an overview of the web portals supporting communities of classmates and outline scenarios at these portals. We highlight the scenario of self-assessment in order to illustrate the complexity behind a thorough support of community scenarios and to show the possibility of applying solutions across domains and communities. In particular, the solutions developed for personnel evaluation can be easily applied to the self-assessment classmates' scenario, provided that the solutions are available as services on the Semantic Web.

2.1. Overview

A summary of typical community web-portals that are created for support of classmate communities is given in Table 1. Geographical coverage and functionality of a portal are important characteristics defining the portals' applicability and usage. Geographical coverage in the context of the classmate portals is its geographical restrictions regarding the countries and cities where ex-classmates used to study. Most observed classmate portals are restricted geographically, i.e., they permit a correct representation of the fact that "somebody studied somewhere" only for restricted values of "somewhere". In Table 1, examples of such national classmate portals are Classmates.com, Odnoklassnik and ILoveSchool. Additional examples are www.passado.de (Germany), www.passado.fr (France), www.passado.at (Austria) and www.chinaren.com (China). At each of these national portals, information is represented solely in the national language of an addressed country. Analyzing functionality of the classmates' portals reveals a tendency of decreasing the portals' functionality with increasing the portals' coverage (including geographical). For instance, Lycos Classmates, Yahoo groups or a widely international alumni portal www.alumni.net provide less of functionality comparing to any other of the national-targeted portals listed in Table 1.

2.2. Self-Assessment Scenario

The demand for self-assessment is often a driving force of a substantial amount of communication activities which take place within the classmate portals. Members of the classmates' portals have a demand for information on positions held by their classmates, what they have acquired, etc. Examples of typical queries for this information are: "What kind of job do you have?", "How many children do you have?", "What kind of car do you drive?". After obtaining answers to their queries, the portal users can evaluate their position and achievement in relation to their classmates' positions and achievements. Self-assessment in relation to one's classmates is often found especially meaningful, due to possession of the similar background and the same starting point.

Self-evaluation support within the classmate portals can be addressed reusing solutions from a job-related domain for the problems such as personnel selection, personnel management, personnel evaluation, assessment of staff's performance. For the emerging services such as self-assessment in the classmates' scenario, reuse of the well-elaborated solutions in similar areas is especially beneficial.

Name	**URL**	**Geographical Coverage**	**Functionality**
Classmates.com	www.classmates.com	USA, Canada and American / Canadian overseas schools	Registration/search, message board, games, chat, photos sharing, "compare" tool, shopping
Lycos / Classmates	www.lycos.com	International – over 40 countries	Registration/search
Odnoklassnik	www.odnoklassnik.ru	Russia	Registration/search, addresses, telephone and ICQ numbers, photos sharing, message board, chat, polls
ILoveSchool	www.iloveschool.co.kr	Korea	Registration/search, mailing lists, games, whiteboard, news of school, avatar, SMS, shopping

Table 1: Web Portals for Classmates

Conventionally, evaluation of job performance can be ***trait-based***, ***behavior-based*** or ***result-based***. Trait-based criteria focus on the personal characteristics of an employee, behavior-based criteria focus on specific behaviors that lead to job success when exhibited, and results-based criteria focus respectively on what an employee has done or accomplished [11]. In addition, evaluation of job performance can employ ***objective*** and ***subjective*** measures. Objective measures are quantifiable measures of performance (e.g., cars/hour, bottles/second, etc.), while subjective measures are less quantifiable (e.g., leadership, presentation, etc.). Another opportunity to classify evaluation systems is to track whether they evaluate somebody's performance on the ***absolute*** scale or ***comparatively*** to other performances.

Normally, a typical personnel evaluation system considers one of the criteria for evaluation of job performance, objective and/or subjective measure and a particular (absolute and/or relative) scale for evaluation of personnel. The approaches for realization of the evaluation systems vary substantially.

For example, BOARDEX [7] is an expert system for selection of the candidates to attend military schools. Evaluation performed by the system is result-oriented, dominantly with objective measures with absolute and relative scales. Knowledge representation of the BOARDEX system is accomplished using Prolog and the selection process is performed by applying rules which check each candidate's resume with respect of several important for military school factors such as height, weight, military education, assignment history, etc. and produce a recommendation on the acceptability of the candidate. The system was reported to attain highly significant correlations and evaluation concordances with the human experts, justifying chosen methodology. Shaout and Al-Shammari [14] describe another expert system which is based on fuzzy logic and performs evaluation of faculty members in an academic department at the educational institution. This system evaluates personnel against behavior and result-based criteria using objective evaluation measures and an absolute scale in order to assign human resources to the goals of the institution.

Herrera et al. [6] apply a genetic algorithm for a personnel assignment task (when the number of positions equals to the number of the candidates) and for a personnel selection task (when the number of candidates is greater than the number of positions). The evaluation factor values are represented as linguistic variables for each candidate. At first, the candidates are assigned randomly at the positions, then a selection mechanism and specific genetic operators such as crossover and mutation are applied to refine the result. The methodology employed in the system is based on trait-based criteria, subjective evaluation measure and relative scale.

In contrast to the outlined above methodologies, assessment of expertise level does not have to necessarily employ representation of personnel skills, results achieved, traits or behaviors, but can rely solely on the behavior of would be experts by using their performance in the domain [13]. Specifically, the approach by Shanteau et al. relies on checking whether a person whose level of expertise is being evaluated demonstrates discrimination and consistency, i.e., if he/she is able to differentiate between similar but not identical cases and repeat his/her judgment in a similar situation. Thus, the proposed approach is behavior-based, employing the objective evaluation measure and the absolute scale. This approach for expertise evaluation is especially appropriate in the absence of a widely accepted standard, when one can not compare experts against the standard and select whoever is closest to the standard.

As a reply to the demand of self-assessment in the classmates' scenario, the Classmates.com portal offers a special tool: the user can answer suggested questions and compare his/her answers to the answers of his/her classmates, represented in percents. Naturally, the questions that can be asked at different portals may vary, depending on the creators of the portals. For example, the questionnaire of Classmates.com portal covers five subjects: leisure/vacations (7 questions), family/relationships/children/home (5 questions), financial status (4 questions), feelings/opinions about life (4 questions), the Classmates.com portal services (4 questions).

3. Ontology Management: from Distributed to Community-Driven

There are examples of ontologies that became widely accepted and reused for the purpose of distributed data exchange and integration. Specifically, RDF, FOAF, Dublin Core and RDFS vocabularies are the most successful with being populated by more than one million of Web documents each [3]. Very often such ontologies were organically grown and quickly found a large number of creative users, even though a for long time they were not endorsed by any of the popular standards committees.

Meanwhile, the amount of available ontologies for reuse and sharing is practically very limited. For example, SchemaWeb[2] is nowadays is an exhaustive resource for publishing ontologies and it links to ca. 250 ontologies only. This quantity of available ontologies refers to ontologies specified in multiple existing different ontology languages (e.g., RDFS, OWL). Many of these ontologies are not supported by a large amount of instance data. The linked ontologies are mostly vocabularies describing limited specific domains (e.g., Person, Publication, Project). Some domains are supported by several ontologies (e.g., Person and Publication), while many domains are not supported by ontologies at all. Finally, the number of domain-independent (functional) ontologies that can be widely applied is negligent, and ontologies for certain aspects like Semantic Web publishing, data delivery and community and personalization support are not available. All these factors diminish ontology usage and thus success of the Semantic Web.

The limitations of centralized ontology development display the need for dynamically extendible large-scale ontologies with distributed character. For example, the RSS working group states that as RSS continues to be re-purposed, aggregated, and categorized, the need for an enhanced metadata framework grows. Channel- and item-level title and description elements are being overloaded with metadata and HTML. Some producers are even resorting to inserting unofficial ad-hoc elements (e.g., <category>, <date>, <author>) in an attempt to augment the sparse metadata facilities of RSS.

The other communities who appreciate usefulness and value of RSS also report that it has reached its limits. There is a demand for more advanced portal syndication which RSS can not satisfy. One initiative in developing technologies to overcome the limitations of simple ontologies for Web publishing comes from Apache Software Foundation and proposes portal syndication with Web services and Cocoon [8]. Another initiative is Atom[3] that is aimed to define a feed format for representing and a protocol for editing Web resources such as Weblogs, online journals, Wikis, and similar content. The feed format is to enable syndication, and the editing protocol is to enable agents to interact with resources by nominating a way of using existing Web standards in a pattern. Overcoming the limits of distributed small-scale ontologies, organization of user-driven ontology

[2] SchemaWeb: http://www.schemaweb.info

[3] AtomEnabled: http://www.atomenabled.org

extension, support and metadata communication within Web portals is generally considered in the approach of the People's portal [16].

The reasons why staying within the scope of simple ontologies (e.g., exchanging FOAF profiles and posting cross linked news stories from RSS) is not enough and far too limited for the existing Web are as follows:

- embedding and personalizing rich content and behavior from remote Web applications are becoming necessity for catering to specific user needs
- extension of simple ontologies, discovery and communication of these extensions are becoming necessity for bringing semantics to a larger amount of Web content
- mapping between simple ontologies and their alignment with other extendible ontologies are becoming necessity for large–scale data integration.

The introduced solutions by the RSS working group to handle the RSS limitations are as follows. One proposed solution is the addition of more simple elements to the RSS core. This direction, while possibly being the simplest in the short run, sacrifices scalability and requires iterative modifications to the core format, adding requested and removing unused functionality. A second solution, and the one adopted in the RSS specification, is the compartmentalization of specific functionality into the pluggable RSS modules. This is one of the approaches used in this specification: modularization is achieved by using XML Namespaces for partitioning vocabularies. Adding and removing RSS functionality is then just a matter of the inclusion of a particular set of modules best suited to the task at hand. No reworking of the RSS core is necessary.

Obviously, the problems and solutions for RSS ontology above are also valid for other simple widely spread ontologies. Having simple and easy to understand ontologies and ontology pluggable extensions on the user side, the complex processes of combination and reuse of these ontology components in ever-changing specification and conceptualization processes of the outside world are left encapsulated on the middleware and application side. Clearly, the development and especially reuse of the pluggable extension modules involve complex problems that are not resolved at the moment. These problems arise from the support requirements for practical large-scale extendible ontology management, such as:

- easy and quick extension opportunity to cater to dynamically arising and changing needs of ontology users
- discovery of existing pluggable extension modules
- composition of existing pluggable extension modules
- decomposition of existing pluggable extension modules
- matching of existing pluggable extension modules and core ontologies with other external ontologies and modules

- tools to support ontology extensions proposed from the user's side, discovery, composition, decomposition, matching and reuse of created earlier ontologies and extensions.

Thus, preserving the successful approach of simple usable ontologies and resolution of the issues above are clearly to be considered as major challenges in the practical state-of-the art distributed ontology management, and are addressed with creating supporting infrastructure for community-driven ontology management.

Specification and development of ontology management components were previously funded and carried out in USA and EU projects (in particular, EC IST projects such as DIP[4], SEKT[5], KnowledgeWeb[6], Esperonto[7], SWWS[8]). Progress in development of community Semantic Web environments brings in new positive influence, usage scope and wider acceptability to the basic ontology management components by setting new requirements such as enabling communities manage their own ontologies, making the ontology management knowledge services more flexible, reusable and proven in real-life scenarios thus attractive enough to make the Semantic Web accepted by the communities.

The scope of the work on community-driven ontology management is in reuse of the existing ontology management practices and tools and enriching them with features for supporting end users and communities to describe and manage community Web portals. One may envision ontology management support consisting of the following components adapted within the scope of community-driven ontology management:

Community-Driven Ontology Editing Service: It is an editor for editing ontologies (creating and updating ontology and instances). The front end is the user-friendly interface, which helps or guides users to easily create and update (add, delete, and modify) ontology and its instances. The backend is the data storage management systems, which can be databases, file systems, plain text files. A specific requirement for an ontology editor to be community-driven is an opportunity to integrate it tightly with Semantic publishing and delivery component, and enable consensual editing for multiple users, i.e. communities. This requirement is grounded on flexibility degree that is needed to provide in a community environment enabling community members to change and influence community processes and structures.

Community-Driven Ontology Storage and Query Management Service: The goal of this component is to efficiently store and query small and large amounts of ontology data and metadata by providing fast indexing, searching and querying to ontologies and its instances. Most current ontology storing and querying components from the functional perspective are similar to database and database

[4] DIP: http://dip.semanticweb.org

[5] SEKT: http://sekt.semanticweb.org

[6] KnowledgeWeb: http://knowledgeweb.semanticweb.org

[7] Esperonto: http://esperonto.semanticweb.org

[8] SWWS: http://swws.semanticweb.org

management system components. In addition, the first Semantic Web search engines start to appear (such as Intellidimension Semantic Web search engine[9]). However, there is a long road to go to making Semantic Web database-like components and Semantic Web search engines mature and attractive to use. Taking into account that the communities publish their information on the Semantic Web in a distributed manner in simple ways (such as putting online FOAF files), in project work, the focus in storage and querying will be on maintaining repositories of reusable adding value Semantic Web content and composition/decomposition of distributed source content that is easy to maintain from the storage and creation point of view, thus involving critical community masses.

Community-Driven Ontology Alignment Service: A regular ontology aligner supports ontology mapping processes that now mostly are performed manually, e.g., OWL Ontology Aligner[10]. A basic ontology inference provides consistency checking, related class or relation name identification, instance updates etc. The front end is the user interface for semi-automatic ontology mapping (such as recommendation lists and help for defining the mapping rules). The back end is the inference support (ontology inference engine). The upgrade of a regular ontology aligner to a community ontology aligner is adding a widely available repository of ontology mapping solutions that result from the usage of the ontology aligner. Special ontologies are used to specify relevance, reusability and reliability of certain ontology mappings from repositories (employing social networking and statistical information). The ultimate goal of the community alignment service activity is to enable knowledge services of external applications to reuse (i.e., gain benefit from) these annotated mapping repositories and alignment services.

Community-Driven Ontology Versioning Service: The versioning service represents different versions of the ontologies, including backward consistency support and related instance versioning. The front end provides a report on version information, changes and their effects, for example, the difference of two versions of the ontologies. The back end supports backward consistency in the different versions of the ontologies and their instances update. The Ontology Versioning Service is to be interoperable with Ontology Editor, Ontology Storage and Query Manager and pluggable inference engines for performing additional optional tasks such as checking consistency. On top of the ordinary functionality of an ontology versioning service, a community versioning service needs to have a set of simple understandable interfaces, be available and easily accessible on the Semantic Web, and track the changes taking place in distributed ontologies and instance data sources, reporting relevant inconsistencies and its resolutions to community versioning service users.

[9] Intellidimension Semantic Web Search: http://semanticwebsearch.com

[10] OWL Ontology Aligner: http://align.deri.org

4. Limitations

In this section, I generalize typical limitations of the classmates' community web portals, and briefly outline the way to overcome these limitations via community-driven ontology management on the Semantic Web.

4.1. Overview

Observation of the functionality of the classmates' portals allows us to identify several limitations restricting their usage. These limitations are general enough to be applicable to existing web portals supporting different communities than classmates. The limitations are as follows.

Geographical restrictions

Most classmates' web portals have geographical restrictions, i.e., a classmate can register adequately only within a portal providing opportunities to state the fact that this classmate comes from a particular school of a particular country.

Absent or simplified functionalities

Most of the reviewed web-portals for interaction of classmates support very basic activities such as registration and search, but not the advanced activities such as maintenance of the common calendar to organize meetings or support of and access to a query service over the instances provided by portal members. Sometimes, the support for advanced activities is present at the classmates' web portals, but usually this facility is not extensive enough. For example, the compare-tool at the Classmates.com portal described in the previous section allows an user to compare his/her answers to the answers of other classmates using only one type of simple predefined queries. Specifically, the user is asked to choose his or her age group, gender and a particular question as the basis of comparison. Thus, for instance, finding out how many of your classmates of your gender and age have cats as home animals is possible, but finding out how many of your classmates of your age and gender live in the USA and have at least two children is impossible. This limitation arises because Classmates.com portal does not support construction and processing of queries with conjunctions or disjunctions. Therefore, in the light of existing personnel evaluation research described in the previous section, the state-of-art support of the self-assessment feature looks especially shallow on the classmates' community web portals.

Generality of services

Apart from the classmates' web portals such as the ones listed in Section 2, other web environments can partially satisfy demands of classmates' communities. For example, Yahoo Groups provide such groupware as registration of a group/group members, mailing-lists, chat, file/link sharing, voting, personal calendar. However, the Yahoo Groups' functionalities prove to be too general, as they are designed to support an environment for any group of people and thus comprise groupware items one can find anywhere else. Therefore, Yahoo Groups and similar general-purpose environments can hardly be considered as a perfect solution for communities of classmates, due to the lack of functionalities and services specifically interesting for these communities.

Data input overhead

Nowadays, a usual need to register and to log in for each web portal/environment every time their functionalities are required incurs overhead. The user has to enter the same personal information (e.g., name, surname, e-mail address, telephone number, etc.) multiple times for each of the different web portals used by him/her and permanently operate with multiple environments. Further, when a community member uploads an object (e.g., text file or image) to a community web portal supporting annotation of the objects (e.g., Microsoft SharePoint Portal Server[11]), most times he/she has to annotate the object manually by inserting data describing document in the form for each portal.

4.2. Overcoming the Limitations

To overcome the limitations of community web portals, the following milestones need to be passed:

Up-to-date annotations for people and objects

Corresponding to the Semantic Web vision, persons or objects should be provided with a machine-processible annotation that can be shared across applications. FOAF[12] and Dublin Core[13] are examples of wide-spread schemata for annotation of people and documents. Further, when certain properties of a person or object are changed (e.g., a person moves to a new flat), the change in the annotation needs to take place is communicated to the Semantic Web environments employing the changed (meta)data. This Semantic Web scenario has a potential to overcome the limitations of data input overhead, and has yet to be elaborated in details and achieved in the future on the broad scale. At present, even at the well-developed Semantic Web community web portals such as KnowledgeWeb[14], extensive data entering is required in order to register community members and introduce new objects for the community.

Access to weaving of the Semantic Web

Enabling wide communities of users and developers to introduce new ontology structures and services is crucial for Semantic Web to adapt to the actual users' needs and to spread widely [16]. An access to participation in the formation of the Semantic Web content is associated with *community-driven ontology management*, where ontology management actions (e.g., ontology editing,

[11] Microsoft SharePoint Portal Server: http://www.microsoft.com/sharepoint

[12] The FOAF project: http://www.foaf-project.org

[13] Dublin Core Metadata Initiatiative: http://dublincore.org

[14] KnowledgeWeb portal: http://knowledgeweb.semanticweb.org

versioning, storage, querying) are performed in a distributed fashion by the users' and developers' communities, in addition to a limited group of web-resource creators and domain experts as conventionally. Letting the communities to weave their own Semantic Web will mitigate such current limitations as geographical and natural language restrictions, absent and simplified functionalities, generality of services.

Community-driven ontology/process alignment

Thus, As the Semantic Web becomes easily and widely extendable, many similar schemata and processes will be developed and maintained by different communities. Under these circumstances, the ability to easily align and combine similar or complementing schemata and processes is of crucial importance for cross-community interoperability. For instance, a person may belong to several communities and employ several Semantic scheduling services, e.g., as the service developed by Payne et al. [12]. Meanwhile, the scheduling services will be helpful to the person only in case of their interoperation, i.e., when making timing proposals, reporting the conflicts in the person's schedule, etc. is done considering the information in the range of all the scheduling services employed by a person. Community-driven ontology/process alignment has a potential to resolve such limitations as geographical restrictions and absent and simplified functionalities by combining or composing available services in personalized, required services.

Semantic desktop

Once the people/objects and processes are being annotated, the Semantic Web is easily extended by the communities of users and developers, and similar and complementing ontologies and processes can be aligned by individuals, presenting massive volumes of Semantic content and workflows to the community members is a major challenge. The solution is expected to stem from the active research fields in the Semantic Web area. For example, Decker and Frank [2] address this problem by combining the current Semantic Web developments in a *Social Semantic Desktop*, which will let individuals collaborate at a much finer-grained level as is possible and save time on filtering out marginal information and discovering vital information. Organizing Semantic Web content and services in personalized, cross-linking and supporting communities Semantic Desktop is the final step in overcoming limitations typical for the current community web portals.

5. Conclusions

Within a domain of ex-classmates' portals, the limitations of existing community web portals are identified. The analysis of the scenarios in the selected domain in general and of the self-assessment scenario in particular reveals an added value in combination of solutions across domains and communities where similar problems are addressed. Moreover, the examples of this paper illustrate that solutions developed for communities substantially vary even within one domain. Therefore, an infrastructure for community-driven ontology management is needed to for timely capture and alignment of the end user and developer efforts. Community-aware approaches such as evolution of Semantic Web annotations with respect to their usage, broad accessibility to creation of Semantic Web content and services, community-driven ontology management and alignment of efforts, and semantic desktop have a high potential to overcome the limitations of the current community web portals.

Acknowledgements

The author thanks Dieter Fensel, Stefan Decker, Sung-Kook Han and Ina O'Murchu for useful discussions. The work is partly funded by the European Commission under the NoE KnowledgeWeb (IST-2004-507482).

References

1. Berners-Lee, T., Hendler, J., Lassila, O., 2001. The Semantic Web. Scientific *American 284(5)*, pp. 34-43.
2. Decker, S., Frank, M.R., 2004. The Networked Semantic Desktop. In *Proceedings of the WWW Workshop on Application Design, Development and Implementation Issues in the Semantic Web*.
3. Ding, L., Zhou, L., Finin, T., Joshi, A., 2005. How the Semantic Web is Being Used: An Analysis of FOAF Documents. In *Proceedings of the 38th International Conference on System Scien*ces, January 2005.
4. Fensel, D., 2003. *Ontologies. A Silver Bullet for Knowledge Management and E-Commerce*. 2nd Edition. Berlin, Heidelberg: Springer.
5. Heflin, J., 2003. *Web Ontology Language (OWL) Use Cases and Requirements*. W3C Working Draft, 31 March 2003.
6. Herrera, F., Lopez, E., Mendaña C., Rodríguez, M.A., 2001. A linguistic decision model for personnel management solved with a linguistic biobjective genetic algorithm. *Fuzzy Sets and Systems 118 (2001)*, pp. 47-64.
7. Hooper, R.S., Galvin, T.P., Kilmer, R.A., Liebowitz, J., 1998. Use of an expert system in a personnel selection process. *Expert Systems With Applications 14 (1998)*, pp. 425-432.
8. Ivanov, I., 2004. *Portal Syndication with Web Services and Cocoon*. 1.0 Technical document.
9. Lausen, H., Stollberg, M., Lara, R., Ding, Y., Han, S.-K., Fensel, D., 2003. *Semantic Web Portals – State of the Art Survey*. Technical report, DERI-TR-2004-04-03. URL: http://www.deri.at/publications/techpapers/documents/DERI-TR-2004-04-03.pdf.

10. O'Murchu, I., Breslin, J.G., Decker, S., 2004. Online Social and Business Networking Communities. In *Proceedings of ECAI Workshop on Application of Semantic Web Technologies to Web Communities*.

11. Mathis, R.L., Jackson, J.H., 1994. *Human Resource Management*. Minneapolis/St. Paul, MN: West Publishing Company.

12. Payne, T.R., Singh, R., Sycara, K., 2002. Processing Schedules Using Distributed Ontologies on the Semantic Web, 2002. In: Bussler, C. et al. (Eds.), *Proceedings of Web Services, E-Business and Semantic Web Workshop*, CAiSE 2002, pp. 203-212.

13. Shanteau, J., Weiss, J.D., Thomas, R.P., Pounds, J.C., 2002. Performance-based assessment of expertise: How to decide if someone is an expert or not. *European Journal of Operational Research 136 (2002)*, pp. 253-263.

14. Shaout, A., Al-Shammari, M., 1998. Use of an expert system in a personnel selection process. *Expert Systems With Applications 14 (1998)*, pp. 323-328.

15. Staab, S., Angele, J., Decker, S., Erdmann, M., Hotho, A., Maedche, A., Schnurr, H. -P., Studer, R., Sure, Y., 2000. Semantic Community Web Portals. *Computer Networks 33 (2000)*, pp. 473-491.

16. Zhdanova, A.V., 2004. The People's Portal: Ontology Management on Community Portals. In *Proceedings of the 1st Workshop on Friend of a Friend, Social Networking and the Semantic Web (FOAF'2004)*, 1-2 September 2004, Galway, Ireland, pp. 66-74.

Supporting User Tasks and Context: Challenges for Semantic Web Research

Tom Heath, Martin Dzbor and Enrico Motta

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

Abstract. Whilst the tasks users perform online are often complex and wide-ranging, the tools currently available may not adequately support them. Attempts to classify user behaviors online have tended to focus on the medium of the web, where searching and browsing are seen as the primary modes of interaction. This paper introduces a comprehensive user-oriented classification of online tasks that emphasizes the user's goals without assuming the use of particular internet tools or technologies. Taking greater account of a user's context is also discussed as an essential component in better supporting performance of tasks online. Finally we consider how Semantic Web technologies can support the development of task-focused context-aware tools.

1 Introduction

The internet provides a platform for users to perform many varied tasks, such as finding information, exploring new ideas and communicating with others. In many circumstances this platform is immensely powerful and user tasks are well supported. For example, someone wanting to find large numbers of documents on a particular subject is likely to have success with regular search engines.

However, not all tasks that users perform (or may wish to perform) online are well supported by current tools and technologies. Consider the following scenarios:

1.1 Scenarios of Internet Usage

Locating a Book. Juan wants to buy a present for his cousin, and is looking for a book that Alice had read and recommended to him. He thinks the book is called "The Sergeant's Guitar", but he can't remember the author. Searching his favourite online bookshop for this title returns no results. Juan has to contact Alice, who tells him the book is actually called "Captain Corelli's Mandolin". With this clarification Juan is able to locate the book in the online bookshop and orders it for his cousin.

Arranging a Trip. Matt is arranging travel from his office in Liverpool to a conference being held in Slovenia. Using a travel web site he looks for flights from local airports to the Slovenian capital Ljubljana. Whilst some flights are available they are infrequent and expensive. Knowing that Adam has been to Slovenia before, Matt consults him for advice before making further plans. Rather than flying to Ljubljana, Adam recommends booking a cheap flight to Klagenfurt in Austria with a budget airline; from there frequent trains run across the border to Slovenia. Whilst the total journey time will be slightly longer, the tickets will be substantially cheaper than if he were to fly directly to Ljubljana.

On the conference web site Matt reads that there is a train station near the conference venue. He follows a link to the web site of the rail company, checks the online timetable, and finds that trains run directly to this station from Austria. Revisiting the conference web site he checks the list of recommended hotels and visits each of their web sites, but finds they're all full for the duration of the conference. He remembers that Adam recommended staying with a local family, and that a Tourist Office could arrange this. He locates the appropriate tourist office web site through a search engine and sends them an email explaining his requirements.

2 Problem Analysis

In the scenarios above, the users expend considerable time and attention in completing their tasks. Whilst the outcomes are generally successful, Juan and Matt encounter a number of obstacles along the way. Some of these obstacles pertain to the specific tools available to them, whilst others reflect wider issues of the technologies and architectures of the internet in its current form.

Query Precision. When Juan is unable to remember the exact title of the book he is looking for, the search engine on the bookshop web site isn't able to accommodate his imprecise query; it takes his query literally and returns no results, even though the terms he has entered bear a strong semantic relation to the real title of the book. As far as the search engine is concerned *captain* has no relation to *sergeant*, as the engine has no representation of the semantic links between terms, just of their linguistic syntax. Furthermore, it certainly isn't aware that Juan knows Alice, and that the "Captain Corelli's Mandolin" he is looking for is the same book that she reviewed favourably on her web log.

Manual Coordination. Planning his journey to the conference requires Matt to make separate arrangements with many different parties, each of which is largely unaware of his overall goal. The travel web site Matt originally uses can only provide information about flight routes he specifically requests. It is incapable of reasoning about alternative means of reaching the same destination, or of using knowledge held by Matt's social network to help complete the task. Similarly the airline is unaware of his final destination and so cannot automatically provide information about train connections from the airport. The tourist office may be aware that he'll be attending a conference if he mentioned it in his email, but they are unlikely to know that the conference starts early every day so his hosts will need to provide breakfast before 7am. Ensuring all of these conditions are met falls to Matt. Information about the task is not shared or reused, meaning he must explicitly state his requirements to each party and manually assemble information from the various sources if his task is to be completed. All other conference delegates must do the same.

In both these cases the user makes the best use of the tools available to them on the internet, even though these tools might not be well adapted to the true task the user is trying to perform. Furthermore, the same tools take little account of the user's context, such as the knowledge and previous experiences of those around them, when often this provides crucial assistance in performing a task.

3 Conceptualising User Tasks Online

To assess how well existing tools support users in completing tasks online, and how they might be better supported, it is important to understand the types of tasks people perform on the internet. The majority of literature in this area focuses specifically on the medium of the web rather than the internet as a whole, an issue discussed in greater detail below.

3.1 Web Activities as Forms of Searching and Browsing

Previous research has sought to identify and classify user behaviours on the web, mainly by identifying specific modes of searching or browsing. At the most basic level Guha, McCool and Miller [1] distinguish between *navigational* and *research* searches. In a navigational search "the user is using the search engine as a navigation tool to navigate to a particular intended document", whereas a research search is characterised by the user "trying to locate a number of documents which together will give him/her the information s/he is trying to find" (pp 702).

Broder [2] describes a taxonomy consisting of three types of web search: *navigational*, *informational*, and *transactional*. The navigational and informational types map closely onto the navigational and research searches proposed by Guha et al [1], with transactional searches consisting of queries where the user intends "to reach a site where further interaction will happen" (pp 6), such as a shopping site or a site where images or music can be downloaded. However, the range of possible transactions a user may wish to perform, and the underlying reason for wishing to perform them is not explored.

Related work by Rose and Levinson [3] yielded top-level categories with many similarities to those of Broder [2], but also a number of more detailed sub-categories such as *download*, *entertainment*, *interact*, and *obtain*. Despite a number of examples being given to illustrate these sub-categories, the distinctions between them are often based on technical aspects of how the target object will be used, rather than the nature of the task the user is performing. For example, the target of the *download* goal is "a resource that must be on my computer or other device to be useful" (pp. 15). The authors give the example of a piece of software, however the same definition could equally apply to the adult movie example used to illustrate the *entertainment* sub-category. In both cases it appears the user is trying to locate something that they can then make use of irrespective of how this is done.

One common factor in these studies is a search- or browse-centric perspective on web use. These "two predominant interface modes" [4] (pp 177-178) are often taken as the window through which to study user actions on the web. However, such a perspective may prevent a real understanding of the user's goals in being online. In the scenarios described above, the users have very clear tasks they wish to perform. To what extent can the classifications outlined here accommodate these tasks?

In the *locating a book* scenario, Juan is trying to locate a book that he knows exists and he uses the search engine on the bookshop web site to try and do so. This could be seen as analogous to the navigational searches proposed by Guha et al [1] where the user tries to locate a known document, or by Broder [2] where the user is searching for the web site of a known organisation or individual. In this case the target is a book, but the principle of trying to locate a known item is the same and this task seems fairly well accounted for by the classifications described above. However in the case of Broder [2], consideration is not given to the reason why the user wishes to locate a particular web site. Presumably visiting the site is not an end in itself, but part of the strategy for performing another task such as finding a phone number or arranging car rental.

The focus on classifying search behaviours means none of the schemes discussed so far can account for the task Matt carries out in the *arranging a trip* scenario. Whilst the *resource-interact* goal of Rose and Levinson [3] and the *transactional* queries of Broder [2] suggest an intention to carry out further interaction beyond the search (perhaps indicating a greater overall goal), the search itself is still seen as the user's primary task. No mention is given of arranging something as an overarching reason for being online, or even carrying out a search. Whilst no queries such as "arrange trip to conference" (the task Matt is performing) are reported, this likely reflects that users are aware of the limitations of search engines and therefore do not enter such queries, rather than a lack of desire to perform such tasks.

3.2 Distinguishing Between Needs and Strategies

Drawing on work in domains such as organisation science Choo, Detlor and Turnbull [5] highlight a distinction between a user's *information needs* and the *information seeking strategies* they employ to meet these needs. A similar distinction could also be made between the task a user intends to carry out online, and the strategies they use to complete this task.

Morrison, Pirolli and Card [6] describe a taxonomy of web activites with three variables: the *purpose* of a search, the *method* used, and the *content* of the information being searched for. Whilst these variables appear neatly defined, the classification of some activities suggests the variables may not be mutually exclusive in the form proposed by the authors. For example, some methods are seen to be triggered by a particular goal (*find*, *collect*) whereas others are not (*explore*, *monitor*). In this case it may be that explore and monitor actually represent goals in their own right, and should be classed under *purpose*.

Sellen, Murphy and Shaw [7] describe a classification that identifies six activities carried out on the web (*finding*, *information gathering*, *browsing*, *transacting*, *communicating*, *housekeeping*), based on a study of web use by twenty-four knowledge workers. This classification is not limited to describing variations of searching or browsing, and does attempt to capture the user's needs or goals in using the web. However, by focusing purely on web-based tasks (excluding communication by email, for example), the classification does make assumptions about the strategies being used in performing tasks online.

3.3 Summary

The literature outlined above demonstrates that there are many ways to conceptualise the activities people perform on the web. But to what extent do these classifications represent a valid account of users' goals when online? In general, the classifications address just a small selection of the tasks users may wish to perform online, they characterise component parts of much larger tasks which are not identified or accounted for, or draw distinctions between tasks where these may not actually exist. By taking a search-centric view of web usage some classifications also make assumptions about the strategies a user might employ. Even some schemes that attempt to distinguish needs from strategies remain driven by the principle of an *information* need and *information* seeking strategy, rather than a *task* need and a strategy for performing it.

These factors suggest that a fuller understanding of the range and nature of tasks performed online is necessary. In contrast to current classifications, any broader conceptualisation must adequately account for the scenarios given at the start of this paper, and must not assume the use of specific technology such as search engines or web browsers. In fact, rather than focusing solely on the web as the medium, the only assumption made should be of the user performing tasks using an internet connected device. Distinguishing the web from the rest of the internet in the case of task performance would be to confuse the task need with the strategy employed.

4 A User-Oriented Classification of Online Tasks

Drawing on the schemes described above and the discussion of their limitations, the following classification is proposed as a model of tasks users perform online.

Table 1. a user-oriented classification of online tasks

Task	Definition	Example
Locating	Looking for an object or chunk of information which is known or expected to exist; it may or may not have been seen before by the user.	Locating an article from a journal, an image for a school project, or information about a book a friend recommended.

Exploring	Gathering information about a specific concept or entity to gain understanding or background knowledge of that concept or entity.	Exploring a philosophical theory to understand its central tenets; getting background information about an organisation before a job interview.
Monitoring	Checking known sources that are expected to change, with the express intention of detecting the occurrence and nature of changes.	Monitoring news web sites during an election; checking email accounts for new messages; watching discussion fora for new ideas or information.
Grazing	Moving speculatively between sources with no specific goal in mind, but an expectation that items of interest may be encountered.	Following links that spark your interest on someone's web log, just to see what you find.
Sharing	Making an object or chunk of information available to others.	Sharing holiday photos through an online photo album; uploading a journal article to your personal web site.
Notifying	Informing others of an event in time or a change of state.	Emailing a group of friends to tell them you will be going to a concert at the weekend.
Asserting	Making statements of fact or opinion.	Writing on your web site that you like a certain film or artist, or that you own a certain book.
Discussing	Exchanging knowledge and opinions with others, on a specific topic.	Posting a comment on a discussion forum stating that you disagree with a previous post, explaining why, and then receiving responses from others.
Evaluating	Determining whether a particular piece of information is true, or assessing a number of alternative options.	Choosing which film to see at the weekend, based on what's showing, where, and at what time.
Arranging	Coordinating with third parties to ensure that something will take place or will be possible at a	Arranging travel and accommodation for an international conference.

	certain time.	
Transacting	Transferring money or credit between two locations; may or may not have some consequence in the offline world.	Paying a bill, or transferring money between accounts.

Relating this classification to the work of others, the *informational* goal of Rose and Levinson [3] maps clearly to the task of *exploring* described in Table 1, whilst the *resource* goal relates closely to the *locating* task introduced above. However, the *navigational* goal of Rose and Levinson [3] has no equivalent here as it is concerned merely with getting to a specific web site the user has in mind; it doesn't address the task the user intends to perform when they reach the site in question. The same criticism applies to the taxonomy of web searches developed by Broder [2], where in both the *navigational* and *transactional* types the user is attempting to reach web sites where they can perform their task. Considering the taxonomy of Morrison et al [6] raises an issue mentioned previously, that *explore* and *monitor* as they characterise it may actually represent tasks not methods. If this is the case then they correspond well to the tasks of *exploring* and *monitoring* introduced here in Table 1.

Several of the activities identified by Sellen et al [7] have direct equivalents in this classification. For example, their activity of *finding* maps directly to that of the *locating* task presented here, whilst both classifications define *transacting* in similar terms. The *information gathering* activity captures aspects of both the *exploring* and *evaluating* tasks introduced in this paper. Similarly their concept of *browsing* encompasses elements of *monitoring* and *grazing*, without distinguishing the two as this classification does. Whilst the similarities between tasks such as *locating* and *finding* provide a degree of validation for this classification, these examples also highlight the greater granularity of the tasks introduced in this paper.

The classification presented here addresses a wider range of user tasks than those described in Section 3 above. One reason for this greater coverage is that it explicitly includes tasks such as *notifying* and *sharing* that assume an audience or recipient other than the user. Secondly, this classification doesn't make assumptions about the technology being used in performing the task, only that the user is online by way of some form of internet connected device. For example, *notifying* might take place via email, and *discussing* could take the form of an instant messaging conversation. This serves to not limit the classification to a specific domain such as searching using a conventional web search engine, or a specific internet medium such as the web.

4.1 Linked Tasks

During any one online session, a user may perform a number of tasks that, whilst distinct, are in some way related; these could be thought of as linked tasks. For example, you may have heard that a concert is on in the city where you live. You would like to go to the concert, and so use a listings web site to find out that it starts at 8pm. Thinking that your friends might like to go as well, you then email them to let them know about the concert, mentioning the start time. In this case the first task is clearly an example of *locating*, as you set out to find a certain piece of information, whilst the second task constitutes *notifying*. Here the two tasks bear a thematic relationship but remain tasks in their own right, each addressing a particular goal. Similarly, *monitoring* a news web site may reveal a story of interest that results in the user *grazing* related sites with the expectation of finding other relevant items. Shopping online can be seen as a further example of linked tasks. The act of paying for goods or services can be classified as *transacting*, and this may be preceded by *locating* a specific item to purchase or *evaluating* a number of different options.

4.2 The Role of User Contexts

As the scenarios introduced earlier demonstrate, users rarely perform tasks in isolation. Taking the *arranging a trip* scenario as an example; without the knowledge gained from those around him Matt would likely have booked the more expensive flight to Ljubljana. He may also have begun a long and detailed search for alternative hotels within reach of the conference venue when he found that all official hotels were full. Similarly Alice's knowledge is crucial in helping Juan *locate* a book in the first scenario, both in recommending the book initially based on her own previous experience of reading it, and in clarifying the title.

In fact, a number of aspects of a user's context can be identified that may have significant roles to play in shaping the nature of the task and the way in which it's performed. These might include factors such as a user's *social networks*, their *previous experiences*, *preferences* they hold, their *current location*, services or third parties they *trust*, or the *resources* they have available for performing the task.

Crucially these context factors are likely to manifest themselves differently depending on the task being performed. For example, in tasks such as *notifying* or *sharing*, members of a user's *social network* may be seen as the audience for the task or the beneficiaries of its outcome, rather than sources of assistance as in the scenarios above; *discussing* on the other hand might involve contribution from all individuals, presumably for mutual benefit. Taking the factor of *trust* as an example, the extent to which a user *trusts* a third party web site may be of great significance if they are carrying out a *transacting* task such as paying for goods or transferring money. However, in contrast, if they are *exploring* a controversial topic and simply want to survey a broad range of opinions it may not matter whether they trust the sources they find or not.

5 Tool Support for Online Tasks

5.1 Conventional Internet Tools

If the classification presented in Table 1 represents the tasks people perform online, how are these tasks supported by current tools available on the web, and the internet as a whole? Some existing tools address the needs of these tasks fairly well. For example, software that reads news feeds from multiple web sites and aggregates the results on a user's desktop are a successful and widely used means of *monitoring* many sources at once. Unfortunately a similar level of uptake has not been seen with tools that monitor multiple email accounts, perhaps due to a lack of standardised ways of accessing web-based email accounts, and users often have to perform this task manually.

In many circumstances traditional search engines are an effective means of locating objects or information, although the *locating a book* scenario illustrates the type of situation where this is not the case. Furthermore, searches are largely limited to textual content due to the complexity of indexing other media such as images or music.

A number of question answering engines such as Ask Jeeves[1] are available that may be able to help *evaluate* if a certain piece of information is true, although the user may not be sure whether to trust the source of the answer. Furthermore, many comparison web sites exist that are able to evaluate the cheapest place to buy a product, or the fastest route between two points, but they are only able to use information explicitly represented in their databases, rather than reasoning about alternatives that may meet the user's criteria. This is highlighted in the second scenario, where the travel web site Matt uses is only able to provide information about routes he specifies, rather than reason about alternative ways of reaching the same destination.

As these examples and the problem analysis given above demonstrate, there is a need for tools and technologies that better support the user in performing tasks online.

5.2 Applications of Semantic Web Technology

A number of tools are discussed below that go some way to addressing these shortcomings, and move towards greater support of the kinds of tasks identified in Table 1. Whilst their features may be described in different terms by their authors, these tools can all be seen to support aspects of the *exploring* task introduced above. To varying degrees they all draw on Semantic Web technologies or principles to support their additional functionality. The Semantic Web vision [8] [9] proposes an extension of the current web that takes it from a collection of interlinked documents for human consumption to a space where information is sufficiently structured, and the rules that define this structure sufficiently explicit, as to allow machines to understand and reason with it. Fundamental to this vision are the basic building blocks of knowledge represented using the Resource Description Framework (RDF), and rules for logical inference in the form of ontologies.

Guha et al [1] describe a system known as TAP, which seeks to support what they term *research searches*. By using Semantic Web data describing concepts and their relationships to others entities, the system is able to provide search results tailored to the concept being searched for. This principle is illustrated with the example of a search for the musician Yo-Yo Ma that returns "his current concert schedule, his music albums, his image, etc." (pp. 702). If however the search term denoted a researcher rather than a musician, the system might return information about the

[1] http://www.ask.com/

person's publications or their research interests. In terms of the *exploring* task, this approach may help the user by providing links to background information not easily assembled using conventional web search engines.

Also supporting users in exploring concepts or entities is the browsing tool Magpie [10]. In contrast to TAP this tool assumes that the user has been able to reach a document that contains some concepts or entities of interest. A user-selected ontological layer over the original document then allows the invocation of semantic services related to those concepts. This serves the purpose of providing related information that may not be explicitly mentioned on the page being viewed.

Another tool that builds on the browser metaphor and applies it to the Semantic Web is Haystack [11]. Here the user is able to browse arbitrary collections of RDF metadata through a point and click interface, with links being made between semantically related items. Crucially this tool is able to gather information on a particular topic from multiple sources and assemble it in one place, in contrast to conventional models of web browsing where the user may have to visit several different pages or sites to gather related pieces of information.

Whilst implemented differently (on the web rather than on the desktop), the application CS AKTive Space [12] provides a similar ability to explore relations between concepts or entities, although in this case the system is limited to the domain of computer science research in the UK.

One feature these tools have in common is the ability to present the user with new pieces of information, or make new connections between concepts or entities that might not otherwise have been apparent; this ability is a key feature of the Semantic Web. To this end they make a significant step towards supporting users in the task of *exploring* concepts or entities to gain additional knowledge or understanding.

However, many of the other tasks identified earlier are not so well supported. For example, resolving the issues highlighted by the scenarios presented in Section 1 requires tools adapted to *locating* and *arranging* that go beyond the traditional search engines and travel web sites currently available. Semantic Web technologies such as those outlined in [9], may provide the technological basis for building such tools, by enabling the creation of large, distributed, and dynamic knowledge bases, and the means to reason across them.

In the *locating a book* scenario, this might enable Juan to specify that the book he is looking for is called something like "The Sergeant's Guitar". A system that could make use of background knowledge about the semantic links between 'sergeant' and 'captain', and between 'guitar and mandolin', might be able to identify "Captain Corelli's Mandolin" as one possible match within the online bookshop, rather than returning no results. Similarly when *arranging a trip*, a Semantic Web application could take Matt's destination as an input, reason about ways of reaching that destination and propose a number of travel itineraries, leaving Matt to choose the one that best meets his needs. In both these cases, tools that draw on aspects of Juan and Matt's contexts, particularly knowledge held by those around them, would be beneficial in completing the tasks.

Of the tools discussed above, perhaps the only one to take any account of user context is Magpie. The user's selection of an ontological layer could be seen to reflect some aspect of their context, in that subscription to a shared conceptualisation likely reflects their perspective on a domain to some extent. However, this representation of context is implicit and does not approach the richness of the factors proposed in this paper.

6 Conclusions: From a Semantic Web to a Task-Focused Context-Aware Internet

In conclusion, unless tools are developed that are adapted to the task the user wishes to perform, and that take into account the contexts in which the user exists, the kind of obstacles highlighted in the scenarios above are likely to remain. Task-focused and context-aware tools could provide a more effective means for users to perform tasks online than current web tools, and Semantic Web technologies may provide the platform for developing them, if the following challenges can be met. Firstly, can a user's contextual data be captured and made available on the Semantic Web in a meaningful and reusable form, and how might this be achieved? Secondly, can tools be developed that are able to reason about the contextual information needed to assist in the performance of a particular task?

Such tools should be extended to cover the full spectrum of tasks users perform online, and also operate across a wider range of internet platforms such as email and instant messaging. Not to do so would draw a distinction between the web and the wider internet based on technical grounds such as the particular protocol being used. Distinctions of this sort hold little meaning for the average user, who is concerned primarily with performing a task irrespective of how tools are implemented. To this end it may be more appropriate to envision a task-focused context-aware internet, where all online activities can benefit from the use of semantic technologies.

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. Guha, R., McCool, R., Miller, E.: Semantic Search. In: Proc. Twelfth International Conference on World Wide Web (WWW2003) (2003) 700-709
2. Broder, A.: A Taxonomy of Web Search. ACM SIGIR Forum 36 (2002) 3-10
3. Rose, D. E., Levinson, D.: Understanding User Goals in Web Search. In: Proc. 13th International Conference on World Wide Web (WWW2004) (2004) 13-19
4. Olston, C., Chi, E. H.: ScentTrails: Integrating Browsing and Searching on the Web. ACM Transactions on Computer-Human Interaction (TOCHI) 10 (2003) 177-197

5. Choo, C. W., Detlor, B., Turnbull, D.: Information Seeking on the Web: An Integrated Model of Browsing and Searching. In: Proc. 62nd Annual Meeting of the American Society for Information Science (1999) 3-16
6. Morrison, J. B., Pirolli, P., Card, S. K.: A Taxonomic Analysis of What World Wide Web Activities Significantly Impact People's Decisions and Actions. In: Proc. Conference on Human Factors in Computing Systems (CHI'01) (2001) 163-164
7. Sellen, A. J., Murphy, R., Shaw, K. L.: How Knowledge Workers Use the Web. In: Proc. SIGCHI Conference on Human Factors in Computing Systems (CHI2002) (2002) 227-234
8. Berners-Lee, T.: Semantic Web Road Map. World Wide Web Consortium (W3C) (1998)
9. Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web. Scientific American 284 (2001) 34-43
10. Dzbor, M., Domingue, J., Motta, E.: Magpie - Towards a Semantic Web Browser. In: Proc. 2nd International Semantic Web Conference (2003)
11. Quan, D., Karger, D. R.: How to Make a Semantic Web Browser. In: Proc. 13th International Conference on World Wide Web (WWW2004) (2004) 255-265
12. Schraefel, M. C., Shadbolt, N. R., Gibbins, N., Harris, S., Glaser, H.: CS AKTive Space: Representing Computer Science in the Semantic Web. In: Proc. 13th International Conference on World Wide Web (2004) 384-392

Web Usage Driven Adaptation of the Semantic Web

Alexander Mikroyannidis, Babis Theodoulidis

School of Informatics, University of Manchester,
Sackville Street, Manchester M60 1QD, United Kingdom
{A.Mikroyannidis, babis.theodoulidis}@manchester.ac.uk

Abstract. The notion of the Semantic Web has emerged as a solution to the problem of organizing the immense information provided by the World Wide Web. However, this information has to be constantly updated and reorganized in order to better serve the changing needs of the web users. A static Semantic Web can therefore be of little use in the environment of the ever-transforming World Wide Web. In the context of the present work, we propose a framework for web usage driven adaptation of the Semantic Web. Based on the usage of the web, we perform evolution of its topology and ontology. This procedure aims to facilitate the way the user interacts with the web, resulting in an increase in the usability of the web through the refinement of its physical and semantic structure.

1 Introduction

Being a large and dynamic information source, both structurally complex and ever growing, the World Wide Web poses great difficulties to its full exploitation. The Semantic Web addresses this problem by "giving information a well-defined meaning, better enabling computers and people to work in cooperation" [1]. This is implemented by expressing the web data in forms that are machine-understandable and machine-processable, in order to be more efficiently maintained by software agents.

Nevertheless, a significant issue, which is usually overlooked, is the usability of the Semantic Web. The way with which a user browses the web is heavily dependent on his needs, knowledge and interests. These needs and interests have to be addressed by the Semantic Web, in order for enhancement to be achieved in the user's interaction with the web. Moreover, since these preferences are altered through time, the Semantic Web must have the ability to satisfy them through a constant adaptation process.

In [8, 9] we introduced a framework for self-adaptive web sites. The present paper extends this work by addressing the adaptation of the Semantic Web, based on web usage data. A framework that employs web usage mining as well as text mining methodologies is presented. The proposed framework adapts the web in order to assist the users in their browsing tasks. Both the physical and semantic structure of the web are targeted. The web site ontology is semi-automatically built and evolves through the adaptation procedure.

The remainder of this paper is organized as follows: Section 2 describes the approaches that have been followed by researchers in the area of web adaptation. Section 3 introduces the theoretical principles upon which our framework was built. Section 4 presents an architecture that implements the framework. Section 5 discusses the results of the proposed approach on the usability of the web. Finally, the paper is concluded and some plans for future work are provided.

2 Related Work

Providing users with assistance in their web navigation can help keep them in a web site, or even attract more visitors. This has always been a popular subject, especially in the e-commerce domain. Several systems have been developed towards this direction. WebWatcher [7] suggests links that may interest a user, based on the online behaviour of other users. Each user is asked, upon entering the site, what kind of information he is seeking. Before he departs, he is asked whether he has found what he was looking for. His navigation paths are used to deduce suggestions for future visitors that seek the same content. These suggestions are visualized by highlighting existing hyperlinks.

The Avanti project [6] tries to predict the user's final objective as well as his next step. A model for the user is built, based partly on information the user provides about him. His interests are also extracted from his navigation paths. Visitors are provided with direct links to pages that are probably the ones they are looking for. In addition, hyperlinks that lead to pages of potential interest to each visitor are highlighted.

A drawback of both the WebWatcher and the Avanti system is that they require the active participation of the users in the adaptation process, by asking them to provide information about themselves. On the other hand, the Footprints [13] system relies entirely on the navigation paths of the users. The system does not perform user identification. All navigation paths are recorded and the most frequent ones are presented to the visitor, in the form of maps or trails. Html pages also display next to each link the percentage of people who have followed it. Nevertheless, as in the WebWatcher and the Avanti systems, no adaptation of the site's structure is performed.

Perkowitz et al [12] have presented a conceptual framework for adaptive web sites. They have focused on the semi-automatic creation of index pages, based on discovering clusters of pages. They assume that if a large number of visitors frequently visit a set of pages, this provides strong evidence that these pages are related. They have developed two cluster mining algorithms, PageGather and IndexFinder. The first one relies on a statistical approach to discover candidate link sets, while the second is a conceptual cluster mining algorithm, as it finds link sets that are conceptually coherent. They have also performed experiments on three web sites by placing the automatically generated pages online and observing the user response.

However, the majority of the existing approaches in web adaptation lack in a crucial factor: they do not address the semantic aspect of the web. The ontological perspective is overlooked and the researchers' attention is drawn mainly by the site topology. Even though the improvement of the site topology is unquestionably significant, we should not disregard the fact that users browse a site mainly for its content. Consequently, the content classification structure should also be adaptive through the evolution of the site ontology. The innovative concept of the Semantic Web is a most suitable region for applying such adaptation methodologies, targeting to the direct benefit of the end users.

3 Framework

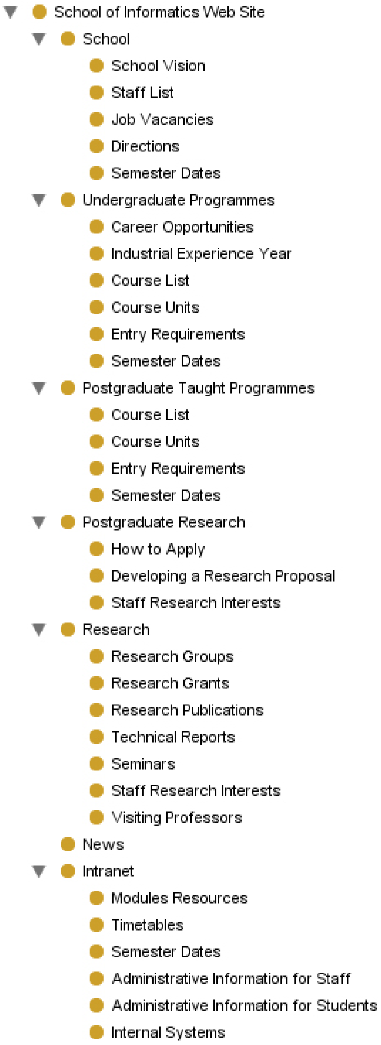
The proposed framework defines the adaptation process as absolutely *transparent* to the user, requiring no active participation from him. In addition, the adaptations of our framework perform *web transformation*, instead of focusing on personalization tasks. Mobasher et al [10] define personalization as "any action that tailors the web experience to a particular user, or a set of users". On the other hand, according to Perkowitz et al [11], transformation is "improving the site's structure based on interactions with *all* visitors". The advantage of this approach is that it does not require user identification, which cannot be safely performed from usage data, unless the user contributes in an explicit or implicit way [5]. Nevertheless, most users are reluctant to give away personal information. Moreover, through transformation, transparency is achieved, as the adaptation procedure relies completely on the data gathered in the access logs.

Coenen et al [2] distinguish between *tactical* and *strategic* adaptations in their framework for self-adaptive web sites. They call tactical the adaptations that can be performed in real time, without the webmaster's approval, since they do not affect the overall site structure. On the other hand, strategic adaptations are the ones that "go against or conflict with the original beliefs of the site, and consequently have an important influence on the original site-structure". Coenen et al suggest that such modifications should be performed offline, with the approval of the webmaster.

The role of the webmaster is considered fundamental in the present framework. Human designers often dedicate a large effort in developing a site. By no means, the adaptation process should undo their work. The framework puts the webmaster in charge of the adaptation procedure, by requiring from him to approve the adaptations. In addition, we propose an adaptation engine that will learn from the webmaster's responses. Instead of predefining which modifications are strategic and which tactical, the adaptation system should gradually learn to classify the adaptations, by studying the webmaster's approvals and rejections of proposed adaptations. Adaptations that are classified by the system as tactical should be applied automatically, without the webmaster's interference. In this way, the site will adapt not only to the end user's preferences, but to the webmaster's as well.

In order to improve the reorganization of the information provided by a web site, we have exploited the semantic aspect of the web. Apart from the topology of the web site, the framework also addresses the evolution of the site ontology. A web site ontology is strongly related to the topology of the site. It is comprised of the thematic categories covered by the site's pages. These categories are the concepts of the ontology. Each web page, depending on its content, is an instance of one or more concepts of the ontology. The concepts can be organized in a hierarchy, representing an "is a" relationship. This means that a class is a subclass of another class if every instance of the second class is also an instance of the first.

Figure 1 illustrates the web site ontology of the University of Manchester School of Informatics (http://www.co.umist.ac.uk). The ontology has been built considering the organization of the thematic categories as this is defined in the current topology of the site. The hierarchy's top level contains seven classes: School, Undergraduate Programmes, Postgraduate Taught Programmes, Postgraduate Research, Research, News and Intranet. These are the main thematic categories of the site. These categories are then expanded to more specific concepts, which are represented by subclasses. All the concepts are instantiated in the web pages of the site.



- School of Informatics Web Site
  - School
    - School Vision
    - Staff List
    - Job Vacancies
    - Directions
    - Semester Dates
  - Undergraduate Programmes
    - Career Opportunities
    - Industrial Experience Year
    - Course List
    - Course Units
    - Entry Requirements
    - Semester Dates
  - Postgraduate Taught Programmes
    - Course List
    - Course Units
    - Entry Requirements
    - Semester Dates
  - Postgraduate Research
    - How to Apply
    - Developing a Research Proposal
    - Staff Research Interests
  - Research
    - Research Groups
    - Research Grants
    - Research Publications
    - Technical Reports
    - Seminars
    - Staff Research Interests
    - Visiting Professors
  - News
  - Intranet
    - Modules Resources
    - Timetables
    - Semester Dates
    - Administrative Information for Staff
    - Administrative Information for Students
    - Internal Systems

Fig. 1. The School of Informatics web site ontology

It must be stressed that the web site ontology is quite different from the domain ontology [4]. The latter describes relationships between the concepts of a domain, whereas the first is based on the organization of the information found in a web site. The ontology of a domain is usually more complex than the ontology of a web site related to the same domain. However, the maintenance of a web site ontology requires considerable effort and has to be performed on a regular basis, since the content of a web site is constantly updated.

The adaptation of the ontology can include the discovery of new associations between its concepts. Moreover, a concept can be found to have more than one superclasses or a web page to be an instance of more than one concepts. Finally, a web page may possibly need to be categorized under a different concept than its current one.

4 Architecture

Figure 2 presents an architecture implementing the theoretical principles of the proposed framework. As it can be seen, the inputs of the adaptation process consist of the server's access logs, the site topology and ontology. The whole procedure aims at the evolution of the topology and ontology of the web site.

The adaptation starts with a preprocessing stage, during which the data stored in the raw access logs are cleaned and visiting sessions are identified. The sessions are then mined with the use of Frequent Itemset Mining algorithms in order to produce *pagesets*. We call pagesets the sets of pages that are frequently accessed together throughout the same session.

The extracted pagesets are classified in relation to certain features of their pages. More specifically, two classification criteria have been used: linkage state and content. The first criterion refers to the connection that the pages of each pageset have, according to the site structure. The key factor is whether the pages contained in a pageset, are directly linked to each other or not. Pagesets of unlinked pages might suggest the insertion of shortcut links between these pages, in order to achieve shorter navigation paths. From the pagesets of linked pages, changes in the appearance of existing links can be extracted. For example, if an index page and some of its links comprise one or more pagesets, then by highlighting these links in the index page, first time visitors would be able to navigate the site easier.

The second classification criterion refers to the content of the pages contained in each pageset. The pages of the pagesets are classified in order to discover new associations between the concepts of the site ontology. More particularly, if a pageset includes pages belonging to concepts that were not previously linked, the ontology should then be modified to reflect the relevance these concepts have, according to the preferences of the users.

Based on the linkage state and content classification, a report containing proposals for the improvement of the site is generated. This report contains proposals for the insertion of shortcut links from source pages to target pages that are frequently accessed together but are currently not linked. It also contains proposals for the change of the appearance of popular hyperlinks. Furthermore, the report contains proposals for the evolution of the site ontology.

After the proposed modifications have been revised by the webmaster, they can be applied to the web site. The site topology is then refined through the insertion of new shortcut links, as well as changes in the appearance of the existing ones. The ontology is also refined in a number of ways.

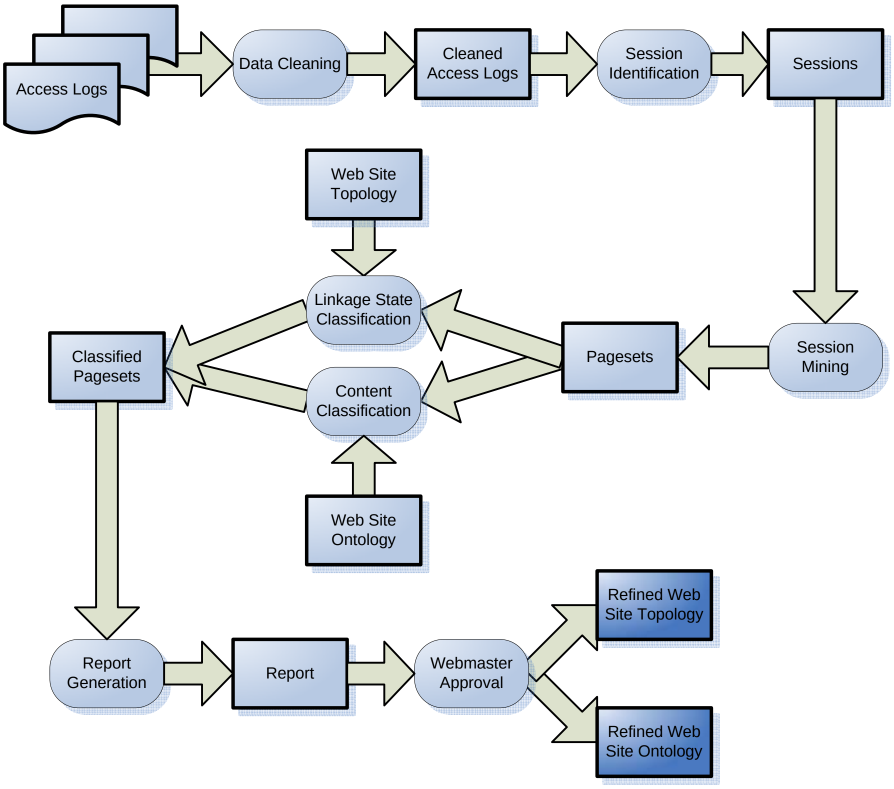
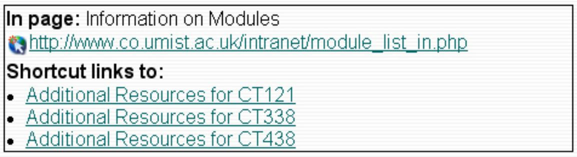


Fig. 2. Web site adaptation architecture

5 Results and Discussion

We have applied our methodology on the University of Manchester School of Informatics web site. The topology of the web site was refined through the insertion of new shortcut links between pages that were not previously linked together, as well as through the highlighting of popular existing links. In addition, the web site ontology was modified in several ways, based on the outcomes retrieved from the classified pagesets.

More specifically, the adaptation system produced two sets of reports: shortcut links reports and highlighted links reports. Figure 3 shows an extract from a report containing proposals for insertion of shortcut links. From a source page, shortcut links to target pages are suggested. The target pages have been found to be frequently visited after the source page. However, the source page is not linked to the target pages, thus forcing the users to follow alternative paths in order to reach them. Shorter navigation paths can be therefore achieved if the source page is linked to the target pages. This is the purpose of this type of report. For instance, some pages that contain additional resources on certain courses are frequently accessed by users after accessing the "Information on Modules" page, which contains a list of all the department's modules. Consequently, as it can be seen in Figure 3, shortcut links are proposed that lead to these pages.



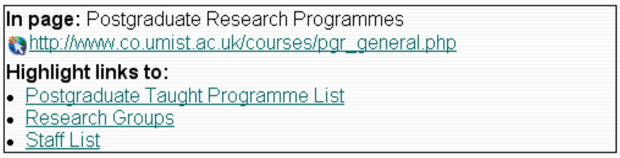
In page: Information on Modules
http://www.co.umist.ac.uk/intranet/module_list_in.php

Shortcut links to:

- Additional Resources for CT121
- Additional Resources for CT338
- Additional Resources for CT438

Fig. 3. Extract from a shortcut links report

The highlighted links report is comprised of suggestions for emphasizing popular hyperlinks. This can be quite useful, especially in pages that contain large amounts of hyperlinks, such as index pages. In such cases, the user can gain valuable time if prompted with the most popular choices. Figure 4 shows an example of a highlighted links report. Certain links, based on their popularity have been proposed to be suggested to the user who visits the "Postgraduate Research Programmes" page.



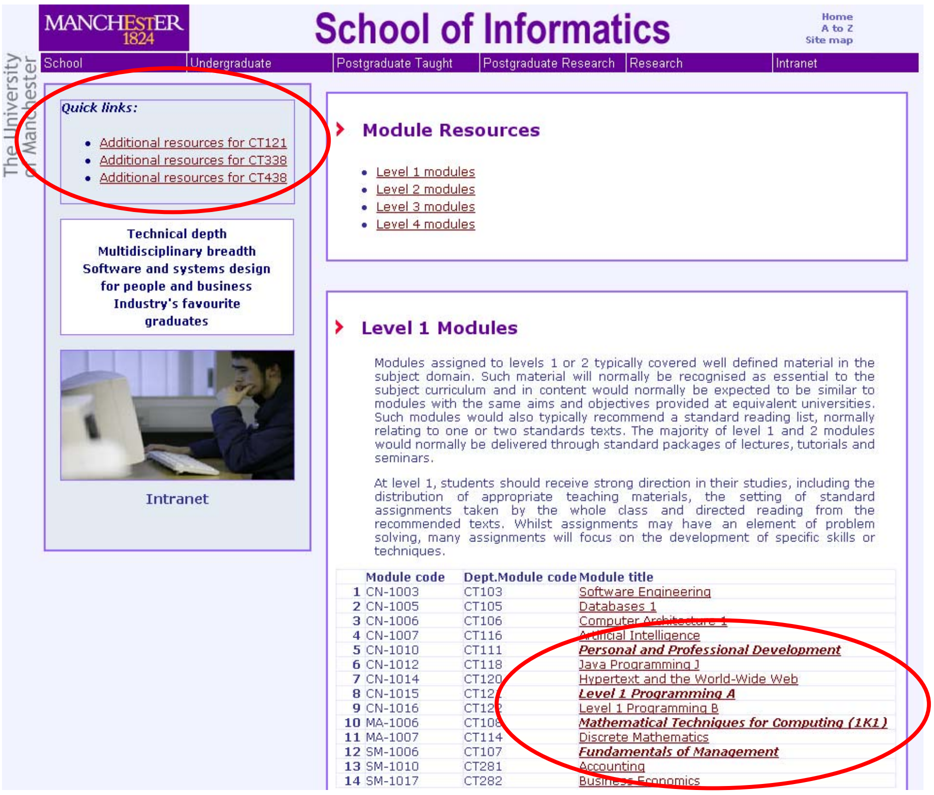
In page: Postgraduate Research Programmes
http://www.co.umist.ac.uk/courses/pgr_general.php

Highlight links to:

- Postgraduate Taught Programme List
- Research Groups
- Staff List

Fig. 4. Extract from a highlighted links report

Figure 5 shows an example of a modified web page, according to our system's suggestions. It is the "Information on Modules" page, which is very popular in the users' preferences. The page has been modified to facilitate the navigation of the users during the first semester. Shortcut links to popular courses of the first semester have been inserted in the left side of the page, under the title "Quick links". Moreover, popular links that already existed, such as the hyperlink leading to the page of the "Personal and Professional Development" course, have been highlighted.



Quick links:

- Additional resources for CT121
- Additional resources for CT338
- Additional resources for CT438

Technical depth
Multidisciplinary breadth
Software and systems design
for people and business
Industry's favourite
graduates

Module Resources

- Level 1 modules
- Level 2 modules
- Level 3 modules
- Level 4 modules

Level 1 Modules

Modules assigned to levels 1 or 2 typically covered well defined material in the subject domain. Such material will normally be recognised as essential to the subject curriculum and in content would normally be expected to be similar to modules with the same aims and objectives provided at equivalent universities. Such modules would also typically recommend a standard reading list, normally relating to one or two standards texts. The majority of level 1 and 2 modules would normally be delivered through standard packages of lectures, tutorials and seminars.

At level 1, students should receive strong direction in their studies, including the distribution of appropriate teaching materials, the setting of standard assignments taken by the whole class and directed reading from the recommended texts. Whilst assignments may have an element of problem solving, many assignments will focus on the development of specific skills or techniques.

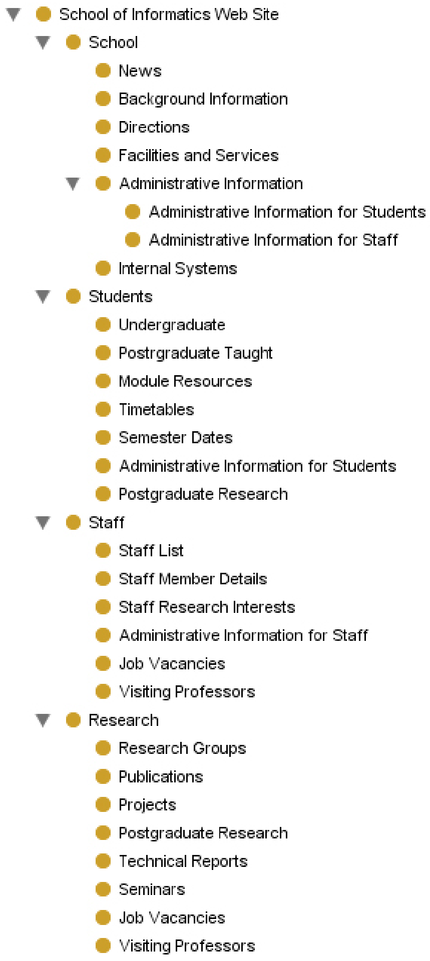
	Module code	Dept.Module code	Module title
1	CN-1003	CT103	Software Engineering
2	CN-1005	CT105	Databases 1
3	CN-1006	CT106	Computer Architecture 1
4	CN-1007	CT116	Artificial Intelligence
5	CN-1010	CT111	***Personal and Professional Development***
6	CN-1012	CT118	Java Programming J
7	CN-1014	CT120	Hypertext and the World-Wide Web
8	CN-1015	CT121	***Level 1 Programming A***
9	CN-1016	CT122	Level 1 Programming B
10	MA-1006	CT108	***Mathematical Techniques for Computing (1K1)***
11	MA-1007	CT114	Discrete Mathematics
12	SM-1006	CT107	***Fundamentals of Management***
13	SM-1010	CT281	Accounting
14	SM-1017	CT282	Business Economics

Fig. 5. Example of modified web page

The web site ontology was modified in several ways, based on the outcomes retrieved from the classified pagesets. The resulting ontology, after the application of adaptations suggested by our system, is shown in Figure 6. Based on these adaptations, the content organization of the site was altered to better satisfy the needs of its visitors. For the content classification of the web pages belonging to the pagesets, a classifier implementing the Support Vector Machines categorization algorithm [3] was used.

First of all, new associations were discovered between concepts. These associations reflect the interests of the users, as documents belonging to these concepts are frequently accessed together. New associations were inserted between the following concepts:

- "Research" and "Students"
- "Research" and "Staff"
- "School" and "Students"
- "School" and "Staff"
- "Students" and "Staff"



- School of Informatics Web Site
  - School
    - News
    - Background Information
    - Directions
    - Facilities and Services
    - Administrative Information
      - Administrative Information for Students
      - Administrative Information for Staff
    - Internal Systems
  - Students
    - Undergraduate
    - Postrgraduate Taught
    - Module Resources
    - Timetables
    - Semester Dates
    - Administrative Information for Students
    - Postgraduate Research
  - Staff
    - Staff List
    - Staff Member Details
    - Staff Research Interests
    - Administrative Information for Staff
    - Job Vacancies
    - Visiting Professors
  - Research
    - Research Groups
    - Publications
    - Projects
    - Postgraduate Research
    - Technical Reports
    - Seminars
    - Job Vacancies
    - Visiting Professors

Fig. 6. Refined web site ontology of the School of Informatics

Reorganization of the concepts' hierarchy was also performed. Further improvements included the creation of new categories, the removal of existing categories, as well as changes to the levels of hierarchy that the concepts belong to. For instance, the "Staff" concept was previously a subconcept of the "School" concept, which resided in the highest level of the ontology. It should be noted that the "Staff" concept has as instances all the web pages that carry information about the staff members of the school. However, the high frequency with which this concept appeared in the pagesets implies the significance that it has in the interests of the users. It would be thus appropriate to transfer this concept to the top level of the ontology, as shown in Figure 6. Based on the performed classification, the undergraduate and postgraduate programmes were grouped under the more general concept "Students". The "School" concept was also extended to include more subconcepts.

The ontology of the site was extended to include multiple instances of concepts or multiple subconcepts. The categorization of the web pages that was carried out, suggested that several pages belong to more than one concept. Moreover, in some cases, web pages and the corresponding concepts were categorized under different concepts than they previously were in the existing ontology. The site ontology should be therefore updated in order to reflect this fact. For example, the "Job vacancies" web page, which corresponds to the "Job Vacancies" concept, was found to be an instance of both the "Staff" and "Research" concepts. The information contained in this page regards mainly research job posts and is also highly related to the "Staff" concept. This page was previously categorized only under the "School" concept. In the updated ontology (Figure 6), the "Job Vacancies" concept has been placed both under the "Staff" and "Research" concepts. The same modification has been applied to the concepts "Visiting Professors", "Administrative Information for Students", etc.

Finally, useful conclusions were deduced about the usage of the web site. Particularly, the thematic category that was the first in the preferences of the users was, as expected, the "Students" concept. This concept contains all pages that support the school's modules, both undergraduate and postgraduate. This is not surprising, since most of the traffic is generated by the students. Second in the users' interests comes the "Staff" concept. The "Research" concept is third, followed by the "School" category. These results can be used to enhance the performance of the server, for example by the use of additional servers that will host the popular resources, or to promote the problematic concepts by making them more easily accessible.

6 Conclusions and Future Work

The present work investigated a web usage driven approach on the adaptation of the Semantic Web. A framework was introduced that enables adaptation of the web topology and ontology to the needs and interests of web users. In addition, an architecture based on the principles of the framework was presented. The proposed adaptation process exploits the access data of the users, together with the semantic aspect of the web, in order to facilitate web browsing.

A real web site was used as a case study, in order to study the impact that the proposed framework can have on the usability of the web. The topology and ontology of the site were refined in several ways. Apart from changes in specific web pages, enhancements of the whole formation of the site were derived. Furthermore, useful knowledge was acquired, regarding the overall usage of the site. The sections that mostly interest the users were identified, leading to further improvements in their usability. Moreover, the regions of the site that need more promotion were revealed.

The current framework regards each web site as a separate unit. In future work, we plan to extend this approach, by performing simultaneous adaptation of multiple web sites. This task requires consideration of the relationships between the topologies and ontologies of different web sites. This extension is necessary in order to view the World Wide Web as an integral whole, towards the development of the Adaptive Semantic Web.

References

[1] Berners-Lee, T., Hendler, J., and Lassila, O. *The semantic web*. Scientific American, 2001. **279**(5): p.34-43.

[2] Coenen, F., Swinnen, G., Vanhoof, K., and Wets, G. *A Framework for Self Adaptive Websites: Tactical versus Strategic Changes.* In *Proc. of WEBKDD'2000 Web Mining for E-Commerce - Challenges and Opportunities, Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2000. Boston.

[3] Cortes, C. and Vapnik, V. *Support Vector Networks*. Machine Learning, 1995. **20**(3): p.273-297.

[4] Daconta, M., Obrst, L., and Smith, K. *The Semantic Web: A Guide to the Future of XML, Web Services, and Knowledge Management.* Wiley, 2003.

[5] Eirinaki, M. and Vazirgiannis, M. *Web Mining for Web Personalization*. ACM Transactions on Internet Technology, 2003. **3**(1): p.1-27.

[6] Fink, J., Kobsa, A., and Nill, A. *User-Oriented Adaptivity and Adaptability in the AVANTI project.* In *Proc. of Designing for the Web: Empirical*. 1996.

[7] Joachims, T., Freitag, D., and Mitchell, T. *WebWatcher: A Tour Guide for the World Wide Web.* In *Proc. of International Joint Conference on Artificial Intelligence*. 1997. Nagoya, Japan, p.770-775.

[8] Mikroyannidis, A. *Development of a framework for self-adaptive web sites.* School of Informatics, University of Manchester, MPhil Thesis, 2004.

[9] Mikroyannidis, A. and Theodoulidis, B. *A Theoretical Framework and an Implementation Architecture for Self Adaptive Web Sites.* In *Proc. of IEEE/WIC/ACM International Conference on Web Intelligence (WI'04).* 2004. Beijing, China, p.558-561.

[10] Mobasher, B., Cooley, R., and Srivastava, J. *Automatic Personalization Based on Web Usage Mining*. Communications of the ACM, 2000. **43**(8): p.142-151.

[11] Perkowitz, M. and Etzioni, O. *Adaptive Web sites*. Communications of the ACM, 2000. **43**(8): p.152-158.

[12] Perkowitz, M. and Etzioni, O. *Towards adaptive Web sites: Conceptual framework and case study*. Artificial Intelligence, 2000. **118**(1-2): p.245-275.

[13] Wexelblat, A. and Maes, P. *Footprints: History-Rich Tools for Information Foraging.* In *Proc. of Proceedings of Human Factors in Computing Systems (CHI).* 1999. Pittsburgh, Pennsylvania, United States, p.270-277.

Data quality issues in collaborative filtering

Miha Grčar, Dunja Mladenič, and Marko Grobelnik

J.Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia
Miha.Grcar@ijs.si,
WWW home page: http://kt.ijs.si/

Abstract. In this paper, we present our experience in applying collaborative filtering to real-life corporate data. The quality of collaborative filtering recommendations is highly dependent on the quality of the data used to identify users' preferences. To understand the influence that highly sparse server-side collected data has on the accuracy of collaborative filtering, we ran a series of experiments in which we used publicly available datasets and, on the other hand, a real-life corporate dataset that does not fit the profile of ideal data for collaborative filtering. We have performed a series of experiments on two standard data sets (EachMovie and Jester) and a real-life corporate data.

1 Introduction and motivation

The goal of collaborative filtering is to explore a vast collection of items in order to detect those which might be of interest to the active user. In contrast to content-based recommender systems which focus on finding contents that best match the user's query, collaborative filtering is based on the assumption that similar users have similar preferences. It explores the database of users' preferences and searches for users that are similar to the active user. The active user's preferences are then inferred from preferences of the similar users. One of the main advantages of pure collaborative filtering is that it ignores the form and the content of items and can therefore also be applied to non-textual items.

The accuracy of collaborative filtering recommendations is highly dependant on the quality of the users' preferences database. In this paper we would like to emphasize the differences between applying collaborative filtering to publicly available datasets and, on the other hand, to a dataset derived from real-life corporate Web logs. The latter does not fit the profile of ideal data for collaborative filtering.

The rest of this paper is arranged as follows. In Sections 2 and 3 we discuss collaborative filtering algorithms and data quality for collaborative filtering. Our evaluation platform and the three datasets used in our experiments are described in Sections 4 and 5. In Sections 6 and 7 the experimental setting and the evaluation results are presented. The paper concludes with the discussion and some ideas for future work (Section 8).

2 Collaborative filtering

There are basically two approaches to the implementation of a collaborative filtering algorithm. The first one is the so called “lazy learning” approach (also known as the memory-based approach) which skips the learning phase. Each time it is about to make a recommendation, it simply explores the database of user-item interactions. The model-based approach, on the other hand, first builds a model out of the user-item interaction database and then uses this model to make recommendations. “Making recommendations” is equivalent to predicting the user’s preferences for unobserved items.

The data in the user-item interaction database can be collected either explicitly (explicit ratings) or implicitly (implicit preferences). In the first case the user’s participation is required. The user is asked to explicitly submit his/her rating for the given item. In contrast to this, implicit preferences are inferred from the user’s actions in the context of an item (that is why the term “user-item interaction” is used instead of the word “rating” when referring to users’ preferences in this paper). Data can be collected implicitly either on the client side or on the server side. In the first case the user is bound to use modified client-side software that logs his/her actions. Since we do not want to enforce modified client-side software, this possibility is usually omitted. In the second case the logging is done by a server. In the context of the Web, implicit preferences can be determined from access logs that are automatically maintained by Web servers.

Collected data is first preprocessed and arranged into a user-item matrix. Rows represent users and columns represent items. Each matrix element is in general a set of actions that a specific user took in the context of a specific item. In most cases a matrix element is a single number representing either an explicit rating or a rating that was inferred from the user’s actions.

Since a user usually does not access every item in the repository, the vector (i.e. the matrix row), representing the user, is missing some/many values. To emphasize this, we use the terms “sparse vector” and “sparse matrix”.

The most intuitive and widely used algorithm for collaborative filtering is the so called k-Nearest Neighbors algorithm which is a memory-based approach. Technical details can be found, for example, in Grcar (2004). The algorithm is as follows:

1. Represent each user by a sparse vector of his/her ratings.
2. Define the similarity measure between two sparse vectors. In this paper, we consider two widely used measures: (i) the Pearson correlation coefficient which is used in statistics to measure the degree of correlation between two variables (Resnick et al. (1994)), and (ii) the Cosine similarity measure which is originally used in information retrieval to compare between two documents (introduced by Salton and McGill in 1983).
3. Find k users that have rated the item in question and are most similar to the active user (i.e. the user’s neighborhood).

4. Predict the active user's rating for the item in question by calculating the weighted average of the ratings given to that item by other users from the neighborhood.

3 Sparsity problem and data quality for collaborative filtering

The fact that we are dealing with a sparse matrix can result in the most concerning problem of collaborative filtering – the so called sparsity problem. In order to be able to compare two sparse vectors, similarity measures require some values to overlap. What is more, the lower the amount of overlapping values, the lower the relialibility of these measures. If we are dealing with high level of sparsity, we are unable to form reliable neighborhoods. Furthermore, in highly sparse data there might be many unrated (unseen) items and many inactive users. Those items/users, unfortunately, cannot participate in the collaborative filtering process.

Sparsity is not the only reason for the inaccuracy of recommendations provided by collaborative filtering. If we are dealing with implicit preferences, the ratings are usually inferred from the user-item interactions, as already mentioned earlier in the text. Mapping implicit preferences into explicit ratings is a non-trivial task and can result in false mappings. The latter is even more true for server-side collected data in the context of the Web since Web logs contain very limited information. To determine how much time a user was reading a document, we need to compute the difference in time-stamps of two consecutive requests from that user. This, however, does not tell us weather the user was actually reading the document or he/she, for example, went out to lunch, leaving the browser opened. What is more, the user may be accessing cached information (either from a local cache or from an intermediate proxy server cache) and there is no way to detect these events on the server side.

Also, if a user is not logged in and he/she does not accept cookies, we are unable to track him/her. In such case, the only available information that could potentially help us to track the user is his/her IP address. However, many users can share the same IP and, what is more, one user can have many IP addresses even in the same session. The only reliable tracking mechanisms are cookies and requiring users to log in in order to access relevant contents.

From this brief description of data problems we can conclude that for applying collaborative filtering, explicitly given data with low sparsity are preferred to implicitly collected data with high sparsity. The worst case scenario is having highly sparse data derived from Web logs. When so, why would we want to apply collaborative filtering to Web logs? The answer is that collecting data in such manner requires no effort from the users and also, the users are not obliged to use any kind of specialized Web browsing software. This "conflict of interests" is illustrated in Figure 1.

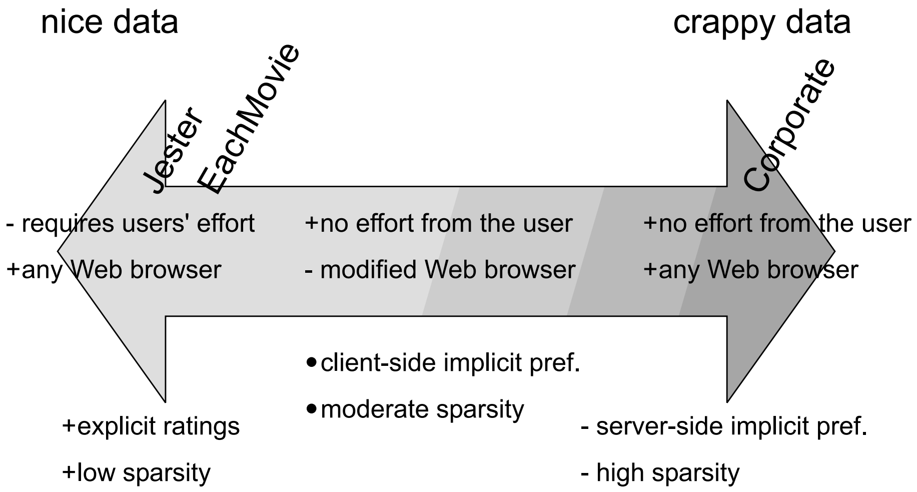


Fig. 1. Data characteristics that influence the data quality, and the positioning of the three datasets used in our experiments, according to their properties.

4 Evaluation platform

To understand the influence that highly sparse server-side collected data has on the accuracy of collaborative filtering, we built an evaluation platform. This platform is a set of modules arranged into a pipeline. The pipeline consists of the following four consecutive steps: (i) importing a user-item matrix (in the case of implicit preferences, data needs to be preprocessed prior to entering the pipeline), (ii) splitting data into a training set and a test set, (iii) setting a collaborative filtering algorithm (in the case of the kNN algorithm we also need to specify a similarity measure) and an evaluation protocol, (iv) making predictions about users' ratings and collecting evaluation results. The platform is illustrated in Figure 2.

Let us briefly discuss some of these stages. In the process of splitting the data into a training set and a test set, we randomly select a certain percentage of users (i.e. rows from the user-item matrix) that serve as our training set. The training set is, in the case of the kNN algorithm, used to search for neighbors or, in the case of model-based approaches, as a source for building a model. Ratings from each user from the test set are further partitioned into "given" and "hidden" ratings, according to the evaluation protocol. For example, 30% of randomly selected ratings from a particular user are hidden, the rest are treated as our sole knowledge about the user (i.e. given ratings). Given ratings are used to find neighbors in the training set, while hidden ratings are used to evaluate the accuracy of the selected collaborative filtering algorithm. The algorithm predicts the hidden ratings and since we know their actual values, we can compute the mean absolute error (MAE) or apply some other evaluation metric.

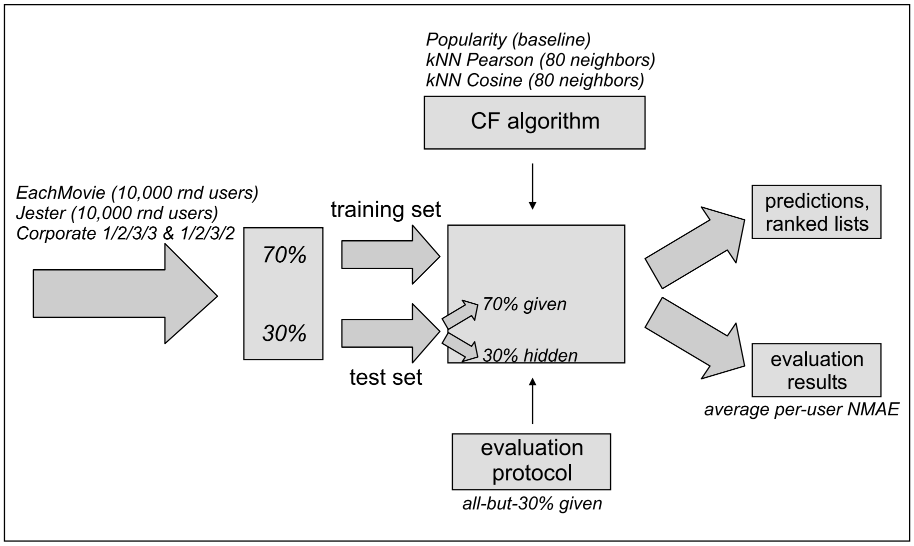

Popularity (baseline)
kNN Pearson (80 neighbors)
kNN Cosine (80 neighbors)
CF algorithm
EachMovie (10,000 rnd users)
Jester (10,000 rnd users)
Corporate 1/2/3/3 & 1/2/3/2
70%
30%
training set
test set
70% given
30% hidden
predictions, ranked lists
evaluation results
average per-user NMAE
evaluation protocol
all-but-30% given

Fig. 2. The evaluation platform. The notes in *italics* illustrate our experimental setting (see Section 6).

5 Data description

For our experiments we used three distinct datasets. The first dataset was EachMovie (provided by Digital Equipment Corporation) which contains explicit ratings for movies. The service was available for 18 months. The second dataset with explicit ratings was Jester (provided by Goldberg et al.) which contains ratings for jokes, collected over a 4-year period. Users were using a scrollbar to express their ratings – they had no notion of actual values. The third dataset was derived from real-life corporate Web logs. The logs contain accesses to an internal digital library of a fairly large company. The time-span of acquired Web logs is 920 days. In this third case the users' preferences are implicit and collected on the server side, which implies the worst data quality for collaborative filtering.

In contrast to EachMovie and Jester, Web logs first needed to be extensively preprocessed. Raw logs contained over 9.3 million requests. First, failed requests, redirections, posts, and requests by anonymous users were removed. We were left with slightly over 1.2 million requests (14% of all the requests). These requests, however, still contained images, non-content pages (such as index pages), and other irrelevant pages. What is more, there were several different collections of documents in the corporate digital library. It turned out that only one of the collections was relevant for the application of collaborative filtering. Thus, the amount of potentially relevant requests dropped drastically. At the end we were left with only slightly over 20,500 useful requests, which is 0.22% of the initial database size.

The next problem emerged from the fact that we needed to map implicit preferences contained in log files, into explicit ratings. As already explained, this is not a trivial task. The easiest way to do this is to label items as 1 (accessed) or 0 (not accessed) as also discussed in Breese et al. (1998). The downside of this kind of mapping is that it does not give any notion of likes and dislikes. Claypool et al. (2001) have shown linear correlations between the time spent reading a document and the explicit rating given to that same document by the same user (this was already published by Konstan et al. (1997)). However, their test-users were using specialized client-side software, which made the collected data more reliable (hence, in their case, we talk about client-side implicit preferences). Despite this fact we decided to take reading times into account when preprocessing Web logs.

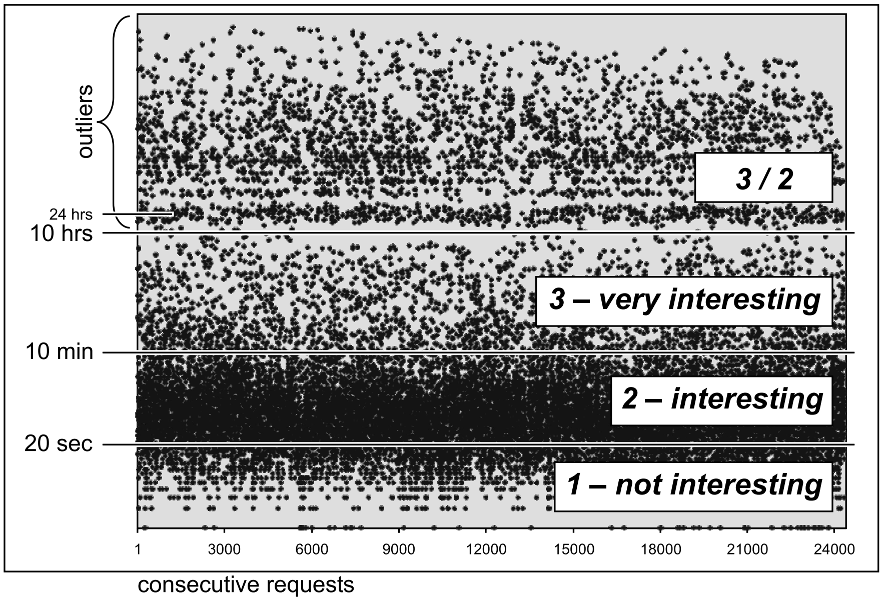
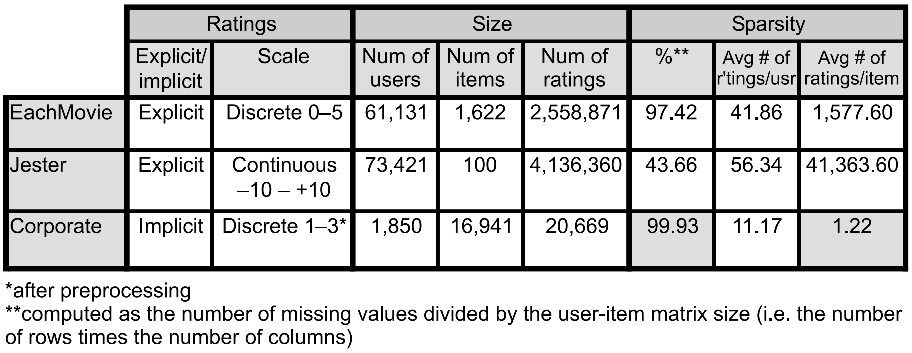


Fig. 3. Mapping implicit preferences contained in the corporate Web logs onto a discrete 3-score scale.

We plotted reading times inferred from consecutive requests onto a scatter plot shown in Figure 3. The X-axis shows requests ordered by their time-stamps, and the y-axis shows the inferred reading time on a logarithmic scale. We can see that the area around 24 hours is very dense. These are the last accesses of a day. People went home and logged in again the next day, which resulted in approximately 24-hour "reading" time. Below the 24-hour line, at approximately 10-hour reading time, a gap is evident. We decided to use this gap to define outliers – accesses above the gap are clearly outliers. We decided to map reding times onto a discrete 3-score scale (scores being 1="not interesting", 2="interesting", and 3="very interesting"). Somewhat ad-hoc (intuitively) we defined two more boundaries: one at 20 seconds and another at 10 minutes. Since items were research papers and 20 seconds is merely enough to browse through the abstract, we decided to label documents with reading times below 20 seconds as “not interesting”. Documents with reading times between 20 seconds and 10 minutes were labelled as “interesting” and documents with reading times from 10 minutes to 10 hours were labelled as “very interesting”. We decided to keep the outliers due to the lack of data. In the first scenario they were labelled as “very interesting” and in the second one as “interesting”. Since we had no reliable knowledge about the outliers, the second scenario should have minimized the error we made by taking them into account.

Table 1 shows the comparison between the three datasets. It is evident that a low number of requests and somewhat ad-hoc mapping onto a discrete scale are not the biggest issues with our corporate dataset. The concerning fact is that the average number of ratings per item is only 1.22, which indicates extremely poor overlapping. Sparsity is consequently very high, 99.93%. The other two datasets are much more promising. The most appropriate is the Jester dataset with very low sparsity, followed by EachMovie with higher sparsity but still relatively high average number of ratings per item. Also, the latter two contain explicit ratings, which means that they are more reliable than the corporate dataset (see also Figure 1).



	Ratings		Size			Sparsity		
	Explicit/ implicit	Scale	Num of users	Num of items	Num of ratings	%**	Avg # of r'tings/usr	Avg # of ratings/item
EachMovie	Explicit	Discrete 0–5	61,131	1,622	2,558,871	97.42	41.86	1,577.60
Jester	Explicit	Continuous –10 – +10	73,421	100	4,136,360	43.66	56.34	41,363.60
Corporate	Implicit	Discrete 1–3*	1,850	16,941	20,669	99.93	11.17	1.22

*after preprocessing
**computed as the number of missing values divided by the user-item matrix size (i.e. the number of rows times the number of columns)

Table 1. The comparison between the three datasets.

6 Experimental setting

We ran a series of experiments to see how the accuracy of collaborative filtering recommendations differs between the three datasets (from EachMovie and Jester we considered only 10,000 randomly selected users to speed up the evaluation process). First, we randomly selected 70% of the users as our training set (the remaining 30% were our test set). Ratings from each user in the test set were further partitioned into “given” and “hidden” ratings according to the “all-but-30%” evaluation protocol. The name of the protocol implies that 30% of all the ratings were hidden and the remaining 70% were used to form neighborhoods in the training set.

We applied three variants of memory-based collaborative filtering algorithms: (i) k-Nearest Neighbors using the Pearson correlation (kNN Pearson), (ii) k-Nearest Neighbors using the Cosine similarity measure (kNN Cosine), and (iii) the popularity predictor (Popularity). The latter predicts the user’s ratings by simply averaging all the available ratings for the given item. It does not form neighborhoods and it provides each user with the same recommendations. It serves merely as a baseline when evaluating collaborative filtering algorithms (termed “POP” in Breese et al. (1998)). For kNN variants, we used a neighborhood of 80 users (i.e. k=80), as suggested in Goldberg et al. (2001). We decided to evaluate both variants of the corporate dataset (the one where the outliers were labelled as “very interesting”, referred to as “1/2/3/3”, and the one where the outliers were labelled as “interesting”, referred to as “1/2/3/2”).

For each dataset-algorithm pair we ran 5 experiments, each time with a different random seed (we also selected a different set of 10,000 users from EachMovie and Jester each time). When applying collaborative filtering to the variants of the corporate dataset, we made 10 repetitions (instead of 5) since these datasets were smaller and highly sparse, which resulted in less reliable evaluation results. Thus, we ran 90 experiments altogether.

We decided to use normalized mean absolute error (NMAE) as the accuracy evaluation metric. We first computed NMAE for each user and then we averaged it over all the users (termed “per-user NMAE”) (see Herlocker et al. (2004)). MAE is extensively used for evaluating collaborative filtering accuracy and was normalized in our experiments to enable us to compare evaluation results from different datasets.

7 Evaluation results

Our evaluation results are shown in Figure 4. The difference between applying kNN Pearson and kNN Cosine to EachMovie is statistically insignificant (we used two-tailed paired Student’s t-Test to determine if the differences in results are statistically significant). However, they both significantly outperform Popularity. In the case of Jester, which has the smallest degree of sparsity, kNN Pearson slightly, yet significantly outperforms kNN Cosine. Again, they both significantly outperform Popularity. Evaluation results from the corporate datasets (two variants of the same dataset, more accurately) show that predictions are less accurate and that NMAE value is relatively unstable (hence the large error bars showing standard deviations of NMAE values). The main reason for this is low/no overlapping between values (i.e. extremely high sparsity), which results in inability to make several predictions. In the first scenario (i.e. with the 1/2/3/3 dataset) we can see that the differences in NMAE of kNN Pearson, kNN Cosine and Popularity are all statistically insignificant. In the second sce-

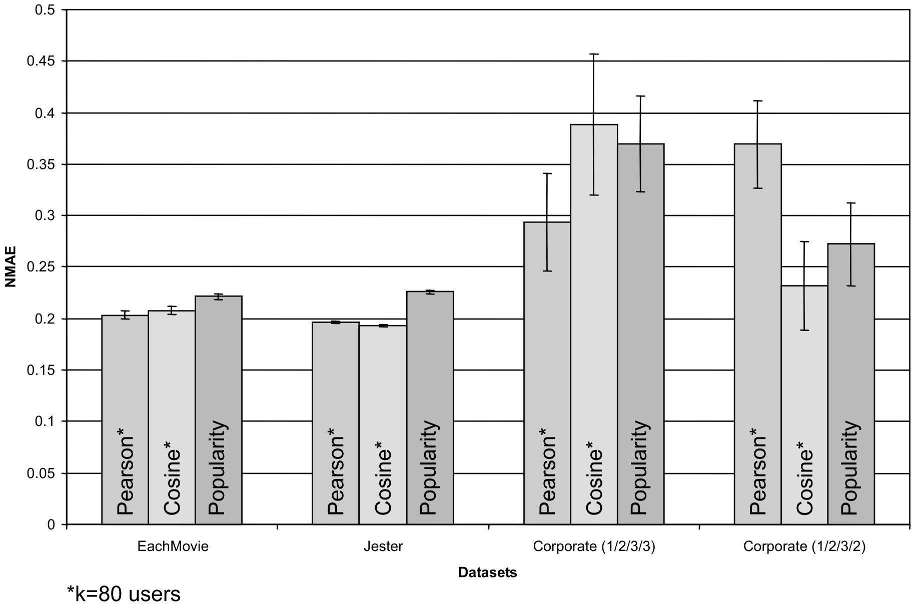


Fig. 4. The evaluation results.

nario (i.e. with the 1/2/3/2 dataset), however, kNN Pearson outperforms kNN Cosine and Popularity, while the accuracies of kNN Cosine and Popularity are not significantly different.

8 Discussion and future work

What is evident from the evaluation results is that the corporate dataset does not contain many overlapping values and that this represents our biggest problem. Before we will really be able to evaluate collaborative filtering algorithms on the given corporate dataset, we will need to reduce its sparsity. One idea is to apply LSI (latent semantic indexing) (Deerwester et al. (1990)) or to use pLSI (probabilistic latent semantic indexing) (Hofmann (1999)) to reduce the dimensionality of the user-item matrix, which consequently reduces sparsity. Another idea, which we believe is even more promising in our context, is to incorporate textual contents of the items. There were already some researches done on how to use textual contents to reduce sparsity and improve the accuracy of collaborative filtering (Melville et al. (2002)). Luckily we are able to obtain textual contents for the given corporate dataset.

What is also evident is that mapping implicit into explicit ratings has great influence on the evaluation results. We can see that going from Corporate 1/2/3/3 to Corporate 1/2/3/2 is fatal for kNN Pearson (in contrast to kNN Cosine). This needs to be investigated in greater depth; we do not wish to draw conclusions on this until we manage to reduce the sparsity and consequently also the standard deviations of NMAE values.

Also interesting, the Cosine similarity works just as well as Pearson on EachMovie and Jester. Early researches show much poorer performance of the Cosine similarity measure (Breese et al. (1998)).

As a side-product we noticed that the true value of collaborative filtering (in general) is shown yet when computing NMAE over some top percentage of eccentric users. We defined eccentricity intuitively as MAE (mean absolute error) over the overlapping ratings between "the average user" and the user in question (greater MAE yields greater eccentricity). The average user was defined by averaging ratings for each particular item. This is based on the intuition that the ideal average user would rate every item with the item's average rating. The incorporation of the notion of eccentricity can give the more sophisticated algorithms a fairer trial. We computed average per-user NMAE only over the top 5% of eccentric users. The power of the kNN algorithms over Popularity became even more evident. In near future, we will define an accuracy measure that will weight per-user NMAE according to the user's eccentricity, and include it into our evaluation platform. We will also consider ways of handling the more eccentric users differently.

Acknowledgements

This work was supported by the Slovenian Ministry of Education, Science and Sport and the IST Programme of the European Community under SEKT Semantically Enabled Knowledge Technologies (IST-1-506826-IP) and PASCAL Network of Excellence (IST-2002-506778). This publication only reflects the authors' views. The EachMovie dataset was provided by Digital Equipment Corporation. The Jester dataset is courtesy of Ken Goldberg et al.. The authors would also like to thank Tanja Brajnik for her help.

References

1. BALDI, P., FRASCONI, P., and SMYTH, P. (2003): Modelling and Understanding Human Behavior on the Web. In: *Modelling the Internet and the Web, ISBN: 0-470-84906-1, 171–209*.
2. BREESE, J.S., HECKERMAN, D., and KADIE, C. (1998): Empirical Analysis of Predictive Algorithms for Collaborative Filtering. In: *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence*.
3. CLAYPOOL, M., LE, P., WASEDA, M., and BROWN, D. (2001): Implicit Interest Indicators. In: *Proceedings of IUI'01*.
4. DEERWESTER, S., DUMAIS, S.T., and HARSHMAN, R. (1990): Indexing by Latent Semantic Analysis. In: *Journal of the Society for Information Science, Vol. 41, No. 6, 391–407*.
5. GOLDBERG, K., ROEDER, T., GUPTA, D., and PERKINS, C. (2001): Eigentaste: A Constant Time Collaborative Filtering Algorithm. In: *Information Retrieval, No. 4, 133–151*.

6. GRCAR, M. (2004): User Profiling: Collaborative Filtering. In: *Proceedings of SIKDD 2004 at Multiconference IS 2004, 75–78.*
7. HERLOCKER, J.L., KONSTAN, J.A., TERVEEN, L.G., and RIEDL, J.T. (2004): Evaluating Collaborative Filtering Recommender Systems. In: *ACM Transactions on Information Systems, Vol. 22, No. 1, 5–53.*
8. HOFMANN, T. (1999): Probabilistic Latent Semantic Analysis. In: *Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence.*
9. KONSTAN, J.A., MILLER, B.N., MALTZ, D., HERLOCKER, J.L., GORDON, L.R., and RIEDL, J. (1997): GroupLens: Applying Collaborative Filtering to Usenet News. In: *Communications of the ACM, Vol. 40, No. 3, 77–87.*
10. MELVILLE, P., MOONEY, R.J., and NAGARAJAN, R. (2002): Content-boosted Collaborative Filtering for Improved Recommendations. In: *Proceedings of the 18th National Conference on Artificial Intelligence, 187–192.*
11. RESNICK, P., IACOVOU, N., SUCHAK, M., BERGSTROM, P., and RIEDL, J. (1994): GroupLens: An Open Architecture for Collaborative Filtering for Netnews. In: *Proceedings of CSCW'94, 175–186.*

Information Delivery for the End User of the Semantic Web

Krzysztof Węcel
Department of Management Information Systems
The Poznań University of Economics, Poland
K.Wecel@kie.ae.poznan.pl

Anna V. Zhdanova
DERI – Digital Enterprise Research Institute
University of Innsbruck, Austria
anna.zhdanova@deri.org

Abstract. We propose the information delivery process for the end user of the Semantic Web, which was divided into three main steps: Collection, integration and aggregation step, Filtering or querying step and Presentation step. Contemporary search engines are our starting point. We analyze them from the users' point of view: how they support users, and which user requirements they try to approach. We also develop a scenario to show how the Semantic Web may solve the problems analyzed. Further we focus on presentation and interfaces for information delivery, since it affects the most overall users' experience in search for the relevant information.

1 Introduction

Information overflow was identified as a problem a long ago: the terms *electronic junk* [1], *information overload* [2] exist for more than 20 years. A large amount of the development in information systems is devoted to delivering to the final user an appropriate amount of information. This is particularly important for the Web where the information is abundant. Many techniques have been developed within information retrieval and filtering [3]. Still, there is a lot of work to be done, and certainly this work should focus on end users. As Lipetz noticed, we would be able to fully satisfy information consumers "when researchers gained a deeper understanding of how humans process information and then endowed machines with analogous capabilities" [4]. So far, we have not achieved such a level of cognition, but new technologies are taking us closer to that goal. One of such promising technologies is the Semantic Web [5].

Some people may claim that the Semantic Web (SW) is quite close to aforementioned objective, as it provides means to represent knowledge (or semantics) in a machine processable form. However, models for knowledge representation have existed before the Semantic Web. Assisting humans with means for efficient search and delivery of information remains to be a challenge on the Semantic Web.

For a better understanding of how people look for the information, we have to draw our attention to user aspects of the Semantic Web environments. However, in the literature the technical approach is prevailing. Therefore we observe the opposite results than promised. Although the Semantic Web is gaining popularity, there are still problems with access to the information:

- the Semantic Web is developed mostly in an unsupervised manner, forming isolated "islands" of ontology and technology reuse
- methodologies and tools that are created are not widely accepted
- the Semantic Web is still too vast to a regular user.

Seemingly ironical, information overflow problem is inherited to the Semantic Web as it exists on the Web. In this paper we propose an approach for user-oriented information delivery and search for coping the information overflow problems on the Semantic Web.

The paper is organized as follows. In Section 2, we provide background of the problem and motivation. In Section 3, we analyze current improvements of the search engines, which are inspirations for better information delivery. In Section 4, we propose the information delivery process, and Section 5 concludes the paper.

2 Background

Information delivery is closely related to searching for the information. Therefore, in order to analyze and present problems that end user may encounter while using the Semantic Web, we refer to the search engines. The analysis is supported by a scenario. We also draw a focus on the user aspects. In scenario we supposed that certain communities and their members create ontologies and certain communities and their members provide the data, therefore users of the information systems and their roles are analyzed.

2.1 Google's Lessons

Search engines have been used almost since the Web went public. Now we observe mainly incremental improvements in search engines technology, and only few breakthroughs have been seen. Last significant improvement was done by Google [6]. Unfortunately, since then people have learned how to misuse Google, e.g., utilize PageRank algorithm to manipulate the results. Nevertheless, people got used to good results from Google and expect further improvements.

The common problems in search can be divided into three classes:

- type of content
- the content itself
- bias in weights.

First, restricting search to particular type of content is not possible. We are not thinking about file types (e.g. PDF, PPT), what is already implemented, but more general categories, like “scientific paper”, “article in encyclopedia”, “definition in dictionary”, sale offer, auction etc. Provided that there are similar numbers and importance of referring pages, referred pages are ranked equally no matter if it is a sale offer or scientific publication, or just a fake page containing prepared set of keywords. And of course, for different users it has different importance. For example, users complain that they often get sale offer when looking for artist information instead of informative content, e.g. biography. Giving the possibility to constrain type of content would significantly improve the search results.

Secondly, there is sometimes a problem with the precision of the content. We get the appropriate type of content, but that content is not semantically coherent to what we expected. Google is just missing context of information. When one types “jaguar”, one receives at lest three clusters of information. Within the top results there is information about cars, about big cats, and surprisingly about Apple’s Mac OS X. The last one codenamed Panther is compared to jaguar only in one sentence. Because of the popularity of Apple’s webpage, “jaguar” there also seems to Google to be important, what is not justified. Further experiment, when we type “panther” in Google, the first result is not a web page on cats but also the main page of Apple. The issue of content matching is not resolvable without introduction of semantics and probably certain human intervention.

The last, third, issue is to some extent connected with the first two. Google’s PageRank uses links and keywords to compute weights and create ranking. In most cases it produces superior rankings of pages. On one hand, the bias in weights may be caused unintentionally for example because of the type of content which is generated automatically from the database. On the other hand, algorithm is well known, and people have learned how to manipulate weights. This unfortunately deteriorates the search results. Either we can find information very quickly or it is really hard to find it. We can modify the keywords but it does not always help.

2.2 “I need this specific information”

Suppose that new employee came to the organization and would like to get to know his co-workers. Usually, there is a company webpage that presents the list of all employees, in which department their work, contact information, sometimes responsibilities. This webpage is very formal and contains only information related to the company. Personal information, which is rather crucial in a social life in a company: photos, hobbies, birthdays, etc can be missing. Some of the users may have built their personal pages, but only rarely a link to that page is present on official employee webpage.

The newcomer has some possibilities. One of them is to launch a web browser, go to a search engine and look for the information somewhere in the Internet. Several problems arise: the query should be repeated for every employee. Moreover, the query will not be unambiguous as we have seen in the previous section. Specifying only first name and family name will return hundreds or thousands of pages. The search engine will not distinguish "John Green" that we are looking for among the other people with the same name; hence there is a need to read most of the result pages. And we are not sure if our searches will succeed: does everybody have a webpage? Further, the user is burdened with integration of the information, and it requires additional effort. The problems encountered so far: manual search for the information, collection of the distributed information, extraction of heterogeneous sources, integration of the information, transforming of the aggregated information into visual form. This tedious task may be made easier by using appropriately structured information. There are some solutions that more or less support this, e.g. FOAF – Friend of a Friend [7], but they are not mature yet.

2.3 Users and Roles

According to the class of information systems, we can distinguish different classes of the users. If we look at the Internet, the basic division is into active users and passive users. Passive users just browse the Internet or navigate from page to page, use search engines to find the information. The most characteristic is that they do not contribute with their own information. Active users are the opposite; they publish new content on the Internet. The classification presented is not unambiguous. Some of the users may become active. Therefore it is better to speak about roles (like in workflow management systems). A user may play different roles according to the context or situation. Because main substance exchanged on the Internet is information, we may talk as well about information consumer role and provider role.

Yet another classification of users stems directly from information society, which is supposed to be built by bringing information technology to the masses. User may use IT to the different extent, and thus play different role in information society, therefore we can distinguish [8]:

- self-informing citizens – know the technology, so they are able to acquire relevant information
- communicating citizens – can communicate with other people in an electronic way
- citizens educating themselves – acquire knowledge that determines the quality of their professional and private lives
- creative citizens – can create digital products or provide digital services which meet the needs of self-informing, communicating and educating citizens.

However, if we focus only on information providers (or creating citizens) we will see that they may be further layered. Both user filling in a form and designer of a portal are information providers. Furthermore, the user may provide the content alone, or in collaboration with other users. Also, the scope of the knowledge used may be different: one may be interested only in instances from a knowledge base, another in structure the knowledge base, i.e. in ontologies.

There are different activities related to the information delivery:

- structuring
- editing

- browsing.

First, a framework for knowledge representation should be created. Taking into account contemporary trends it will have a form of ontology. Commitment of many users is required therefore proper management is a must here. Then users may introduce their own information by creation of instances of the concepts taken from the ontology. It may also be done in a collaborative way. The first two activities may be jointly referred to and are covered by ontology management. Finally, another group of users may browse the knowledge base for the required information. As a result of interaction, information may be delivered to the final user.

3 Towards the Semantic Web

Some of the problems addressed in the previous section can be solved by better use of the Semantic Web technology, especially in the support of the end-users. Main problem of search engines consisted in lack of semantics. To convince users of usefulness of the Semantic Web we need clear and easy to use interface and also outstanding search results.

Focus on end user is crucial. Different users differently perceive information. They have different abilities to cope with the abundant information. Also, the amount and type of information they need in their work is not the same for everybody. Taking all the factors that may influence information needs of the user we have obtain a so called user context, which may include user knowledge, user location, user activity. It will be also useful to keep a track of what user looked for and how did find information.

3.1 User Support

People will positively perceive the Semantic Web if it supports them in their activities in an easy manner. Every well-designed information system should suggest how to work with it. Semantic Web shall not be an exception here.

Today we can observe only many small improvements in various search engines. *Google suggest*[1] auto completes the search terms based on a few first letters, working similarly to combo box in Windows. Thus the query may be typed faster. AOL search engine supports users in another way: using its *Smartbox Suggestions* gives access not only to general purpose web search but also to more specialized search engine or even specialized databases, e.g. stock quotes.

In the Semantic Web search users should have the possibility to select options to narrow their query. Sometimes we may want to choose the type of information we are looking for, e.g. white paper, product info, advice from the discussion forum, technical problem, definition, biography etc., not to mention a picture. For a long time Google is offering a special search for pictures. Others also join, e.g. A9.com offers buttons on the right side of the window that allow restricting query for certain

[1] http://www.google.com/webhp?complete=1&hl=en

information: web, books, images, movies, reference, yellow pages. It is also possible to see the history of searches.

Other search engines also collect history of searches. This will be obviously also important in Semantic Web. The user may know that she had found the information once, but cannot remind how. This is especially addressed in one of the Microsoft's projects *Stuff I've seen*[2], which will be included in Longhorn.

All these suggestions cause that if user already knows or may know something, she does not have to start from scratch.

3.2 User Context

Introduction of context will allow answering the question how to intelligently reduce amount of information in an answer to the query. Information needs are related to user activities, therefore it will be useful to take them into account. We can distinguish many contexts: time, space, user's knowledge, users' history etc.

One of the most visible contexts is geographical context. According to Microsoft's MSN Search about a quarter of all searches refer to geographic information[3]. Therefore the user has the possibility to search only pages relating to her location. "NearMe" button can return results based on proximity to a place. Unfortunately, it does not work for Innsbruck. When we typed "Japanese restaurant" or "theatre" there were no results. Typing "Innsbruck restaurant" helped, which shows that the location discovery is not well elaborated.

Another example of geographic information is AOL. It is capable of distinguishing some geographical names, and present possible contexts to the user. However, it does not affect effectiveness of retrieval greatly. It may be useful but not precise. For example "Warsaw (US City)" and "Warsaw (International city)" yield the same results. When we compare "Poland (US City)" and "Poland (country)", the results differ, although they are mixed – no real distinction between city and country.

A noticeable application of geographical context was introduced in January 2005 by A9.com. In the Yellow Pages service it is possible not only to look for information on local businesses but also display their photos taken from the street. Moreover, it is also possible to take a virtual walk and see information about other businesses which are seen on the photo. This feature is called "Block View". Such functionality is available for several cities, including New York, Atlanta, San Francisco and Seattle.

We can also look at context from the results' point of view. One possibility to use context is during query formulation, and another while interpreting results. Some of the search engines present clustered results, e.g. Northern Light. That is also a good proposal for improving usability of the Semantic Web, when users are not aware if there are different meanings of the query. It may be a solution for Google's problem, i.e. too many documents on one topic, and lack of documents for another topic, represented by the same set of keywords.

[2] http://research.microsoft.com/adapt/sis/

[3] http://www.msn.com

As part of user context we may also consider vertical searches. As in case presented by us in the previous section, users usually have very specific questions, e.g., find me all instances of class Employee. It means that usually they have the idea of what they are looking for. From the interface to the Semantic Web they expect help in refining their queries. Also in this direction we may observe some research. Amazon's A9.com has opened its search site to specialized search engines. Users may select thousands of vertical search options. As Bezos, CEO of Amazon, said, they want to "do for search what RSS has done for content." The added value of this approach is subject-matter expertise; it is very similar to ontology layering: upper-level vs. domain ontologies. In the next section we show that such vertical knowledge bases may be developed by different communities, and thus improving the overall quality. Company expects that there will be a significant number of vertical search engines that will be interested in joining the project. Better search results should be achieved by limiting number of sources that are looked up for relevant information.

3.3 From Databases to the Semantic Web

More and more search engines associate databases with query, for example Yahoo weather, movies on AOL, books on A9.com. As Ramez Naam (MSN Search) said "Having the trusted data, what we know is a right answer, and not asking them to trawl around, that's a huge advantage for the user."

In databases there is a lot of digital content that is usually not visible to the search engines, unless somebody puts some effort on integration. Resources are generated on demand, and therefore it is called a hidden web. It requires different indexing mechanism.

A database is not what the end user would like to use for representing knowledge about the world. It has fixed structure and is not flexible in storing different kinds of information. Nevertheless, it is better to have metadata on it and retrieve information on demand, not just to have to annotate all the documents with sophisticated algorithm without being sure if it is done correctly. For a Semantic Web it is as good basis, but it is not enough. Another issue is delivery of the information. From a database, it is easy to create well annotated documents, but still it is not convenient for information seekers.

So far search engines have developed certain solutions. Ask Jeeves introduces new technology that will further extend the answering capabilities of its engine. New feature is called *Direct Answers From Search* and consist in searching for natural language questions across entire Web rather than focusing on own database. This is the idea closest to the Semantic Web.

3.4 Community-Driven Approach

In contemporary search engines we observe two factors that negatively affect the precision of the returned results:

- information is weakly structured
- lack of human annotations.

The first problem may be overcome by the Semantic Web. It is easy to talk about semantics from the technical point of view. For computers our annotations are merely strings of characters.

The latter problem requires engagement of people. The semantics in order to be used in a broadly understood user context, should be first introduced by somebody else. Thus we came to the point where human intervention is required. Due to the large effort required to create the content, one has to take into account that a large number of users will be involved into creation and evolution of the Semantic Web. For example, semantics of sources may be enhanced by means of ontology acquisition from Web users [9]. We believe that distributed online content developed by user communities strongly influences the information delivery process.

4 Information Delivery

Distributed community-generated Semantic Web content is published and accessed differently comparing to the ordinary Web content. In particular, Web content is normally generated in a centralized way, and a webmaster has an overview of the web-site content and has control over delivery of the content to the final user. For the Semantic Web, existing information search practices (e.g., search engines discussed in Section 2), recommendation practices (e.g., established by Amazon.com), accessibility practices [10] are not sufficient and not trivial to apply. In this section, at first, we present a model for information delivery process of distributed community-driven Semantic Web content. Further, we identify points important for usability and accessibility guidelines for delivering distributed Semantic Web content. Finally, we show that the specified process and guidelines are applicable in the context of the Semantic Web to the “I need this specific information” scenario described in Section 2.

4.1 Information Delivery Process

Generalizing current experiences of presentation and delivery of the distributed community-generated Semantic Web content, we present delivery process for such content. In Fig. 1, the steps of information delivery process on the Semantic Web are depicted.

Initially, content is distributed over the Web as the communities develop and specify it. As for the Web content delivery, the main steps in delivery of the Semantic Web content to the final user are (1) collection, integration and aggregation, (2) filtering or querying, (3) presentation of the content. Meanwhile, unlike the Web content, the Semantic Web content is not necessarily associated with human-oriented presentation data, and therefore presentation of the Semantic Web content to the end user in a human-readable and accessible form is a problem requiring a solution. Below, we identify steps in the overall process of delivery of the distributed Semantic Web content to the end user.

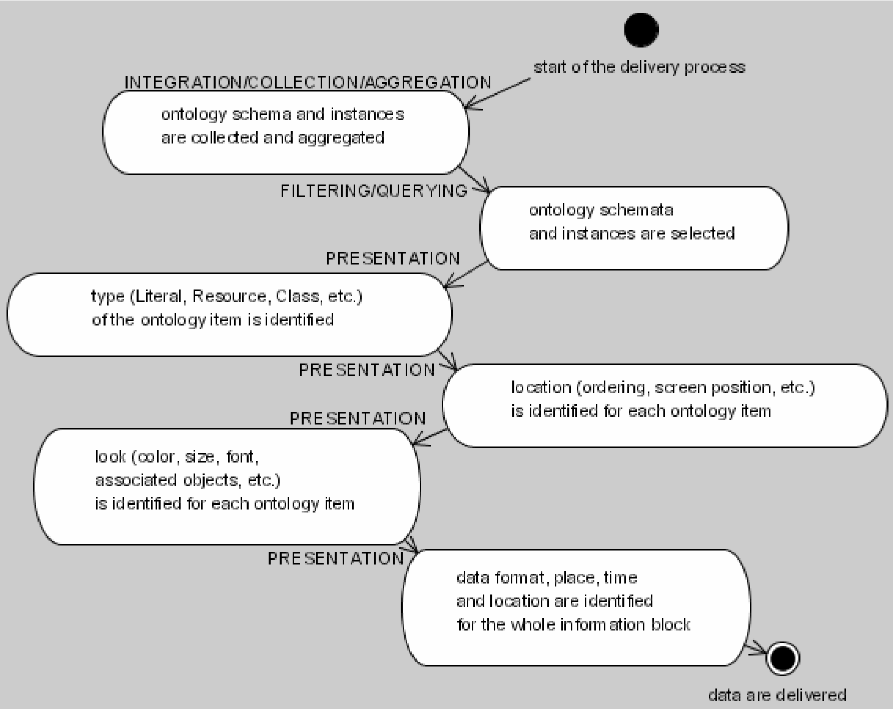

start of the delivery process
INTEGRATION/COLLECTION/AGGREGATION
ontology schema and instances are collected and aggregated
FILTERING/QUERYING
ontology schemata and instances are selected
PRESENTATION
type (Literal, Resource, Class, etc.) of the ontology item is identified
PRESENTATION
location (ordering, screen position, etc.) is identified for each ontology item
PRESENTATION
look (color, size, font, associated objects, etc.) is identified for each ontology item
PRESENTATION
data format, place, time and location are identified for the whole information block
data are delivered

Fig. 1. Information Delivery Process

Collection, Integration and Aggregation step:

1) The ontology schemata and instance data should be continuously integrated, collected and aggregated. This process is similar to indexing known from the classical search engines. There are several solutions that crawl the Web and extract semantic information, e.g., SemanticWebSearch[4]. into information set which is of potential relevance to the final user.

Filtering or querying step:

2) As the amount of data of potential interest to the final user can be larger than the user can access (information overflow problem), the data should be downsized to its subset.

There can be two different approaches to get information from the Semantic Web: push and pull. The first one can be related to already known information filtering. In this case user gets overview of changes in the Semantic Web according to her profile. Profile represents relatively stable information needs. The latter one resembles information retrieval, where user specifies queries. Query represents temporary user needs. Unlike in the first case, this delivery is done on demand.

[4] http://www.semanticwebsearch.com

Presentation steps:

3) The ontology instances should be identified by type. Knowing the type of the instances is necessary, as a mechanism of rendering can be specified with the help of ontologies supporting rendering processes. For example, an instance of a class *Person* can be specified to be shown in a specific color with certain associated ontology concept or property values, such as *Name* and *Email address*.

4) The location of the ontology and ontology items (classes, properties, instances, etc.) on the screen is established. Specifically, the order of the items on the screen and their positions are established.

5) At this step, visual characteristics of each ontology item should be identified, such as the item's color, size, font and objects that are associated with an item and need to be shown on the screen for adequate rendering of the item. Such associated objects can be images, multimedia, etc.

6) At the last step, the commonly used personalization techniques [11] are applied, namely delivery of information relevant to an individual or a group of individuals in the format and layout specified and in time intervals specified.

After all the steps are executed in turn, the data are being delivered to the end user.

4.2 Information Delivery Interfaces

In this subsection, we identify the application and human related features substantial for the development of the information delivery processes, and illustrate them with the state-of-the art examples. We focus on the end user interfaces resulting after presentation steps of the information delivery process of the Semantic Web content, and particularly, on their accessibility and usability. Despite a high number of works on Semantic Web visualization [12], accessibility and usability features of user-side of Semantic Web content delivery interfaces were not explicitly identified before.

4.2.1 Interfaces for Semantic Web Applications

The following features are substantial in construction of information delivery related interfaces for the Semantic Web applications.

1) Satisfying Software-Related Requirements: Content Negotiation

When an application (e.g., a Web browser) requests information, reception of different content depending on the requester (e.g., graphical images if they are supported by the application or a textual description otherwise) is possible[5]. However, existing protocols do not allow applications to request ontological data of certan types, i.e., operation with Semantic Web annotations remains underspecified in the content negotiation practices.

[5] Apache HTTP Server Content Negotiation, http://httpd.apache.org/docs/ content-negotiation.html

2) Satisfying Hardware-Related Requirements: Different Reception Devices

As well as the Web content, the Semantic Web content can be accessed with different means: personal computers, mobile phones, etc. The delivered content depends on the device of delivery by quality and quantity. Supporting negotiation techniques for identification of the content preferred by the device on the basis of semantic annotations would be a step towards semantically enabled cross-device information delivery.

4.2.2 Interfaces for Human Users

The following features are substantial in construction of information-rendering end user interfaces on the Semantic Web.

1) Supporting Simple-to-use Navigation and Orientation

Web pages, resulting from Semantic Web content and further post-processing, should enable the final user to easily locate the required data on the pages, and easily switch to accessing next sets of Semantic Web content.

2) Making the Context of the Information Explicit to the User

Keeping the user aware of the context of the represented material is important. For example, if an application allows a user to change ontology items, the user should be aware of the consequences of his/her changes. Another example, if a user requests for information about “Warsaw”, the presentation of the ontological content should keep the user aware whether information about an US or Polish city is delivered.

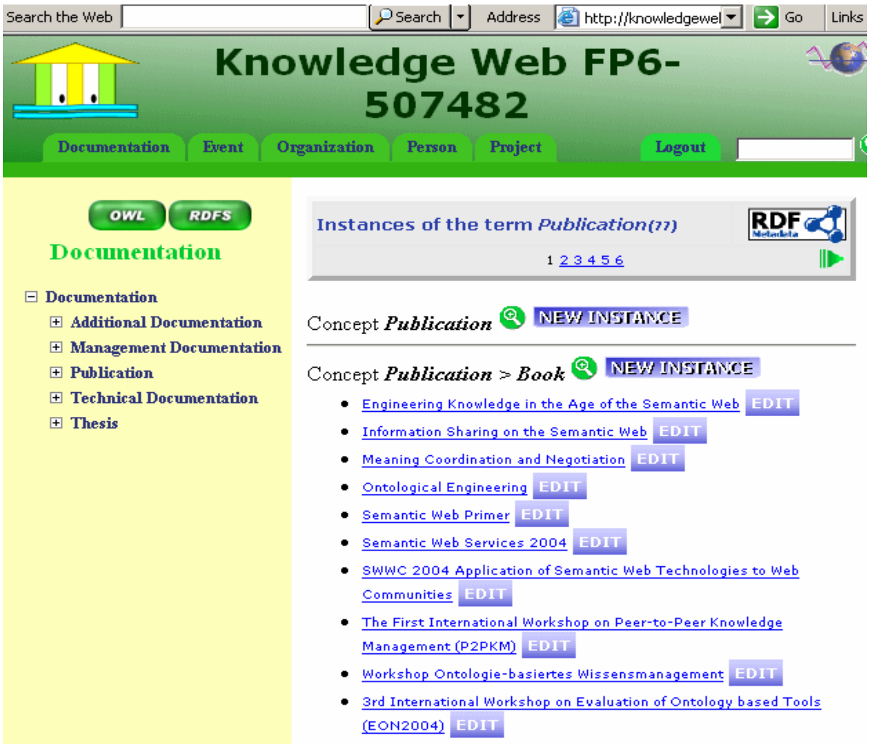
3) Automatically Organizing Semantic Web Content on the Device of the End User

Information of arbitrary quantity and quality arriving to the end user should be organized on the user’s receiving device (e.g. computer screen) in an accessible way without causing information overload on the page. If necessary, information can be presented on several cross-linked pages. On the Semantic Web, ontology-based algorithms can be applied to describe, analyze and adequately render arriving information. For example, after analysis of social networks of trust [13], information from less trusted sources can be automatically displayed in a less highlighted manner comparing to the information from more trusted sources.

4) Providing Visual Links to Semantic Web Annotations

Despite that the Semantic Web content is primarily made for machine consumption, experience reveals that humans expect to have a visible link to the Semantic data. In particular, buttons providing a link to the Semantic annotations are present at many applications delivering Semantic Web content, e.g., Knowledge Web portal[6] (Fig. 2) and People’s Portal [9] .

[6] Knowledge Web portal: http://knowledgeweb.semanticweb.org



Instances of the term *Publication*(77)

1 2 3 4 5 6

Concept *Publication*

Concept *Publication* > *Book*

- Engineering Knowledge in the Age of the Semantic Web EDIT
- Information Sharing on the Semantic Web EDIT
- Meaning Coordination and Negotiation EDIT
- Ontological Engineering EDIT
- Semantic Web Primer EDIT
- Semantic Web Services 2004 EDIT
- SWWC 2004 Application of Semantic Web Technologies to Web Communities EDIT
- The First International Workshop on Peer-to-Peer Knowledge Management (P2PKM) EDIT
- Workshop Ontologie-basiertes Wissensmanagement EDIT
- 3rd International Workshop on Evaluation of Ontology based Tools (EON2004) EDIT

Fig. 2. Access to Information Editing at the Knowledge Web Portal

5) Supporting Internationalization and Multilingualism

End users worldwide use different natural languages for communication. Delivering information in the most preferable natural language to the end user is another challenge for the Semantic Web applications. At the moment, there are agreed ways to annotate resources represented in certain natural languages (e.g., using XML and languages layered on top of XML). An ability to understand a certain language or a cultural context can be encoded in (semantic) profiles of individual users and user communities (e.g., adopting FOAF). When such user profiles are broadly available, matching resources and profiles to identify the content in the preferred natural language or cultural context is possible as a part of filtering step (step 2) in the information delivery process.

6) Supporting Disabled Users and Users with Special Requirements

Similar to the preferences of accessing information using one or another natural language, users might need to have the information rendered in special ways such as in an enlarged font (in case of poor sight), in a more granular manner (in case of employment of a small screen), etc. Information delivery in a manner accessible to disabled users and users with special requirements can also be assisted by specifying accessibility details in (semantic)-profiles of users and user communities, and taking data from these profiles as an input in information delivery process at the steps 3, 5, 6 (cf. subsection 4.1).

4.3 The Semantic Web Answer to the "I need this specific information" Scenario

As the information delivery process on the Semantic Web is specified, one can see that the integration, collection, aggregation and filtering, querying parts of the process become more formalized comparing to the Web. Meanwhile, the presentation part of the information delivery on the Semantic Web becomes a challenge. Unlike the Web applications, the Semantic Web applications normally need to render metadata which are evolving independently of visualization mechanisms for these specific data.

Let us consider the described in section 2 "I need this specific information" scenario, where a person starts to work in a new company and is interested in knowing more about her colleagues. If a company had a framework for representation of personal information, there could be one repository for holding references to chunks of personal information specified in semantic annotations. The scope of the information would be defined in an ontology. Every employee could update his personal information in conformance with the ontologies shared by the company members. This personal information could be easy to integrate and query. And the query that could be asked by a newcomer will be as easy as "show me all the instances of a class "http://www.mynewcompany.com/Employee" who have the value of attribute "http://www.mynewcompany.com/Hobby" specified. Meanwhile, as the company employees can evolve and query their profiles in an arbitrary manner, even a simple query might unexpectedly yield information set, presentation of which is not predefined in the framework. Therefore, developers of the applications delivering Semantic Web content to the end user should pay specific attention to ensuring accessibility and usability of the resulting interfaces.

5 Conclusions

Summarizing, there are not yet developed appropriate techniques to effectively support user in the usage of the Semantic Web. The technology starts to exist in the end-users' minds, but there are no agreements on what it actually is. There are also claims undermining the potential of this technology, stating that there are no problems to solve [14]. But indeed there are many problems.

Since the technology is promising and many people are eager to use it, we should think how encourage users of the Semantic Web. User interfaces are one of the issues, which we discussed in this paper. Security, immunity to exploitation and privacy are important issues here.

We foresee problems, and techniques for coping with them should be developed in advance. One of the problems is that the Semantic Web might not meet the users' expectations. When the Semantic Web technology becomes widespread, more and more people will contribute. The quality of contribution might become a problem. Therefore, measures should be taken to make sure that real collaboration on the Semantic Web occurs, and not only what we can call semi-collaboration – people publishing content without conforming to certain standards and propagating their own practices.. Having failed on establishment of community-driven approaches and collaboration will imply that users still will have the problems with finding relevant and credible information, even after introduction of the Semantic Web.

From users' point of view it is relatively easy to define requirements that will enable broad acceptance of this technology. Using the Semantic Web should be as easy as asking an expert for an advice or a friend for a rumor, and just getting an answer, without further need to process the information (e.g. read the document). Taking this approach we have to acknowledge that the Semantic Web should be invisible for the user, no matter how sophisticated are the underlying algorithms. Still those algorithms should also be improved.

Acknowledgement

The work is partly supported by the European Commission under the Marie Curie ToK *enIRaF* (IST-509766) and NoE *KnowledgeWeb* (IST-2004-507482).

References

1. Denning, P.J., *Electronic junk.* Communications of the ACM, 1982. **25**(3): p. 163-165.
2. Palme, J., *You have 134 unread mail! Do you want to read them now?*, in *Proc. of the IFIP WG 6.5 working conference on Computer-based message services*. 1984, Elsevier North-Holland, Inc.: Nottingham, United Kingdom. p. 175-184.
3. Bayeza-Yates, R. and B. Ribeiro-Neto, *Modern Information Retrieval*. 1999: ACM Press.
4. Lipetz, B.A., *Information Storage and Retrieval.* Scientific American, 1966.
5. Berners-Lee, T., J. Hendler, and O. Lassila, *The Semantic Web.* Scientific American, 2001. **284**(5): p. 34-43.
6. Brin, S. and L. Page, *The anatomy of a large-scale hypertextual Web search engine*, in *Proceedings of the seventh international conference on World Wide Web 7*. 1998, Elsevier Science Publishers B. V.: Brisbane, Australia. p. 107-117.
7. FOAF, *Friend of a Friend Project*. 2004.
8. Abramowicz, W., *Citizens of the global information society*, in *Poland and the Global Information Society: Logging on. Human development report*, W. Cellary, Editor. 2002, United Nations Development Programme: Warsaw. p. 121-134.
9. Zhdanova, A.V. *The People's Portal: Ontology Management on Community Portals*. in *1st Workshop on Friend of a Friend, Social Networking and the Semantic Web (FOAF'2004)*. 2004. Galway, Ireland.
10. Chisholm, W., G. Vanderheiden, and I. Jacobs, *Web Content Accessibility Guidelines 1.0, W3C Recommendation*. 1999.
11. Won, K., *Personalization: Definition, Status, and Challenges Ahead.* Journal of Object Technology, 2002. **1**(1): p. 29-40.

12. Geroimenko, V. and C. Chen, eds. *Visualizing the Semantic Web*. 2003, Springer.

13. Golbeck, J., B. Parsia, and J. Hendler. *Trust Networks on the Semantic Web*. in *Cooperative Intelligent Agents*. 2003. Helsinki, Finland.

14. Festa, P., *Next big step for the Web* - *or a detour?* 2005, ZDNet News.

Web Mining Approach for a User-centered Semantic Web

Junichiro Mori[1,2], Yutaka Matsuo[2], Koichi Hashida[2], and Mitsuru Ishizuka[1]

[1] University of Tokyo, Japan
jmori,ishizuka@miv.t.u-tokyo.ac.jp
[2] National Institute of Advanced Industrial Science and Technology (AIST), Japan
y.matsuo@carc.aist.go.jp, hashida.k@aist.go.jp

Abstract. In this paper, we propose a Web mining approach for the Semantic Web. The approach uses a search engine and the traditional web as a source of information to produce semantically rich information. In particular, we assess one community and obtain the social network and related information from the Web. As an example, we extract the social network of an academic society and show that extracted information can be incorporated into FOAF representation and utilized to measure the authoritativeness of a member in terms of social trust or individual trust. To demonstrate our Web mining approach in the real application, we show a researcher mining and retrieval system. Finally, we discuss the manner in which the Web mining approach contributes to availability to users of the Semantic Web.

1 Introduction

The Semantic Web [2] is designed to let users make explicit statements about any resource, and maintain that data themselves in an open and distributed manner. Several standards such as the Resource Description Framework (RDF) [18] and Web Ontology Language (OWL) [19] have been developed to realize the layer cake of the Semantic Web.

From the viewpoint of end users, expressing semantics about people and their relationships has garnered considerable interest. The Friend of a Friend (FOAF) project [4] is an extremely popular ontology of the Semantic Web [6]. It is essentially a vocabulary for describing people and whom they know. The FOAF ontology is not the only one people use to publish social information on the Web. For example, it is reported that more than 360 RDF Schema or OWL classes are defined with the local name “person” [1]. In fact, many vocabularies for user semantics have been developed [20, 5, 12].

Supported by these user-side ontologies, users are gradually coming to adopt Semantic Web technologies both explicitly and implicitly. For example, in Weblogs, which are diary-like sites, users attach a FOAF profile to a Weblog and publish various contents by the RDF site summary (RSS). Some social networking sites that allow users to maintain an online network of friends associates for social or business purposes publish their users’ social network data in FOAF format. Approaching the top of the Semantic

[1] http://swoogle.umbc.edu

Web layers, calculation of a “Web of Trust” on a FOAF-based network is also proposed [10].

Users are beginning to accept FOAF and its extensions as something of a standardized ontology for representing user semantics on the Semantic Web. While some users are explicitly authoring their FOAF files, others use FOAF file that systems automatically create using their Web pages. In fact, considering the personal information that the FOAF vocabulary expresses, we find that much information is contained in the traditional Web. For example, imagine a researcher: that researcher’s information might be in an affiliation page, a conference page, an online paper, or even in a Weblog. A method that can process the vast amount of information on the current “non-semantic” Web and can thereafter produce semantic information would facilitate and accelerate the use of the Semantic Web. For example, reusing existing sources of information on the Web would solve semantic annotation problems by helping users to create their metadata.

In this paper, we propose a Web mining approach for the Semantic Web. The approach uses a search engine and the traditional web as an information resource to produce semantically rich information. In particular, we examine one community and extract its social network and related information from the Web. As an example, we infer the social network of an academic society and show that extracted information can be incorporated in FOAF representation. It can then be used to measure the authoritativeness of a member as social trust or individual trust. To demonstrate our Web mining approach in an actual application, we show a researcher mining and retrieval system. Finally, we discuss how the Web mining approach contributes to user aspects in the Semantic Web.

The remainder of this paper is organized as follows: section 2 describes the proposed Web mining method and its application. Section 3 presents discussion of the Web mining approach for user aspects in the Semantic Web. Section 4 shows a comparison of our method with related works. Finally, we conclude this paper in section 5.

2 Web Mining Approach for the Semantic Web

This study specifically addresses one community and obtains the social network and related information from the Web. One reason for focusing on a community is that we believe that a huge “Web of Trust” over the entire Web comprises the superposed local “Webs of Trust” in each community to which a person or an organization belongs to.

Numerous communities exist in the physical world and online. We specifically examine an academic society: Japanese Society of Artificial Intelligence (JSAI). We choose JSAI because of its inherent availability of related information on the Web. Information related to this academic society in computer science is available online to a great degree. Another reason is that we are actually working mainly in JSAI so we can evaluate the extracted information. The following sections show how to automatically obtain JSAI members’ social networks and related information from the Web.

2.1 Social Network Extraction

Before extracting the social network, we choose the participants to the last four annual JSAI conferences as active members of the JSAI community. Each active member of JSAI is represented as a node in a social network. A node is labeled with the name of its corresponding person.

Next, edges between nodes are added using Web information. A simple approach to measure the relevance of two nodes is to use word co-occurrence information. Herein, we define co-occurrence of two words as word appearance in the same Web page. If two words co-occur in many pages, it is assumed that those two have a strong relation. The co-occurrence information is acquired by the number of retrieved documents of a search engine result. For example, assume we are to measure the relevance of two names “Junichiro Mori”(denoted x) and “Yutaka Matsuo” (denoted y). We first address two names $n1$, $n2$ as a query “$n1$ and $n2$” to a search engine and get $|N1 \cap N2|$ documents including those words in the text. Therein, N denotes a Web page set that includes a name n. Additionally, we make another query “$n1$ or $n2$” and obtain $|N1 \cup N2|$ matched documents. The relevance between $n1$ and $n2$ is approximated by the Jaccard coefficient $|N1 \cap N2|/|N1 \cup N2|$. If $n1$ and $n2$ have a strong relation, the retrieved documents might include $n1$’s and $n2$’s homepages, their publication pages, a laboratory’s member list page, a conference program page and so on. In that case, $|N1 \cap N2|$ becomes large compared to $|N1 \cup N2|$. However, the Jaccard coefficient generally gives a famous person few edges because the denominator $|N1 \cup N2|$ is very large in comparison to $|N1 \cap N2|$. We can modify denominator $|N1 \cup N2|$ to $min(|N1|, |N2|)$, which places too much weight on a person with few edges. Therefore, the relevance of node $n1$ and $n2$ is represented by the following threshold-based Simpson coefficient:

$$R(n1, n2) = \begin{cases} \frac{|N1 \cap N2|}{min(|N1|,|N2|)} & if|N1| > k \; and \; |N2| > k, \\ 0 & otherwise \end{cases}$$

We set $k = 30$ for JSAI case. If we wish to estimate the co-occurrence more precisely to a person with small hits, we can pursue other alternatives to calculate statistical reliability. If relevance $R(n1, n2)$ of a node pair is larger than the given threshold, an edge is added with its weight equal to the relevance.

In the same manner as with the edge relation extraction, we can extract information of each node by considering the co-occurrence between the name and the term. For example, the search result of a query “Tim Berners-Lee and Semantic Web” returns about 76500 documents while about 9850 documents are returned for the query “Tim Berners-Lee and Software engineering”. In this manner, we can infer that “Semantic Web” is more relevant to “Tim Berners-Lee” than “Software engineering”. The term set of each node is acquired by retrieving the person’s name that represents the node. Among the set, the term that often co-occurs with a person’s name is chosen as his or her node keyword [2].

It is more useful to assign each edge a “label” for the relationship between two persons. For example, two nodes have the relation of “colleagues of the same research

[2] As a measure of co-occurrence, we use the Jaccard coefficient.

Table 1. Obtained rules.

Class	Rule[3]
Coauthor	SameLine=yes
Lab	(Number_of_Cooccurrence = more_than_one & Word_Group_in_Title(D)=no & Word_Group_in_First_Five_lines(A, E) = yes) or ...
Proj	(SameLine=no & Word_Group_in_Title(A)=no & Word_Group_in_First_Five_lines(F)=yes) or ...
Conf	(Word_Group_in_Title(A)=no & Word_Group_in_First_Five_lines(B)=no & Word_Group_in_First_Five_lines(D)= yes) or ...

Word groups
A: publication, paper, presentation, activity, theme, award, authors etc.
B: member, lab, group, laboratory, institute, team, etc.
C: project, committee
D: workshop, conference, seminar, meeting, sponsor, symposium, etc.
E: association, program, national, journal, session, etc.
F: professor, major, graduate student, lecturer, etc.

Table 2. Higher-ranked keywords of the "Mitsuru Ishizuka" node

Yutaka Matsuo, Hiroshi Dohi, Character Agent, Koichi Hashida, Life-like Interface
Naoaki Okazaki, University of Tokyo, Life-like Agent, Hypothetical Reasoning

institute", "professor-student", "members of the same committee", and so on. We discern the relationship by consulting retrieved page contents and applying classification rules. These rules are obtained through a machine-learning approach. We define labels for each edge as follows: *Coauthor* (Coauthors of a technical paper), *Lab* (Members of the same laboratory or research institute), *Proj* (Members of the same project or committee), *Conf* (Participants of the same conference or workshop). Each edge has multiple labels. For example, the relations might be both *Coauthor* and *Lab*. We first retrieve the top five pages returned for the query "$n1$ and $n2$". Then we extract some features from the contents of each page. We apply classification rules to the features and thereby obtain labels of the relation between $n1$ and $n2$. We employ C4.5 [16] to derive classification rules because of their ease of interpretability. Some of the obtained rules are shown in Table 1: For example, if two names cooccur in the same line, they are classified as coauthors. if the number of cooccurrences is more than one, and the title does not include word group D, but the first five line includes word groups A and E, then the relation is classified as members of the same laboratory.

Figure 1 portrays a part of the social network of the JSAI community. A node is labeled as the corresponding participant name (in Japanese), and an edge is labeled as *Coauthor*, *Lab*, *Proj*, or *Conf*. The whole network is shown in Fig. 2. We have more than 1500 people in the community from which we choose about 150 members to illustrate this network. Table 2 shows higher-ranked keywords of the node – "Mitsuru Ishizuka" – a co-author of this paper and current chairperson of JSAI.

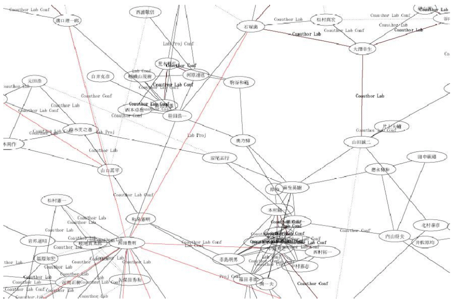


Fig. 1. Part of the JSAI social network

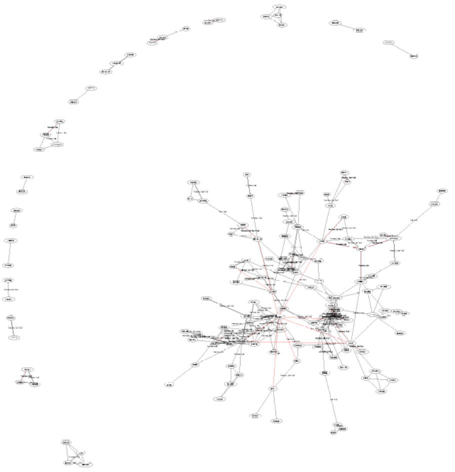


Fig. 2. JSAI social network

2.2 Trust Calculation

Trust on the Social Network Anyone can say anything on the Web. For that reason, lacking trust, we are unable to determine whom to believe. Trust is a necessary condition for users to fully utilize a semantic web.

We focus on the locality of a “Web of Trust”. Initially, a local community will develop a small “Web of Trust” within the community. The small “Web of Trust” in a local community is helpful for judging the reliability of a person, an organization, or a piece of information. Some nodes have a high degree of trust edges: they are considered reliable. A newcomer can gain trust by somehow tying himself to a trusted node. The small “Web of Trust” has its *raison d’etre* within the community. Subsequently, small “Webs of Trust” will appear one by one in different communities. These local “Webs of Trust” will be superposed one by one because a person or an organization belongs to several communities at the same time. Finally, they will come to comprise a huge “Web of Trust” that spans the entire Web, encompassing many local trust networks.

The physical world already offers a “Web of Trust”, as a kind of social network. I trust one of my friends; consequently, I also trust a person introduced by that friend. I trust a company because one of my companies is dealing with that company. In this way, our social network works well to assess trustworthiness. Such a mechanism is likely to work well on the Semantic Web. Using the social network, we can obtain the authoritativeness of a node. It can be considered as reliability or social trust. On the other hand, the network is used to calculate trust that can be accorded to that person: individual trust.

Social and Individual Trust The Google search engine uses a link structure for ranking Web pages, called PageRank [3]. A page has a high rank if the sum of the its for-

Table 3. Result of Authority Propagation

	Name	Activation	Freq	Comment (in 2004)
1	Toyoaki Nishida	5.53	624	Former Commissioner of JSAI, Prof.
2	Toru Ishida	4.98	574	Former Commissioner of JSAI, Prof.
3	Hideyuki Nakashima	4.52	278	Former Commissioner of JSAI, Prof.
4	Koiti Hashida	4.49	345	Commissioner of JSAI
5	Mitsuru Ishizuka	4.24	377	Commissioner of JSAI, Prof.
6	Hiroshi Okuno	3.89	242	Commissioner of JSAI, Prof.
7	Riichiro Mizoguchi	3.60	404	Commissioner of JSAI, Prof.
8	Seiji Yamada	3.35	168	Associate Prof.
9	Hideaki Takeda	3.22	435	Associate Prof.
10	Takahira Yamaguchi	236	624	Prof.
11	Yukio Ohsawa	2.98	185	Associate Prof.
12	Hozumi Tanaka	2.90	465	Chairperson of JSAI, Prof.
13	Takenobu Tokunaga	2.89	302	Associate Prof.
14	Koichi Furukawa	2.77	141	Former Commissioner of JSAI, Prof.
15	Kawahara Tatsuya	2.74	440	Prof.

Table 4. Result of Authority Propagation from Yutaka Matsuo

	Name	Activation	Freq.	Comment (in 2004)
1	Yutaka Matsuo	230.6	136	Target node
2	Mitsuru Ishizuka	28.7	377	Former supervisor, co-author
3	Yukio Ohasawa	19.5	185	Former project leader, co-author
4	Toyoaki Nishida	14.5	624	Professor of lecture at university
5	Masahiro Matsumura	13.5	82	Former colleague, co-author
6	Seiji Yamada	12.7	168	Acquaintance
7	Yasushi Takama	12.3	16	Former researcher of the former laboratory
8	Toru Ishida	12.1	574	Advisory Board of current research center
9	Takahira Yamaguchi	11.5	236	Acquaintance
10	Hidehiko Tanaka	11.3	842	Professor at university

ward links evenly contribute to the ranks of the pages to which they point. PageRank is a global ranking of all Web pages and is known to perform very well.

We employ here a PageRank-like model to measure authoritativeness of each member [13]. Each node v has an authority value $A_n(v)$ on iteration n. The authority value propagates to neighboring nodes in proportion to the node relevance:

$$A_{n+1}(v) = c \sum_{v^{'} \in Neighbor(v)} \frac{R(v, v^{'})}{Rsum(v)} A_n(v^{'}) + cE(v)$$

$$Rsum(v) = \sum_{v^{''} \in Neighbor(v)} R(v, v^{''})$$

where $Neghbor(v)$ represents a set of nodes, each of which is connected to node v, c is a constant for normalization, and E represents a source of authority value. We set E

as uniform over all nodes for simplicity (but it can be set depending on v). If we set a certain node v_{target} as a source of authority value, the result can be interpreted as showing authority for the node: individual trust. We set the initial authority as follows.

$$E(v) = \begin{cases} 1.0 \; if v = v_{target}, \\ 0.0 \; otherwise \end{cases}$$

For mathematical details, see [3].

Table 3 shows a result applied to the JSAI community extracted from the Web. Among 1509 people in the community, these people have high authority value $A(v)$ (denoted as Activation) after 1000 iterations. Although the hits (denoted as Freq) are few, some people are ranked highly. Present or former JSAI Commissioners are 9 of 15 people. Others are younger; they are not yet Commissioners, but they are active researchers who are mainly working in JSAI.

The top listed people by this algorithm are authoritative and reliable in the JSAI community. However, authoritative people are not always listed highly by our approach. For example, JSAI currently has 20 commissioners (including a Chairperson and two Vice-chairpersons), but we can extract only 5 current commissioners of the top 15. In other words, our approach seems to have high precision, but low recall. This drawback is attributable to the lack of information online. Especially, elder authorities tend to have produced many publications before the WWW came to daily use.

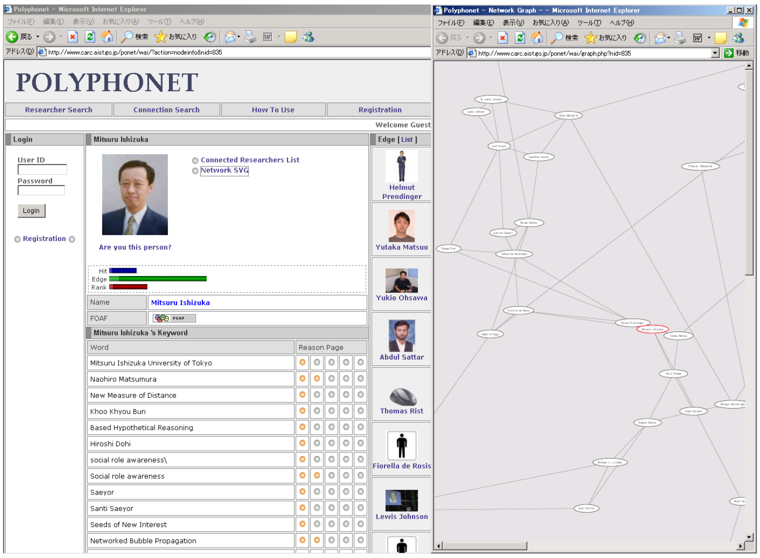
Table 4 shows a result obtained by setting v_{target} as node “Yutaka Matsuo”. The familiar persons for him, e.g., a supervisor, a project leader, colleagues, and co-authors are ranked highly. This ranking is useful as a proxy for individual trust. For example, if a person is judged as very familiar to me, then she can automatically have permission to access my work libraries. Otherwise, she must ask my permission.

2.3 Application

To demonstrate our Web mining approach in the real application, we develop a researcher mining and retrieval system called Polyphonet (Fig. 3). The system is an example of an end-user application that integrates Web mining into the Semantic Web. The system is intended to provide a search function based on the relation of researchers and promote efficient collaboration. For example, a user can find what research topic a researcher is doing or whom she is working with. Social networks is used for finding path to other researchers or recommending related researchers. If the researcher is not found in the system, a user can register his name. Subsequently, the system automatically extracts information from the Web using the proposed Web mining method.

Extracted users’ information is easily incorporated in the RDF representation [11]. For example, the network ties and the interest associations are represented in RDF using the `foaf:knows` and `foaf:interst` properties. Similarly, the relation become `foaf:Persons` with the appropriate relations. Some extensions of the FOAF model are necessary for expressing the relation labels. Figure 4 shows a FOAF file that was generated based on extracted information. Each researcher can have metadata included in the system. because extracted information is stored as a FOAF file.

Trust gives an authoritativeness of a person which is useful when finding an important researcher in the field. If we trace the node which has high individual trust from


Polyphonet – Microsoft Internet Explorer
ファイル(F) 編集(E) 表示(V) お気に入り(A) ツール(T) ヘルプ(H)
戻る 検索 お気に入り
アドレス(D) http://www.carc.aist.go.jp/ponet/wai/?action=nodeinfo&nid=835
POLYPHONET
Researcher Search
Connection Search
How To Use
Registration
Welcome Guest
Login
User ID
Password
Login
Registration
Mitsuru Ishizuka
Connected Researchers List
Network SVG
Are you this person?
Hit
Edge
Rank
Edge [List]
Helmut Prendinger
Yutaka Matsuo
Yukio Ohsawa
Abdul Sattar
Thomas Rist
Fiorella de Rosis
Lewis Johnson
Polyphonet – Network Graph – – Microsoft Internet Explorer
アドレス(D) http://www.carc.aist.go.jp/ponet/wai/graph.php?nid=835
移動

Name	Mitsuru Ishizuka
FOAF	FOAF

Mitsuru Ishizuka 's Keyword

Word	Reason Page				
Mitsuru Ishizuka University of Tokyo	●	○	○	○	○
Naohiro Matsumura	●	●	○	○	○
New Measure of Distance	●	○	○	○	○
Khoo Khyou Bun	●	○	○	○	○
Based Hypothetical Reasoning	●	○	○	○	○
Hiroshi Dohi	●	○	○	○	○
social role awareness\	●	○	○	○	○
Social role awareness	●	●	○	○	○
Saeyor	●	○	○	○	○
Santi Saeyor	●	○	○	○	○
Seeds of New Interest	●	○	○	○	○
Networked Bubble Propagation	●	●	○	○	○

Fig. 3. Polyphonet: a researcher mining and retrieval system

antecedent node, we can find the circle of trust which comprises the small "Web of Trust" in a community.

3 Discussion

Hereafter, we address some of the workshop issues and discuss how our approach contributes to user aspects in the Semantic Web.

- Which baseline technologies are used and how are they combined?
- What aspects of end user activity does the technique affect?
- Can you describe convincing use-case scenarios demonstrating the power and usefulness of this approach?

Users are coming to accept FOAF and its extensions as something of a standardized ontology for representing user semantics on the Semantic Web. It has been a popular ontology of the Semantic Web. In other words, users are actively disseminating their social information on the Semantic Web. Our approach is to support those user-side trends by reusing the current Web as a source to produce such users' information. We employ various Web mining techniques such as a search engine, statistical word co-occurrence information and machine learning. Our approach assists users in extracting relevant information from the Web and integrating it with the Semantic Web technologies. Furthermore, it encourages users to publish their information on the Semantic Web. In the proposed researcher mining and retrieval system, novice users can naturally approach the Semantic Web technologies such as Ontology and "Web of Trust" because those technologies are included in the system.

```
<rdf:RDF
xmlns:rdf=”http://www.w3.org/1999/02/22-rdf-syntax-ns#”
xmlns:foaf=”http://xmlns.com/foaf/0.1”
xmlns:acsn=”http://www.carc.aist.go.jp/ y.matsuo/acsn/0.1”>
<foaf:Person>
<foaf:mbox rdf:resource=”ishizuka@miv.t.u-tokyo.ac.jp”/>
<foaf:name>Mitsuru Ishizuka</foaf:name>
<foaf:interest rdfs:label=”Character agent”
rdf:resource=”http://www.miv.t.u-tokyo.ac.jp”/>
<foaf:currentProject rdfs:label =”Life-like interface”
rdf:resource=”http://www.miv.t.u-tokyo.ac.jp”/>
<foaf:workplaceHomepage rdfs:label=”University of Tokyo”
rdf:resource=”http://www.miv.t.u-tokyo.ac.jp”/>
<acsn:Coauthor>
<foaf:Person>
<foaf:mbox rdf:resource=”y.matsuo@aist.go.jp”/>
<foaf:name>Yutaka Matsuo</foaf:name>
</foaf:Person>
</acsn:Coauthor>
</foaf:Person>
```

Fig. 4. An example of a FOAF file tha tis based on extracted information from the Web.

– What is its potential to improve/simplify users’ tasks?

There is often discussion about how metadata annotation is facilitated and accelerated. Consequently, users often find it difficult to collect and describe their information according to the Semantic Web standards. Reusing the existing sources of information on the Web would be a solution of the semantic annotation problem by minimizing the associated effort and helping users create their metadata.

In the Semantic Web, it is important to know whom to believe so that users can determine whether or not the source of information is reliable and credible. However, users often find it difficult to determine whom to believe in the distributed and heterogeneous environment of the Semantic Web. Our community-based approach would provide important clues for a Web of Trust on such a Semantic Web. Based on such a trust network, the system can help users determine the veracity of trustworthy persons, resources, and information.

– Why do we need the Semantic Web for this?

In the process of reusing the current Web as a source of information to produce semantically rich information, the Semantic Web provides a rich framework to describe semantics of the extracted information. In addition to the FOAF ontology and its extensions that we are currently using, we are extracting myriad community information in different contexts from the Web and converting it into semantic information. Our future work will explore the kinds of service that can be provided using semantically rich information as a resource.

4 Related works

The emerging field of social network mining provides methods for discovering social interactions and networks from legacy sources such as e-mail archives [1, 17], schedule data, Web citation information [15], and FOAF files [6]. It would be useful to incorporate such other information sources to obtain a more accurate social network, but such resources involve particular concerns of privacy: people do not want e-mail data to be analyzed.

Kautz and Selman developed a social network extraction system from the Web, called *referral web* [9]. This pioneering work particularly emphasizes co-occurrence of names on Web pages using a search engine. Mika pursued a similar approach [14] to extract a social network of a community. He also proposed a method to determine whether or not a certain person is associated with a certain interest. Both studies employ the Jaccard coefficient as a co-occurrence index. Although the fundamental idea resembles that of our approach, we further develop the mining algorithm. We use an overlap coefficient rather than a Jaccard coefficient based on experimental evaluation. We apply text processing and machine learning to determine the class of relation. Whereas Mika gives a list of interests, we can capture the various aspects of personal information from different Web pages. Furthermore, our method demonstrates the applicability of calculating the trust of each person.

Golbeck proposed an algorithm for generating locally-calculated trust ratings from a FOAF-based social network [10]. In a peer to peer context, Kamvar developed the EigenTrust system [8], which computes global trust values for peers. Although both approaches calculate trust on the network, we extract a social network of a community from the Web, which realizes more end-users and real-world oriented design for a “Web of Trust”. Many research issues require investigation to realize a “Web of Trust” on the Semantic Web.

5 Conclusions

This paper presents an advanced Web mining approach to extract users’ social networks and their related information from the current Web for the Semantic Web. In particular, we focus on an academic community and then argue the manner in which local trust networks will finally constitute a huge “Web of Trust”. We show that the social relation is utilized to measure the authoritativeness of a member as social trust or individual trust. As an actual application that integrates Web mining with the Semantic Web, we presented a researcher mining and retrieval system.

We target researchers because of their associated information has relatively high availability on the Web, but our approach is not limited to that domain by any means. More and more information related to ordinary people online makes our approach feasible in various domains. More possibilities for using a search engine and mining the “non semantic” Web will arise in the future. For example, an ontology can be constructed using a search engine. We believe that merging the vast amount of information on the current Web and producing semantic information might help users fully utilize a Semantic Web and contribute to its further diffusion.

References

1. L. Adamic and E. Adar. Friends and Neighbors on the Web, http://www.parc.xerox.com/istl/groups/iea/ papers/web10/, 2001.
2. T. Berners-Lee, J. Hender and O. Lassila. The Semantic Web. Scientific American, 2001.
3. S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine, *In Proc. 7th WWW conf*, 1998.
4. D. Brickley and L. Miller. FOAF: the 'friend of a friend' vocabulary. http://xmlns.com/foaf/0.1/, 2004.
5. I. Davis and E. Vitiello Jr. RELATIONSHIP: A vocabulary for describing relationships between people, http://vocab.org/relationship/, 2004.
6. L. Ding, L. Zhou, T. Finin and A. Joshi. How the Semantic Web Is Being Used: An Analysis of FOAF Documents. *In Proc. of the 38th Ann. Hawaii International Conference on System Sciences*, 2005.
7. L. C. Freeman. Centrality in social networks: Conceptual clarification, *Social Networks*, Vol.1, pp.215–239, 1979.
8. S. D. Kamvar, S. T. Mario and H. Garcia-Molina. The EigenTrust Algorithm for Reputation Management in P2P Networks, *In Proc. WWW 2003*, 2003.
9. H. Kautz, B. Selman and M. Shah. The Hidden Web. *AI Magazine*, Vol. 18, No. 2, pp.27–36, 1997.
10. J. Golbeck, J. Hendler and B. Parsia. Trust networks on the semantic web, *In Proc. WWW 2003*, 2003.
11. J. Mori, Y. Matsuo, M. Ishizuka, and B. Faltings. Keyword Extraction from the Web for FOAF Metadata. *In Proceedings of the 1st Workshop on Friend of a Friend, Social Networking and Semantic Web*, 2004.
12. Y. Matsuo, M. Hamasaki, J. Mori, H. Takeda, and K. Hasida. Ontological Consideration on Human Relationship Vocabulary for FOAF. *In Proceedings of the 1st Workshop on Friend of a Friend, Social Networking and Semantic Web*, 2004.
13. Y. Matsuo, H. Tomobe, K. Hasida, and M. Ishizuka. Finding Social Network for Trust Calculation, *In Proc. 16th European Conf. on Artificial Intelligence (ECAI2004)*, 2004.
14. P. Mika, Bootstrapping the FOAF-Web: an experiment in social networking network mining, *Proc. of 1st Workshop on Friend of a Friend, Social Networking and the Semantic Web*, 2004.
15. T. Miki, S. Nomura and T. Ishida. Semantic Web link analysis to discover social relationship in academic communities. *In Proc. 2005 International Symposium on Applications and the Internet (SAINT)*, 2005.
16. J. R. Quinlan, *C4,5: Programs for Machine Learning*, Morgan Kaufmann, California, 1993.
17. J. Tyler, D. Wikinson and B. Huberman. Email as spectroscopy: automated discovery of a community structure within organizations. *Communities and Technologies*, Kluwer, B.V. pp.81–96, 2003.
18. Resource Description Framework(RDF) Schema Specification. In *W3C Recommendation*, 2000.
19. OWL Web Ontology Language. In *W3C Recommendation*, 2004.
20. Representing vCard Objects in RDF/XML. http://www.w3.org/TR/2001/NOTE-vcard-rdf-20010222/