International Workshop on

Adaptation and Personalization for Web 2.0

AP WEB 2.0

http://ailab.dimi.uniud.it/en/events/2009/ap-web20/

This volume is published by Antonina Dattolo, Carlo Tasso, Rosta Farzan, Styliani Kleanthous, David Bueno Vallejo, and Julita Vassileva.

ISSN 1613-0073

Proceedings of the International Workshop on Adaptation and Personalization for Web 2.0 (AP-WEB 2.0 – 2009), Trento, Italy, June 22, 2009.
Edited by: Antonina Dattolo, Carlo Tasso, Rosta Farzan, Styliani Kleanthous, David Bueno Vallejo, and Julita Vassileva
Submitted by: Antonina Dattolo and Rosta Farzan

Published on CEUR-WS:
ONLINE: http://CEUR-WS.org/Vol-485/
ARCHIVE:
ftp://SunSITE.Informatik.RWTH-Aachen.DE/pub/publications/CEUR-WS/Vol-485.zip

AP-WEB 2.0 site:
http://ailab.dimi.uniud.it/en/events/2009/ap-web20/

Table of Contents

Preface …………………….. I-V

Full Papers

A Framework for Flexible User Profile Mashups……………………………................. 1-10
Fabian Abel, Dominikus Heckmann, Eelco Herder, Jan Hidders, Geert-Jan Houben, Daniel Krause, Erwin Leonardi, and Kees van der Slujis

Handling Users Local Contexts in Web 2.0 …………………………………………..… 11-20
Mohanad Al-Jabari, Michael Mrissa, and Philippe Thiran

Context-Aware Notification Management in an Integrated Collaborative Environment 21-30
Liliana Ardissono, Gianni Bosio, Anna Goy, and Giovanna Petrone

A General Framework for Personalized Text Classification and Annotation………..… 31-39
Andrea Baruzzo, Antonina Dattolo, Nirmala Pudota, and Carlo Tasso

A Personalized Tag-Based Recommendation in Social Web Systems ……………........ 40-49
Frederico Durao, and Peter Dolog

Using Asynchronous Client-Side User Monitoring to Enhance User Modeling in Adaptive E-Learning Systems ……………………………………………………………………….. 50-59
David Hauger

Customized Edit Interfaces for Wikis via Semantic Annotations ……………………… 60-68
Angelo Di Iorio, Silvia Duca, Alberto Musetti, Silvia Righini, Davide Rossi, and Fabio Vitali

Visualising web server logs for a Web 1.0 audience using Web 2.0 technologies …..… 69-78
Ed de Quincey, Patty Kostkova, and David Farrell

New Generation of Social Networks Based on Semantic Web Technologies ………….. 79-87
Liana Razmerita, Martynas Jusevicius, and Rokas Firantas

Balanced Recommenders: A hybrid approach to improve and extend the functionality of traditional Recommenders ……………………………………………………………… 88-98
Javier G. Recuenco, and David Bueno Visualizing

Reciprocal and Non-Reciprocal Relationships in an Online Community…………….…. 99-118
Kadhambari Sankaranarayanan, and Julita Vassileva

A User-Centric Authentication and Privacy Control Mechanism for User Model Interoperability in Social Networking Sites ……………………………………………………………110-119
Yuan Wang, and Julita Vassileva

Experiences from Implementing Collaborative Filtering in a Web 2.0 Application …… 120-129
Wolfgang Woerndl, Johannes Helminger, and Vivian Prinz

Short Papers

A Collaborative System Based on Reputation for Wide-Scale Public Participation …… 130-134
Ana Fernández and Jens Hardings

An Approach towards the Exploration of User Preference Adaptation for Cross Device, Cross Context Video Content Recommenders in Web 2.0 Environments …………………… 135-138
Kevin Mercer

MyHealthEducator: Personalization in the Age of Health 2.0 ………………………… 139-142
Luis Fernandez-Luque

New Tagging Paradigms for Content Recommendation in Web 2.0 Portals …………... 143-147
Andreas Nauerz, Fedor Bakalov, Martin Welsch, and Birgitta Konig-Ries

Personal Navigation in Semantic Wikis ……………………………………………... 148-151
Diego Torres, Alicia Diaz, Hala Skaf-Molli, and Pascal Molli

Preface

Adaptation and Personalization for Web 2.0

Antonina Dattolo[1], Carlo Tasso[1]
Rosta Farzan[2], Styliani Kleanthous[3], David Bueno Vallejo[4], Julita Vassileva[5]

[1]Artificial Intelligence Lab, Department of Mathematics and Computer Science, University of Udine, Italy.
[2] Personalized Adaptive Web Systems Lab, University of Pittsburgh, USA
[3] University of Leeds, UK.
[4] Department of Computer Science, Universidad de Málaga, Spain
[5] Department of Computer Science, University of Saskatchewan, Canada
carlo.tasso@dimi.uniud.it; antonina.dattolo@uniud.it;
rostaf@gmail.com; stellak@comp.leeds.ac.uk; bueno@lcc.uma.es; jiv@cs.usask.ca

Abstract. AP-WEB 2.0, the International Workshop on Adaptation and Personalization for Web 2.0, held in Trento in connection to the first and seventeenth international conference on User Modeling, Adaptation and Personalization, UMAP 2009, aimed at discussing the challenges and approaches in adaptation and personalization for Web 2.0. Here we present an overview of the workshop. Thirteen full papers and five short papers were accepted, covering both theoretical and practical aspects of Personalization for Web 2.0. The papers discuss a wide range of areas including user awareness, recommender systems, user-generated content, and social networks.

Workshop Website: http://ailab.dimi.uniud.it/en/events/2009/ap-web20/

Keywords: User Modeling, Personalization, Web 2.0, Recommender Systems, Social Navigation, Knowledge sharing, Folksonomies and tagging, Social Networks, User Awareness.

Web 2.0 users generate a meaningful part of Web contents and traffic: they collaborate, connect, create, share, tag, remix, upload, and download new or existing resources in an architecture of participation, where user contribution and interaction add value. Web 2.0 is growing daily, both in terms of users and applications. Nevertheless, the effective use of adaptation and personalization methodologies within social systems is still an open challenge and current systems rarely go beyond user-driven customization.

User data is shared frequently in social network applications. Facebook, for example, is evolving as a platform that provides user identity and allows third party applications to share user data. A large variety of other social platforms including

MySpace, Hi5, Bebo, Ning, LinkedIn, Plaxo, Six Apart, Orkut, and Yahoo, use open standards such as OpenSocial, and OpenID to allow user identity to be established, third party applications to be added by users, and user data to be shared. Especially when using open standards, this sharing can happen in a decentralized way, by exchanging partial profiles for a purpose, in line with the decentralized/ubiquitous user modeling paradigm.

General open issues include understanding what (truly) adaptive personalized services can empower user interaction and information access and what kind of adaptation can be performed starting from the rich amount and variety of information available about Web 2.0 users, groups, and communities. Limiting the focus of attention to textual information, most of user generated contents are unstructured, expressed in natural language, and this raises other significant open questions: what viable natural language and text analysis techniques are adequate for the above tasks; what knowledge can be automatically extracted from the analysis of collective behavior, and how this knowledge can be exploited for personalization.

Another related key concept is social navigation, based on folksonomies and social tagging: how can tags be exploited for building user interests profiles and personal conceptual spaces; how different user perspectives coming from different tagging approaches can be consistently merged in order to improve social search and navigation; can 'personal' ontologies be derived from user tags and later be exploited for recommending tags in a personalized way to the user and what benefits could come from such an automatic tagging.

How can Web 2.0 application developers attract and sustain active user participation: various approaches have been proposed, varying from explicit incentive mechanisms to community visualizations, displaying user participation levels in order to encourage to improve the user's reputation and to trigger social behavior patterns that would benefit the community. User modeling and adaptation to individual and contextual factors are essential in the design of such incentive mechanisms and community visualizations (open group user models).

Finally, blogs, wikis, and forum systems are (one of) the main highway of user generated contents: how can they be analyzed in order to identify different personalized views and how can these be adaptively exploited for reducing information overload, can information extraction from user generated content be personalized, how standard social network analysis can be improved and innovated by means of content-based and adaptive approaches, are just a few open issues to be explored.

Three specific questions motivated this workshop:

1. How adaptation and personalization methodologies can augment Web 2.0 environments? And how can social adaptation mechanisms be evaluated?
2. What models, techniques, and tools are the most adequate to better support Web 2.0 users?

3. How much the introduction of tools for structuring personal user spaces (currently flat) can improve the creation and navigation processes and social awareness?

This workshop has received 21 submissions of which 13 were accepted as full papers and 5 as short papers. The accepted papers explore a wide range of themes, summarized in the following three macro-areas:

User Awareness: Wang and Vassileva discuss user awareness for reuse and integration by presenting personalized privacy control mechanisms for mashups transparent to the users, and Abel et al. provide a model for the effective interchange and mashup of user profiles. In the context of e-collaboration environments, Ardissono et al. propose a notification management model for supporting the selective deferral of context-aware notifications on the basis of the user's focus of attention, while Al-Jabari et al. analyse several use cases in order to improve the context interpretation of Web contents.

Recommender systems: This theme is treated focusing on integrating collaborative filtering recommender systems into existing Web 2.0 applications (Woerndl et al.), utilizing tags (Durao and Dolog), content (Nauerz et al.) and video content (Mercer) to improve recommendation, recommending personalized tag and content annotation (Baruzzo et al.), applying hybrid approaches and using different sources of data (Recuenco and Bueno).

User-generated content an social networks: Here the focus is on issues regarding the visualization of user interpersonal relationships in a social site (Sankaranarayanan and Vassileva) and of user visit patterns on a Web 1.0 website to acquire richer data about users (Quincey et al.); on the use of Web 2.0 technologies to acquire detailed data about user browsing (Hauger); on the social data portability and object-centered sociality supported by ontologies (Firantas et al), on authoring, tagging and form structuring (Di Iorio et al.), and navigation (Torres et al.); on reputation mechanisms for representing the qualification of user-generated data (Fernández and Hardings).. Finally personalization is discussed in (Luque) in the Health 2.0 domain.

We wish to express our sincere thanks to all the authors who submitted papers, the members of the Program Committee, who reviewed them on the basis of originality, technical quality, and presentation, and the numerous participants.

Organizing Committee

Carlo Tasso	University of Udine, Italy.
Antonina Dattolo	University of Udine, Italy.
Rosta Farzan	University of Pittsburgh, USA.
Styliani Kleanthous	University of Leeds, UK.
David Bueno Vallejo	Universidad de Málaga, Spain
Julita Vassileva	University of Saskatchewan, Canada

Program Committee - Promoters

Peter Brusilovsky	University of Pittsburgh, USA
Joan DiMicco	IBM T.J. Watson Research Cambridge
Vania Dimitrova	University of Leeds, UK
Jon Dron	Athabasca University, Alberta, Canada
Susan Gauch	University of Arkansas, USA
Werner Geyer	IBM T.J. Watson Research Cambridge
Alessandro Micarelli	University "Roma Tre", Roma, Italy
Cecile Paris	CSIRO ICT Centre, Sydney, Australia
Carlo Strapparava	FBK-irst, Italy

A Framework for Flexible User Profile Mashups

Fabian Abel[1], Dominikus Heckmann[2], Eelco Herder[1], Jan Hidders[3], Geert-Jan Houben[3], Daniel Krause[1], Erwin Leonardi[3], Kees van der Slujis[4]

[1] L3S Research Center, Leibniz University Hannover, Germany
{abel,herder,krause}@l3s.de
[2] DFKI GmbH, Saarbrücken, Germany heckmann@dfki.de
[3] Web Information Systems, TU Delft, The Netherlands
{a.j.h.hidders,g.j.p.m.houben,e.leonardi}@tudelft.nl
[4] Department of Computer Science, Eindhoven University of Technology, The Netherlands k.a.m.sluijs@tue.nl

Abstract. Exploiting the rich traces of users' Web interaction promises to enable cross-application user modeling techniques, which is in particular interesting for applications that have a small user population or that are used infrequently. In this paper we present a framework for the effective interchange of user profiles. In addition to derivation rules for user profile reasoning, the framework employs flexible mash-ups of RSS-based user data streams for combining heterogeneous user data in a Web 2.0 environment.

1 Introduction

With the increased use of search engines, e-commerce systems and social networking sites – with famous examples such as Amazon, Facebook, Flickr, Delicious and Google – user modeling and Web personalization has evolved from a rather marginal activity to a mature technology that is exposed to the majority of Web users on a daily basis. Most techniques are based on collaborative filtering and social network analysis [1]. What they have in common is that they are rather straightforward and depend on a sufficiently large number of users that regularly interact with the system [2].

Apart from the major players in the field, many systems cannot boast on a large user base. These systems vary from startups to well-established sites that serve a specialized audience. As an example, e-learning systems inherently have a limited audience, in particular if the system is specifically used by one institution. For these stakeholders, it would be beneficial to have user profile information from other applications. Recent research suggests that, if carefully designed and tested, heterogeneous types of data can be used for reliably classifying users [3, **?**]. Other motivations for cross-application user modeling include the synchronization of recommendations and user interaction between applications and better support of user migration.

Obviously, the idea of cross-application user modeling is not new. In the 1990s several *generic user modeling servers* have been developed, to be used by a wide range of applications (for example [4]). One of the major reasons that this approach has never been successful is that these servers were centralized, making use of predefined structures. By contrast, user models differ significantly between applications, depending on the adaptation goals, the context of use, privacy concerns, the design philosophy and many other factors.

New trends from the Web 2.0 as well as the related work, as will be discussed in Section 2, motivate an infrastructure for cross-application user modeling. This infrastructure, which we introduce in Section 3, is heavily inspired by social networking approaches and is based on the assumption that adaptive systems (or rather the system administrators) themselves are the ones who know best what the system needs. The infrastructure relies on the brokerage of user models, with system administrators searching, discussing, adopting, rating and recommending third parties' user models. Section 4 outlines how to use the framework to reason on distributed user profiles and demonstrates how user profiles can be mashed-up by combining RSS feeds in so-called *user pipes*.

2 Background and Related Work

As described in Tim O'Reilly's Web 2.0 design patterns [5], small sites with a small user population and specific demands make up the bulk of the Web 2.0 domain. Whereas the exchange of login credentials is already facilitated by initiatives such as OpenID[1], still in most cases users need to build their user profiles from scratch for every application. A recent trend is the combination of functionality from multiple Web 2.0 applications in so-called *mashups*. For mashups, the ability to share user profiles is particularly essential for a better integration and cooperation between the single applications.

For the exchange and interpretation of user profile data, common semantics user profile statements are needed [6]. Possible formats for user profiles include the General User Model Ontology (GUMO) [7] or Friend of a Friend (FOAF) [8]. However, as we have seen in the introduction of this paper, these kinds of predefined and static user profile ontologies do not sufficiently cater for the diverse needs of applications. Therefore, we argue that these types of shared models should rather be built bottom-up, starting from successful implementations in specific systems [9].

As a further development, we can see a shift from author-predefined adaptation rules to collaborative filtering techniques and the use of Web 2.0 interaction mechanisms [10]. With a huge pool of data, many candidate user groups to compare the user with, and several methods at hand, it becomes even more important to experiment with and optimize the conceptual adaptation decisions [11].

In essence, there are two ways to ensure interoperability between two adaptive systems and their user models. The first approach involves a *lingua franca*, an agreement between all parties on a common representation and semantics [12]. As described in the introduction, this is the philosophy underlying the generic user model server approach, used by CUMULATE [13] or PersonIs [14]. Given the wide variety in system objectives and the associated user models, generic

[1] http://openid.net/

user model servers have never gained wide acceptance. An alternative approach, which is more flexible, involves *conversion* between the different systems' user models.

Conversion allows for flexible and extensible user models, and for systems to join into a platform. Moreover, in contrast to a fixed lingua franca approach, conversion is suitable for 'open-world user modeling', which is not restricted to one specific set of systems [15]. This flexibility comes at a price, though. In addition to possibly losing information in the conversion process, it might be that models are simple incompatible (in the sense that there is no suitable mapping) or that mappings are incomplete (information required in one model is not available in the other). Given that there are suitable mappings, the observations in the different systems may lead to contradictions [15]. Several methods for conflict detection and resolution are conceivable, among others reliability weighting and majority voting - again, which method to use, may be a subjective design decision.

As pointed out by [16], computer-based representation of *provenance data* is crucial for users who want to analyze, reason, and decide whether or not they trust electronic data. In the article, the generic concept of p-statements is explained: each statement should contain a track record of the input data, the processing and a description of the output data. With this information, a derivation record can be built for analysis purposes. The DCMI Metadata Terms [17] is a collection of properties and classes together with vocabulary and syntax encoding schemes that can be applied to describe the provenance of data as well. The DCMI terms allow to describe metadata of things, such as the creator, time of creation, copyright and modifications.

3 A Framework for User Modeling 2.0

Results from the preceding section provide support for the exchange of user models between applications. From the related work we have seen that incorporating user profile information from other contexts is not a straightforward process, though. The poor take-up of the generic user modeling servers, developed in the 1990s, suggests that a centralized approach, with predefined ontologies, does not cater the needs of the multitude of adaptive systems, which are very heterogeneous in nature.

Based on the above, we designed a framework that facilitates the *brokerage* of user profile information and user model representations. This framework, which we call the Grapple User Modeling Framework (GUMF), is designed to meet the following requirements. First, various types of systems should be able to connect to the framework. Further, the framework should provide a *flexible user model format* that allows for new types of statements and derivation rules. Sufficient *metadata* should be given to indicate its origin, contents and validity. The browsing and searching of user data or model extensions, provided by the connected systems, should be supported by *rating mechanisms*. As several systems may provide *competing* models of, for example, user interests, and as the quality of these models can vary significantly it is important that a system

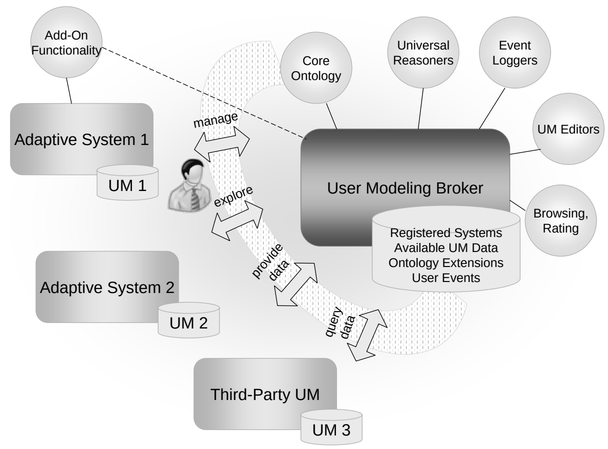


Fig. 1. Generic overview of the functionality of the User Modeling Framework.

administrator (i.e. a user of the framework) can take a motivated decision which alternative is most suitable for his personalization purposes.

The core element of the framework can be considered a *broker*, which provides the means for other systems to share and make use of their user data. In this section we provide an overview of the elements that are needed for setting up this framework.

3.1 Architecture

In Figure 1, a generic overview of the GUMF architecture is depicted. The central element of the framework is the Grapple User Modeling Broker (GUMB), which manages the communication between the connected systems. The broker keeps track of the registered systems, the available user model data and ontology extensions. Further, it keeps a centralized repository of user events. The framework provides Web-based administrative interfaces for *managing* the system configuration and for *exploring* the available user data streams, reasoning mechanisms and ontology extensions. The target audience of these interfaces consists of the administrators and programmers of client (adaptive) systems, in order to find and incorporate suitable user data streams and to offer their own data streams. For most mapping, merging and reasoning tasks, administrators can utilize generic reasoning plugins (cf. Section 4) and hence generate user profile data in a format that perfectly fit their applications' needs. For more specific reasoning tasks, administrator can create own reasoning plugins an provide them to the GUMF community. Once configured, the client systems can exchange user data without human intervention. The *provision* of data takes place in the form of statements, of which the structure is explained in more detail in Section 3.2.

The *querying* of user data – summarized in statements – is realized through three alternative interfaces. The RESTful interface provides a light-weight querying approach for retrieving statements that match a certain simple pattern. A more elaborate interface is provided by a SOAP interface, which is more flexible,

property	description
ID	The globally unique ID of the statement.
type	In the current version of the UM ontology we differentiate between *gc:Statement*, which is a basic user profile statement, and *gc:Observation*, which is a specialization of *gc:Statement* and models a (user) observation made in some application.
subject	The entity (usually the user) the statement is about.
predicate	Refers to a property (of a domain ontology) that either characterizes the *subject* (e.g. *foaf:interest* or *k:hasKnowledge*) or describes some action the *subject* has performed (e.g. *nop:hasBookmarked* or *nop:hasClicked*).
object	The value of the *predicate* (e.g. *"ItalianFood"* or *dbpedia:semantic_web*).
created	Specifies when the statement was created.
creator	Refers to the entity that created the statement. In case of a *gc:Observation* it identifies the entity that reported the observation.
temporal	Allows to define temporal constraints on the validity of the statement.
evidence	If a statement was produced by a reasoning process then *evidence* can be used to show how the statement was deduced.
rating	The *rating* of a statement indicates the level of trust in the statement.

Table 1. Important properties of a Grapple statement as defined in the Grapple User Modeling Ontology (see: http://www.kbs.uni-hannover.de/gumf.owl).

at the cost of a more complicated syntax and communication costs. A third interface allows applications to subscribe to an RSS-based data stream that matches a query, to be notified upon changes. The latter interface is particularly useful for event-driven personalization mechanisms, which depend on events in other systems.

The GUMF architecture is inspired by the Personal Reader Framework [18], with as main enhancements the extensible user modeling ontology format, flexible query interfaces and a community-based way of sharing and ranking user models.

3.2 User Modeling Ontology

The Grapple User Modeling Ontology specifies the lingua franca for exchanging user profile information and user observations in a User Modeling 2.0 infrastructure. It follows the approach of the General User Model Ontology [7] (GUMO) and UserRDF [19], as it is built upon the notion of reified *subject-predicate-object* statements. The *subject* models the entity (usually the user) that the statement is about. The *predicate* refers to a property that either characterizes the subject (e.g. *foaf:interest* or *k:hasKnowledge*) or describes some action the subject has performed (e.g. *nop:hasBookmarked* or *nop:hasClicked*). The *object* contains the corresponding value (e.g. *"ItalianFood"* or *dbpedia:semantic_web*). Each statement has a globally unique ID and is enriched with metadata (see Table 1), such as the creation date or details about the provenance of the statement.

```
gc   = http://www.grapple-project.org/grapple-core/
foaf = http://xmlns.com/foaf/0.1/
gc:Statement {
  gc:id:        gc:statement-peter-2009-01-01-3234190;
  gc:user:      http://www.peter.de/foaf.rdf#me;
  gc:predicate: foaf:interest;
  gc:object:    http://en.wikipedia.org/wiki/Italy;
 }
```

In the example above, the subject (gc:user), predicate, and object refer to entities that are not part of the Grapple Core ontology. gc:user identifies the user Peter by referring to his FOAF profile, which is a separate document located at "http://www.peter.de/foaf.rdf". The value of the predicate is "foaf:interest"., which is a property defined in the FOAF ontology [8]. To find out about the actual meaning of "foaf:interest", one has to look up the FOAF ontology[2]:

```
<rdf:Property rdf:about="http://xmlns.com/foaf/0.1/interest"
     vs:term_status="testing"
     rdfs:label="interest"
     rdfs:comment="A page about a topic of interest to this person.">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
  <rdfs:domain rdf:resource="http://xmlns.com/foaf/0.1/Person"/>
  <rdfs:range rdf:resource="http://xmlns.com/foaf/0.1/Document"/>
  <rdfs:isDefinedBy rdf:resource="http://xmlns.com/foaf/0.1/"/>
</rdf:Property>
```

The definition of "foaf:interest" gives us the actual meaning of the Grapple statement. The comment describes the semantics of the predicate, to be read by people that want to use the property. Making use of the definitions of the domain and range, we can deduce that "http://www.peter.de/foaf.rdf#me" is of the type "foaf:Person", that "http://en.wikipedia.org/wiki/Italy" is a "foaf:Document" and that the predicate "foaf:interest" reflects 'A page about a topic of interest to this person'.

4 User Profile Reasoning

The Grapple User Modeling Framework allows to dynamically utilize reasoning plugins to enable user profile reasoning. In this section we present two generic solutions that can be utilized directly by the GUMF client applications: (1) a rather classical rule-based approach and (2) a novel approach, which we call *User Pipes*, that allows user profile reasoning by mashing up different user profile data streams. However, client administrators can also create own reasoning plugins and share them with the community. A user interface within the client administrator backend allows to search for and publish own reasoning plugins.

4.1 Reasoning Plugins

Reasoning plugins are software components that can be integrated into the Grapple User Modeling Framework (GUMF). In general, they deduce new information about a user based on existing user profile data or based on some observations. Reasoning plugins can come in different flavors. For example, a plugin might gather and align user data from different social networking services in order to create a more comprehensive user profile.

The first generic reasoning plugin is rule-based and applies derivation rules, which can be defined and adjusted by client applications. These derivation rules enable GUMF to generate new Grapple statements. Rules allow to express simple

[2] More precisely, the ontology that is identified via foaf = http://xmlns.com/foaf/0.1/

types of inference in terms of premise-conclusion rules that derive new statements from the existence of other statements. These rules can, for example, (i) infer statements that embody new knowledge, (ii) they can be used to map between different ontologies or (iii) they describe how to solve problems where statements or rules conflict with each other. A simple derivation rule that infers new knowledge about a user might express the following: If a user has bookmarked a website that has topic t then the user is interested in t. Such a rule can, for example, simply be formulated as a SPARQL query:

```
PREFIX foaf:  <http://xmlns.com/foaf/0.1/>
PREFIX gc:    <http://www.grapple-project.org/grapple-core/>
PREFIX gnop:   <http://www.grapple-project.org/nop/>

CONSTRUCT { gc:derivedStatement  gc:user      ?user .
            gc:derivedStatement  gc:predicate foaf:interest .
            gc:derivedStatement  gc:object    ?topic }
WHERE {
            ?originalStatement   gc:user      ?user
            ?originalStatement   gc:predicate gnop:hasBookmarked .
            ?originalStatement   gc:object    ?document .
            ?document            foaf:topic   ?topic . }
```

A mapping rule could simply map one value to another value or it can compose a new value from other values or decompose one value in different separate values. Conflict resolution rules can be used to define preferences among different types of statements or preferences among different rules.

4.2 User Pipes

In addition to the rule-based approach described in the section above, GUMF enables deduction of user profiles also by mashing up different (user profile) data streams in RDF or RSS-format by utilizing Semantic Web Pipes[3] or Yahoo Pipes[4]. In this chapter, we focus on the processing of RSS data by utilizing Yahoo pipes as this enables the usage of a huge amount of structured data on the web. Different RSS streams are syndicated to so-called *User Pipes*.

How this works is shown by our GUMF demonstrator[5]. A specific profile stream *searchedFor* of the user *fabian* can be retrieved by requesting */user/fabian/predicate/searchedFor*. An extract of the data stream is given as follows.

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF ...>
  <channel rdf:about="http://semweb.kbs.uni-hannover.de:8082/grapple-umf/user/fabian">
    <title>GUMF data stream matching the query 'user = fabian'</title>
    <link>http://semweb.kbs.uni-hannover.de:8082/grapple-umf/user/fabian</link>
    <items>
      <rdf:Seq>
        <rdf:li rdf:resource="http://semweb.kbs.uni-hannover.de:8082/grapple-umf/62715"/>
        <rdf:li rdf:resource="http://semweb.kbs.uni-hannover.de:8082/grapple-umf/63526"/>
        ...
      </rdf:Seq>
    </items>
  </channel>
```

[3] http://pipes.deri.org/
[4] http://pipes.yahoo.com
[5] Available at http://semweb.kbs.uni-hannover.de:8082/grapple-umf/

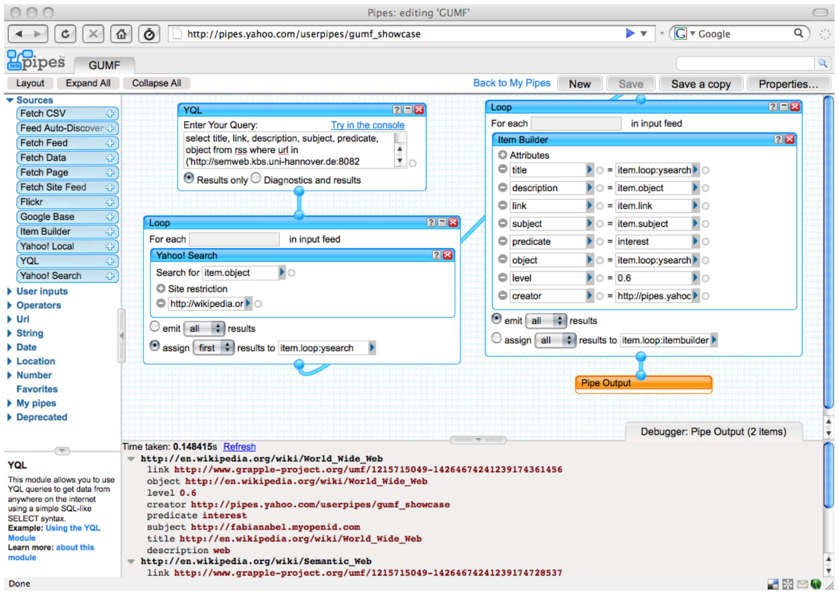


Fig. 2. User Pipe: Mashing up user profile data streams from different sources (here: GUMF search activity stream and Delicious bookmarks). Online available at: http://pipes.yahoo.com/userpipes/gumf_showcase

```
  <item rdf:about="http://semweb.kbs.uni-hannover.de:8082/grapple-umf/62715">
    <title>user 'fabian' searched for 'Trento'</title>
    <link>http://semweb.kbs.uni-hannover.de:8082/grapple-umf/62715</link>
    <gc:subject>http://fabian.myopenid.com</gc:subject>
    <gc:predicate>http://www.grapple-project.org/nop.owl#searchedFor</gc:predicate>
    <gc:object>Trento</gc:object>
    <gc:level>1.0</gc:level>
    <gc:created>2009-03-20T18:23:50Z</gc:created>
    <gc:creator>http://bookstore.example.org</gc:creator>
    ...
  </item>
  ...
</rdf:RDF>
```

This data stream can be combined with other data streams to deduce new user profile information. For example, it can be combined with information from the feed */user/fabian/predicate/interest* to deduce whether the user's interests and search activities are thematically similar or it can even be mashed up with other RSS feeds from the Web.

To demonstrate how meaningful streams can be created by embedding profile data from social networking sites, we created a simple user pipe[6] that combines the search activity stream listed above with the latest bookmarks that the user created at Delicious[7]. Figure 2 shows the editor view of the user pipe. The given user pipe detects those keywords that a user applied for both search and tagging of his latest bookmarks, which is expressed via the following YQL query.

```
SELECT title, link, description, subject, predicate, object FROM rss WHERE url in
```

[6] Available at http://pipes.yahoo.com/userpipes/gumf_showcase
[7] http://feeds.delicious.com/v2/rss/fabianabel

```
  ('http://semweb.kbs.uni-hannover.de:8082/grapple-umf/user/fabian/predicate/searchedFor')
 AND object in
  (select category from rss where url in ('http://feeds.delicious.com/v2/rss/fabianabel') )
```

The result of the YQL query is then passed to a component that tries to map the detected keywords to Wikipedia articles that further explain the concepts that are referred by the keywords. In the last stage, an *Item Builder* component is used to generate new Grapple statements. Similar to the example in Section 3.2, the above item makes use of the FOAF vocabulary (*foaf:interest*) to express that the user is interested in *http://en.wikipedia.org/wiki/Trento* (cf. bottom of Fig. 2):

```
<?xml version="1.0" encoding="UTF-8"?>
  ...
  <item rdf:about="http://www.grapple-project.org/umf/1215715049-1426467424123917436145 6">
    <title>http://en.wikipedia.org/wiki/Trento</title>
    <gc:subject>http://fabian.myopenid.com</gc:subject>
    <gc:predicate>http://xmlns.com/foaf/0.1/interest</gc:predicate>
    <gc:object>http://en.wikipedia.org/wiki/Trento</gc:object>
    <gc:creator>http://pipes.yahoo.com/userpipes/gumf_showcase</gc:creator>
  </item>
  ...
```

The benefit of the user pipe approach is that user pipes result in user profile streams that can again be used by other profile reasoners, which allows for flexible and extensible user profile reasoning. For publicly available data streams it is also possible to directly use the Yahoo Pipe editor, which provides an easy drag-and-drop user interface to process, combine, and perform various operations on data streams. This means that not only programmers or experts familiar with SPARQL or rule-based languages are enabled to create profile reasoners, but also leisure user as they can create such reasoners (user pipes) visually.

The critical point of this approach is the immensely huge amount of RSS data on the Web that could slow down the processing of a pipe. Therefore, we are going to explore caching strategies (e.g. the precompute pipes regulary and deliever the cached results) as proposed in [13] and will conduct performance measures as well.

5 Conclusions and Future Work

In this paper we motivated and introduced a framework for cross-application user modeling. Based on several pieces of earlier work, the framework provides a domain-independent, decentralized approach for combining several user models. In a collaborative manner, the connected systems can create, share, select, mashup, adopt and rate their user models, supported by a basic infrastructure that includes search and browse facilities, editors and universal reasoning mechanisms.

Although the framework provides the basic infrastructure for cross-application modeling, its success depends on the take-up by a critical mass and the availability of the necessary tools. In the GRAPPLE project, we are currently integrating the framework, to be used by a number of different e-learning systems. By evaluation and experimentation, we expect to find additional requirements and success factors for building an ecology of adaptive systems that exchange parts of their user models.

Acknowledgements The work presented in this paper has been sponsored by the EU FP7 STREP Project GRAPPLE.

References

1. Frias-Martinez, E., Magoulas, G., Chen, S., Macredie, R.: Modeling human behavior in user-adaptive systems: Recent advances using soft computing techniques. Expert Systems with Applications **29** (2005) 320–229
2. Bilenko, M., White, R.: Mining the search trails of surfing crowds: Identifying relevant websites from user activity. In: Proc. WWW2008. (2008)
3. Korth, A., Plumbaum, T.: A framework for ubiquitous user modeling. In: Proc. Information Reuse and Integration 2007. (2007)
4. Kobsa, A., Koenemann, J., Pohl, W.: Personalized hypermedia presentation techniques for improving customer relationships. The Knowledge Engineering Review **16 (2)** (2001) 111–155
5. Oreilly, T.: What is web 2.0: Design patterns and business models for the next generation of software. Social Science Research Network Working Paper Series (2007)
6. Ankolekar, A., Krötzscha, M., Trana, T., Vrandecic, D.: The two cultures: Mashing up web 2.0 and the semantic web. In: Web Semantics: Science, Services and Agents on the World Wide Web. (2008)
7. Heckmann, D., Schwartz, T., Brandherm, B., Schmitz, M., von Wilamowitz-Moellendorff, M.: Gumo - the general user model ontology. In: Proc. of the 10th Int. Conf. on User Modeling, Edinburgh, UK (2005) 428–432
8. Brickley, D., Miller, L.: FOAF Vocabulary Specification 0.91. Namespace document, FOAF Project (November 2007) http://xmlns.com/foaf/0.1/.
9. Paramythis, A., Loidl-Reisinger, S.: Adaptive learning environments and e-learning standards. Electronic J. of e-Learning **2 (2)** (2004) Paper 11
10. Chatti, M.A., Jarke, M.: The future of e-learning: a shift to knowledge networking and social software. Int. J. Knowledge and Learning **3 (4/5)** (2007) 404–420
11. Nielsen, J.: Personalization is over-rated. Alertbox (October 1998)
12. Stewart, C., Celik, I., Cristea, A., Ashman, H.: Interoperability between aeh user models. In: Proc. APS 2006. (2006)
13. Yudelson, M., Brusilovsky, P., Zadorozhny, V.: A user modeling server for contemporary adaptive hypermedia: An evaluation of the push approach to evidence propagation. In: 11th International Conference on User Modeling. (2007) 27–36
14. Assad, M., Carmichael, D., Kay, J., Kummerfeld, B.: Personisad: Distributed, active, scrutable model framework for context-aware services. (2007) 55–72
15. Aroyo, L., Dolog, P., Houben, G., Kravcik, M., Naeve, A., Nilsson, M., Wild, F.: Interoperability in pesonalized adaptive learning. J. Educational Technology & Society **9 (2)** (2006) 4–18
16. Moreau, L., Groth, P., Miles, S., Vazques-Salceda, J., Ibbotson, J., Jiang, S., Munroe, S., Rana, O., Schreiber, A., Tan, V., Varga, L.: The provenance of electronic data. Communications of the ACM **51 (4)** (2008)
17. DCMI Usage Board: DCMI Metadata Terms. DCMI Recommendation, Dublin Core Metadata Initiative (January 2008)
18. Henze, N., Krause, D.: Personalized access to web services in the semantic web. In: SWUI 2006 - 3rd International Semantic Web User Interaction Workshop, Athens, Georgia, USA (nov 2006)
19. Abel, F., Henze, N., Krause, D., Plapper, D.: User modeling and user profile exchange for semantic web applications. In: 16th Workshop on Adaptivity and User Modeling in Interactive Systems, Wuerzburg, Germany. (2008)

Handling Users Local Contexts in Web 2.0: Use Cases and Challenges

Mohanad Al-Jabari[1]*, Michael Mrissa[2], and Philippe Thiran[1]

[1] PReCISE Research Center, University of Namur, Belgium
[2] SOC Research Team, LIRIS, University of Lyon 1, France

Abstract. Creating, updating, and aggregating Web contents from different Web users and sites form the heart idea of Web 2.0. However, Web users originate from different communities, and follow their own semantics (referred to as local contexts in this paper) to represent and interpret Web contents. Therefore, several discrepancies could rise up between the semantics of Web authors and readers. In this paper, we present several Web 2.0 use cases, and illustrate the possible challenges and trends to handle the local contexts of Web users in these use cases.

1 Introduction

During the last years, the emergence of the Web 2.0 has revolutionized the way information is designed and accessed over the internet. The term *Web 2.0* was officially coined by Tim O'Reilly in [11] as a set of design principles and exemplified by sites such as *Wikipedia*[3], *MySpace*[4], *Upcoming*[5], *etc.* However, several researchers including Tim O'Reilly himself argue that there is no clear-cut definition of this term [6, 2, 3].

The heart idea of Web 2.0, in addition of using Web technologies as a platform, lies into the sharing of Web contents from different sources. *Community collaborations and contents mashups* are the most common Web 2.0 features [3]. To illustrate these features, let us distinguish them from the classical Web (called "Web 1.0") features as follows:

- **Community collaboration**. In Web 1.0, a few *Web authors* create and update Web contents for relatively passive *Web readers*. However, Web 2.0 sites enable Web users not only to browse the Web but also to create, update, and share Web contents in usually self-organizing manner. Hence, Web users now can act as active *Web authors*.
- **Contents mashups**. In Web 1.0, Web contents (information and services) on a single web page are usually belong to one Web site. In Web 2.0, contents from several sites can be aggregated, mixed, and displayed together.

* Supported in part by the Programme for Palestinian European Academic Cooperation in Education (PEACE).
[3] http://wikipedia.org
[4] http://www.myspace.com
[5] http://upcoming.yahoo.com

The emerging results of community collaboration and contents mashups could not be achieved by individual users and individual Web sites, respectively. Each user gains more from the systems than he puts into it. Also, one Web site can not satisfy all the users' needs. Contents from different sites are to be aggregated and mixed together to satisfy complex users' requests.

1.1 Users Local Contexts

The Web gathers billions of Web users from all over the world. These users originate from different communities, and follow their local contexts for interacting with Web contents. By *local context*, we means a set of common knowledge such as a common language and common cultural conventions such as measure units, keyboard configurations, character sets, notational standards of writing times, dates, numbers, currency [14, 5].

Since different communities usually have different local contexts, a same concept (a Web concept) could be represented differently by different *Web authors*. Also, the same Web content (the representation of a Web concept) could be interpreted in different ways by different *Web readers*. Hence, several discrepancies could be arisen between the semantics of Web authors and readers. For example, assume a French reader who wants to interpret a *price* Web content which is authored by a British author. In this context, the price is represented in British Pound and follows the British currency format (e.g., 1,234.50). As the French currency is Euro and different format is used (e.g., 1 234,50), the price must be converted from British Pound to French Euro by the reader. Note that the situation can be even worse if the reader wants to interpret a *date* content. The reader could misinterpret the date content (e.g., 07/08/2008) as the 7^{th} of August 2008 (following the French format) instead of the 8^{th} of July 2008 (following the British format). Similar situations may occur with other pieces of Web contents that are related to users' local contexts.

1.2 Web 2.0 and Users Local Contexts

The emergence of the Web 2.0 raises new challenges. Web contents in a single page can be authored (created and updated) by several authors who have different local contexts. Moreover, contents authored from several authors on several Web sites could be dynamically aggregated, mixed, and displayed together in a single Web page. This paper presents several possible Web 2.0 use cases and explores some possible challenges and trends for handling users' local contexts in these use cases.

This paper is organized as follows. Section 2 presents several Web 2.0 use cases. Section 3 introduces a set of concepts that could be represented and interpreted according to users' local contexts and the challenges of handling them in the Web 2.0 use cases. Section 4 introduces semantic annotation as a possible solution. Finally, Section 5 concludes the paper.

2 Web 2.0 Use Cases

In this section, we describe several possible use cases that users could perform when they use Web 2.0 systems. By no means we aim at covering all Web 2.0 use cases, but we attempt to classify the aforementioned Web 2.0 features (i.e., community collaboration and contents mashups) into three use cases: Web contents creation, Web contents update, and Web contents aggregation.

2.1 Web Contents Creation

Several Web 2.0 systems enable Web authors to create Web contents, without giving the opportunities to update the published contents or parts of them. We focus on the Web contents creation in this use case. To illustrate this, let us consider the following Web 2.0 services:

- *Weblog (also called blog)*. Web 2.0 systems such as *WordPress*[6] allow a single author to create Web contents (e.g., scientific articles,privacy issues, etc.) called *post*, whereas other secondary users can add comments to contents created by the original author as new html nodes.
- *Bulletins Section*. Web 2.0 social systems such as *Facebook*[7] and *MySpace* provide a service to a group of users called "bulletin board". Bulletin board allows a user to add a piece of Web content (e.g., text message), whereas other users on the group list can see this content. Bulletins can be useful to contact an entire friends list without resorting to messaging users individually.
- *Group Section*. Social systems also provide a service called "group section". One or more users can create a common page (i.e., group section). The group creator(s) can invite any one to join, deny user's join request, delete or update users' contents, etc. Joined users, in addition to the group creator(s), usually can browse and create contents on the group section.

2.2 Web Contents Update

Several Web 2.0 systems enable Web authors to update Web contents after publishing. In this use case, a Web author could update the Web contents that she/he creates (referred to as a personal contents update) or the Web contents that other Web authors create (referred to as a community contents update). The following Web 2.0 services illustrate this use case:

- *Personal contents update*. Web 2.0 commerce systems such as *eBay*[8] allow a Web user to update the contents about the items she/he wants to sell. A user can update the contents concerning these items like the price, the photos, the selling location, etc. Other users can not change these pieces of contents. In addition, social systems allow a user to update his own profile such as login name and password, preferred language, interests, etc.

---

[6] http://wordpress.org/
[7] http://www.facebook.com/
[8] Available on http://www.ebay.com/.

- *Community contents update.* Wiki systems such as *Wekipedia* allow one or more users (usually authorized users) to create Web contents as a set of interlinked Web pages and update these contents using creating and editing services. For example, a Web user can define the term *local context* or update the existing definition authored from other author(s). In addition, collaborative editing systems such as *Google Docs* allow a group of users (might be from different locations) to collaboratively create and update documents (e.g., word document) online. Finally, the group creator(s) of the group section presented above can update the contents created by joined users.

2.3 Web Contents Aggregation

Several Web 2.0 systems and technologies provide Web contents aggregation and mixing services. In this sense, the aggregation and mixing services could be performed on client-side (referred to as a client-side aggregation) or on a specific server-side application (referred to as a server-side aggregation). The following Web 2.0 services illustrate this use case:

- *Client-side aggregation.* RSS feed reader (aggregator) is the most known technology that allows client-side applications (e.g., Web browser) to find out and collect Web contents from RSS-enabled Web sites[9]. In addition, *Piggy bank* [7] and *Kalpana* [4] provide client-side aggregation services. These services aim at enabling Web readers to extract and aggregate personal information from different Web sites, and to store them locally in RDF formats.
- *Server-side aggregation.* Several Web 2.0 systems mix Web contents from different sites. For example, Google provides an advertisement service called *adSense*[10] which enables Web site to add text, image, or video advertisement from other Web sites. In addition, several Web 2.0 systems provide aggregation services for specific types of Web contents. For example, *Technorati*[11] aggregates and indexes different types of contents such Weblogs, photos, news, DVDs, etc. Also, Technorati allows readers to search these contents in different ways (e.g., readers can search Weblogs according to Weblogs' langauge). *Upcomming* is another system that aggregates events from users communities and commercial sites. Users can indicates their plans by marking that they are "going" to or "interested" in events that are occurred in a location, date, future periods, etc. Also, users can choose which events who are interested in such as education, music, sports, etc.

  Finally, several E-commerce systems compose Web services together (e.g., airplane ticket reservation, car rental reservation, and hotel reservation) from different service providers (i.e., Web sites) to satisfy a complex user request. In this sense, we can assume these systems as server-side aggregators.

[9] Any website that offers RSS feeds for its content.
[10] http://www.google.com/adsense
[11] http://technorati.com/

3 Web 2.0 Use Cases and Users Local Contexts

As we mentioned, several discrepancies could be arisen between the semantics of Web authors and readers, since they could have different local contexts. In this section, we initially presents a set of concepts that could be represented and interpreted according to users' local contexts. Then, we discuss the challenges of handling the local contexts of these concepts in the above Web 2.0 use cases.

3.1 Context-Sensitive Web Concepts

Based on local context, we aim at classifying Web concepts into context[12]-sensitive and non-context-sensitive concepts. *Context-sensitive concepts* refer to the concepts which could be represented in different ways by different authors. The following list identifies a set of context-sensitive concepts. By no means we claim that this list covers all context-sensitive concepts, but we try to address the main concerns that are rose up in the aforementioned use cases [10, 9].

- **Date/time**. Date refers to a particular day of a month or a year within a calendar system (e.g., Gregorian, Islamic, Japanese, etc.). In addition, different communities represent Date in different ways. The day, month, and year are ordered differently, and different separators are used. Also, text representation of Date depends on user's local language and country. Finally, Time could be represented in 12-hour AM/PM or 24-hour style, and with different time zone.
- **Number**. In mathematics, Numbers are mainly used for counting and measuring amounts or quantities of objects based on a number system. Different local symbols are used to represent numbers (also called numerals such as English and arabic numerals[13]). Also, different decimal and thousands separators (i.e., dot and comma) are used in different countries.
- **Price**. Price refers to a numerical monetary value assigned to a good, service or asset. Prices are expressed in different formats, currencies[14], and Tax systems (Tax rates, included/excluded, etc.).
- **Physical quantities**. Physical quantities such as weight, length, temperature, etc. are measured using a set of units called measure units. Countries are used different measure systems (mainly Imperial and Metric systems), different unit prefixes, and different error percentage[15].
- **Telephone number** refers to a unique sequence of numbers used to identify a telephone endpoint. Based on ITU[16] numbering plan $E.164$, each country has a different *international call prefix* and *country calling code*. Furthermore, each country uses a specific telephone number's format.

[12] Context here refers to the local context.
[13] See numeral systems on http://en.wikipedia.org/wiki/Numeral system
[14] See ISO 4217 for used currency list.
[15] More information available on http://en.wikipedia.org/wiki/Units of measure
[16] International Telecommunication Union: http://www.itu.int/

3.2 Challenges of Handling Users Local Contexts

We can conclude that the local context represents a part of the semantic for the above Web concepts. Also, the semantic discrepancies that could arise do not relate to these concepts themselves, but rather to the local contexts of Web authors and readers that are implicity used when they represent and interpret these concepts.

In order to address this issue, several approaches have been proposed to adapt Web contents to be suitable to readers' local contexts [12, 14, 8]. These approaches are mostly based on two assumptions: (1) the semantics of target Web contents to be adapted are known in advance; (2) Web contents are represented according to a single local context.

However, the use cases presented above illustrate that these assumptions are not valid anymore. Web contents are shared (created, updated, and aggregated) from different sources (i.e., Web users and Web sites). Hence, they are represented according to different local contexts and have heterogenous semantics. Therefore, the following challenging issues should be tackled:

1. *Semantic identification.* What is the information that required to identify the semantics of Web contents and the local contexts of Web users?
2. *Semantic information management.* How can the contents' semantics and the users' local contexts information be managed in terms of acquiring, representing, and storing this information? Also, what is the local context that used for representing each piece of Web content?

Semantic Identification

As mentioned before, Web contents could be created, updated, and aggregated from different sources. In this sense, different Web contents from different sources could refer to the same context-sensitive concept. For example, different authors could use *cost*, *price*, and *amount* contents to refer to the *price* concept. In addition, the value of the price concept could be represented in different ways, according to the authors' contexts.

Moreover, Web contents can be *stored, aggregated and hosted* on the server-side and can be *aggregated and presented* on the client-side. Server-side and client-side applications can not interpret Web contents if they are represented only using XHTML. Hence, a server-side application can not be aware if Web contents such as *cost*, *price*, and *amount* refer to the *price* concept or not, and it can not know which local context was used for representing them.

In this sense, several questions could be raised here. Firstly, what is the information that required to identify the semantics of Web contents, so that server-side and/or client-side applications can interpret that Web contents from different sources refer to one context-sensitive concept? Secondly, what is the (minimum) information required to identify the users' (authors and readers) local contexts, so that server-side and/or client-side applications can adapt Web contents from authors' contexts to readers' contexts. One could argue that the local context depends on users' countries, which can be obtained from the IP address contained in the HTTP header. However, this assumption is not valid as one country can have several communities (e.g., Belgium). Another question: how can we identify the local contexts of cross-sites aggregated contents?

Semantic Information Management

In addition to the aforementioned issues, the information required to identify the semantics of Web contents and the local contexts of Web users needs to be acquired, represented, and stored.

In this sense, several questions have to been tackled. Firstly, how can the required information be acquired from different sources (i.e., users and sites). Assume the Web contents creation and update use cases. Does the required information be acquired directly from the authors or be acquired (predicted) from the server-side applications? Also, when the required information be acquired? (i.e., before contents creation or update, during contents creation or update). Assume the Web contents aggregation use case. How can this information be acquired from different sites.

Secondly, how should the required information be represented and where it should be stored (i.e., on the server-side or on the client-side), so that the local contexts of context-sensitive concepts can be handled in the above use cases. For example, the required information should be accessible from the client-side applications in order to handle the client-side aggregated contents. Also, it should be accessible from the server-side applications in order to handle the server-side aggregated contents.

Finally, how to specify the local context that used for representing each piece of Web content? Assume the community update use case where one Web author can update the contents created by other authors (e.g., Wiki contents). The question here: are the updated contents related to the context of the original author or the context(s) of the author(s) who update these contents? Moreover, assume, in contents aggregation use case, the case where the authors' local contexts for parts of the aggregated contents are not specified. How can this case be handled?

4 Possible Solution

One possible solution to handle the aforementioned challenges is to directly rely on the authors for annotating Web contents with semantic metadata, so that the former become machine interpretable [15]. Semantic metadata are used to describe contents' semantics and users' local contexts explicitly. In this sense, Client-side and server-side applications can interpret a Web content (e.g., cost) that is related to a specific context-sensitive concept (e.g., *price*). Also, they can interpret that this content is represented according to a specific local context. Therefore, Web contents can be adapted from authors' local contexts to different readers' local contexts.

In addition, semantic metadata are accessible from server-side and client-side applications, as they are combined with Web contents. In the content aggregation use case, Server-side and/or client-side applications aggregate Web contents together with the corresponding semantic metadata. Finally, the Web authors, in contents update use case, should update Web contents and also the corresponding semantic metadata.

In this field, there are two alternative approaches. The first approach aims at standardizing the representation of Web contents and their semantics for all sources. For example, representing the Date/Time concepts according to the ISO 8601 specification[17]. The second approach aims at allowing authors to represent Web contents in different ways, but explicitly annotate them with semantic metadata (i.e., contents' semantics and authors' local contexts). Microformats technology[18] follows the first approach and RDFa[19] technology follows the second one [13, 10].

4.1 Microformats

Microformats propose a set of standards, or *specifications*, and reuse XHTML attributes such as *id* and *class* to embed those specifications into XHTML documents. For example, the *hCard* specification identifies vocabularies based on the vCard[20] specification that provide semantic information about people and organization. Microformats specifications standardize the representation of Web contents and their semantics at different three levels as follows:

- *Schema level*. Identifying a specific schema for each Microformats specification in terms of concepts and sub-concepts (called classes and subclasses) that can appear and their cardinalities (e.g., required, optional, etc.), the ordering of schema classes, etc. For example, hCard should have *vcard* class, *fn* and *n* subclasses at minimum.
- *Concept level*. Identifying a specific semantic vocabulary (Semantic label) for every class and subclass in each Microformats specification. Therefore, standardizing contents' semantics.
- *Representation level*. Identifying a specific representation for each class's and subclass's values. The authors should follow these representations as much as possible, so that Microformats parsers can interpret these representations. Therefore, standardizing authors' local contexts.

Server-side and/or client-side applications can interpret Web contents annotated with Microformats (i.e., exchange, aggregate, adapt, etc.) without significant loss of meanings. However, Microformats are not extensible and do not fulfill all authors' use cases. In our previous work, we conclude that Microformats remain rather limited as they propose a finite set of specifications [10]. Technically, Web authors can create new specifications, but it is not recommended without extensive discussion with the Microformats community for a general (i.e.

17 http://en.wikipedia.org/wiki/ISO 8601
18 More information available on http://microformats.org/
19 More information available on RDFa wiki: http://rdfa.info/wiki/RDFa.
20 More information available on http://www.isi.edu/in-notes/rfc2426.txt

worldwide) adoption. Until this point is reached, Microformats parsers could not interpret what are considered as "exotic" Microformats specifications.

4.2 RDFa

RDFa provides a more abstract solution that aims at expressing RDF statements in XHTML documents. More precisely, RDFa provides a collection of XHTML attributes (reuses existing attributes such as *content* and *rel* and introduces new ones such as *about* and *property*) to embed RDF statements in XHTML, whereas it provides processing rules for extracting RDF statements from XHTML.

Web authors can reuse existing RDF-based semantic metadata (e.g., Dublin Core and FOAF metadata) and create their own semantic metadata. Therefore, RDFa is fully extensible. However, since Web contents and semantic metadata from different sources are represented in different ways; the interpretation of these contents (i.e., exchange, aggregation, adaptation, etc.) require a prior semantic reconciliation between server-side and client-side applications [3].

In [1], we propose an approach that uses RDFa to annotate context-sensitive concepts with authors' local contexts, so that these concepts can be adapted into different readers' local contexts.

5 Conclusion

The main strength of the Web lies in its capacity to interconnect billions of users from all around the world. However, this gathering of communities can lead to the misunderstanding of Web contents as each community of users uses its own context for interacting with Web contents. In this paper, we identified new challenges in improving the context interpretation of Web contents in some typical Web 2.0 use cases. We also explained how existing technologies such as RDFa and Microformats can help people to better understand each other on the Web. Based on [1], our future work aims at providing an intuitive way for helping authors to annotate context-sensitive concepts with contextual attributes.

References

1. M. Al-Jabari, M. Mrissa, and P. Thiran. Towards web usability: Providing web contents according to the readers contexts. In *Proceedings of the First and Seventeenth International Conference on User Modeling, Adaptation, and Personalization (UMAP'09)*, Lecture Notes in Computer Science, 2009.
2. S. Amer-Yahia, V. Markl, A. Halevy, A. Doan, G. Alonso, D. Kossmann, and G. Weikum. Databases and web 2.0 panel at vldb 2007. *SIGMOD Rec.*, 37(1):49–52, 2008.
3. A. Ankolekar, M. Krötzsch, T. Tran, and D. Vrandecic. The two cultures: Mashing up web 2.0 and the semantic web. *J. Web Sem.*, 6(1):70–75, 2008.
4. A. Ankolekar and D. Vrandecic. Kalpana - enabling client-side web personalization. In *Hypertext*. ACM, 2008.

5. W. Barber and A. Badre. Culturability: The merging of culture and usability. In *the 4th Conference on Human Factors and the Web*, 1998.
6. G. Cormode and B. Krishnamurthy. Key differences between web 1.0 and web 2.0. *First Monday*, 13(6), 2008.
7. D. Huynh, S. Mazzocchi, and D. R. Karger. Piggy bank: Experience the semantic web inside your web browser. *J. Web Sem.*, 5(1):16–27, 2007.
8. P. S. (Innsbruck). Website localization and translation. In H. G.-A. S. S. N. (Saarbrcken), editor, *MuTra 2005 - EU-High-Level Scientific Conference:Challenges of Multidimensional Translation*, May 2005.
9. T. Jevsikova. Localization and internationalization of web-based learning environment. In R. Mittermeir, editor, *ISSEP*, volume 4226 of *Lecture Notes in Computer Science*, pages 310–318. Springer, 2006.
10. M. Mrissa, M. Al-Jabari, and P. Thiran. Using microformats to personalize web experience. In *Proceedings of the 7th International Workshop on Web-Oriented Software Technologies (IWWOST08)*, 2008.
11. T. O'Reilly. What is web 2.0? design patterns and business models for the next generation of software. 2005.
12. K. Reinecke, G. Reif, and A. Bernstein. Cultural user modeling with cumo: An approach to overcome the personalization bootstrapping problem. In *Workshop on Cultural Heritage Systems in the Semantic Web 2007*, Lecture Notes in Computer Science, 2007.
13. E. Torres. Open data in html. XTECH CONFERENCE, 2007.
14. O. D. Troyer and S. Casteleyn. Designing localized web sites. In *WISE*, volume 3306 of *Lecture Notes in Computer Science*, pages 547–558. Springer, 2004.
15. Ü. Yoldas and G. Nagypál. Ontology supported automatic generation of high-quality semantic metadata. In R. Meersman and Z. Tari, editors, *OTM Conferences (1)*, volume 4275 of *Lecture Notes in Computer Science*, pages 791–806. Springer, 2006.

Context-Aware Notification Management in an Integrated Collaborative Environment

Liliana Ardissono, Gianni Bosio, Anna Goy, and Giovanna Petrone

Dipartimento di Informatica, Università di Torino, Italy
{liliana,bosio,goy,giovanna}@di.unito.it,
WWW home page: http://www.di.unito.it/Anagrafica

Abstract. The e-collaboration tools available in open environments offer services supporting the interaction and the synchronization between users, who are typically involved in parallel activity spheres; e.g., different projects and home schedules. However, such tools provide the user's with separate views on her activities and collaborations; moreover, they support workspace awareness by delivering unstructured notification streams, which challenge the user's attention and cannot be filtered or grouped on a relevance basis.
As an answer to this issue, we present an Integrated Collaborative Environment offering a unified view of the user's parallel collaboration spheres. This environment includes a notification management model supporting the selective deferral of notifications on the basis of the user's focus of attention.

1 Introduction

With the large availability of wireless connectivity, broad band internet connections and mobile devices, people result being on-line most of the time. This Web presence offers opportunities to manage different spheres of activity, concerning both work collaborations, as done in project management and cooperative work (e.g., see [1]), and personal commitments related to the users' personal life. For instance, as described in [2], working parents increasingly tend to handle life scheduling needs as a "holistic management of personal, family and professional schedules across settings and calendaring tools". Moreover, people exploit various Web applications to share photos, and other kinds of documents, with groups of friends, as well as with larger communities.

Currently, many applications support the synchronous and asynchronous interaction between users; e.g., Instant Messaging, Shared Workspaces, Forums, e-mail, audio and video Web conferences, and similar. However, with the exception of some Project Management vertical tools [3, 4] and domain-specific awareness support tools [5], these applications are mostly available as distinct services which can hardly be integrated in a unified environment supporting the user's activities. Thus, the user is provided with separate views on the state of the collaborations she is involved in. Specifically:

- Each tool separately handles a local definition of the collaboration groups and most tools are unable to import the group definitions from other tools. [1] Thus, the user is forced to manage multiple instance of her collaboration groups.

[1] Unless they are strictly related, such as some Google Apps.

- Each tool presents the state of the user's collaborations concerning the kind of activities it supports; e.g., calendar information, versus document sharing information. Thus, the user is provided with a partial view on the overall set of events concerning her collaboration spheres (e.g., activities, commitments, and tasks to be performed).
- As each tool separately handles its own awareness information, the user is overloaded with a flow of unrelated notifications, which cannot be filtered on a relevance basis, or managed by following a specific notification policy.

In order to get an overview of the events occurred within a specific activity sphere (e.g., those concerning the presence of collaborators, and the scheduling of meetings), the user has therefore to extract the relevant awareness information from the parallel streams of notifications generated by the collaboration tools she exploits. This issue becomes even more relevant if the user uses multiple tools supporting the same collaboration functions, e.g., one at work and a different one at home, as all such information streams have to be fused as well. For instance, [2] reports the difficulties in importing feeds from external calendars (e.g., those used by the children's school to schedule meetings) in the user's working one.

The above discussion highlights the need for collaboration environments that enable users to manage all their spheres of activity, by supporting the integration of external applications, and offering a unified awareness support. The last aspect is particularly important to make awareness information easily accessible and to prevent the user from being overloaded by flows of unrelated notifications.

As an answer to this issue, we developed an open Integrated Collaboration Environment (ICE) supporting e-collaboration in multiple spheres of activity. The ICE is based on the integration of a set of collaboration tools and can be extended with additional applications, in order to provide new collaboration features, or to comply with specific user requirements. Given the set of integrated applications, the ICE manages a unified view of the state of the collaborations the user is involved in and it provides a context-aware delivery of the awareness information to the user. This is achieved by:

- Replacing the subjective view on collaboration groups, which most collaboration tools offer, with a centralized management of the activity spheres and of the associated user groups.
- Enabling the fusion of the awareness information generated by each of the integrated applications. This fusion is based on the introduction of an agent, the *Notification Manager*, that acts as an intermediary between the user and the collaboration environment and generates personalized notifications for the user.

This paper focuses on the provision of awareness information and presents the notification management model developed in our ICE to adapt the awareness support to the user's notification preferences. This model is based on alternative mediation policies [6], which can be selected by the users. In particular, we introduce a selective deferral of notifications based on their relevance to the sphere of activity representing the user's focus of attention.

In the following, Section 2 deals with awareness in e-collaboration environments. Section 3 describes the ICE and the awareness management. Section 4 provides some technical details. Sections 5 and 6 describe the related research and conclude the paper.

2 Workspace awareness and interruption management

2.1 Background

The effects of interruptions on people's activities have been thoroughly studied in the literature: it has been repeatedly noted that an interruption has a disruptive effect on both a user's task performance and emotional state [7–9].

Interruptions are particularly critical in collaboration environments, which base the awareness support on the delivery of notifications to their users. For instance, in Computer Supported Cooperative Work, the notifications of other people's activities support group awareness during synchronous or asynchronous collaboration [11, 12].

A collaborative workplace poses novel issues concerning interruptions: coworkers may be involved in multiple tasks, belonging to different projects [1], therefore multiplying potential interruptions from colleagues or automatic agents. In addition to commonly used e-mail and instant messaging, other software agents such as shared calendars and shared maps can become a source of notifications and, thus, of interruptions.

2.2 Evaluation of notification management policies

The previous discussion suggests that a correct handling of interruptions is critical to achieve a balanced trade-off between interruptions and awareness. In this perspective, we analyzed the impact on users of a set of notification policies providing different filtering criteria for the organization and presentation of the awareness information. Our hypothesis was that the overhead on users might be reduced by mediating the notification delivery. As a collaboration environment can be used to manage parallel activity spheres, each one generating its own awareness information, we hypothesized to filter and defer notifications on a contextual basis. This led us to hypothesize some context-based notification management policies that could be robust enough to significantly reduce the disruptive impact of interruptions in user's work, but also as flexible as to give the user an acceptable level of awareness of her collaborators' activities.

We performed a test with final users (21 participants, 11 males and 10 females) to evaluate the effects of interruptions by notifications in a collaboration environment. Users were divided into three groups of 7 participants each, and each group experimented a different notification policy:

- In the *no filter* situation, all the notifications, from all the projects the user was involved into, were submitted (7 notifications from 3 different spheres, originated from regular users and administrators).
- In the *context filter* situation, only the notifications from the user's focus of attention (i.e. the project she was actually working at) were submitted (3 notifications from the sphere of activity of the user, originated from regular users and administrators).
- In the *priority filter* situation, the notifications of administrators' activities from all projects, plus those included in the context filter, were submitted (4 notifications from 3 different spheres; notifications from administrators were considered with high priority and submitted even if originated from spheres of activity different from the user's focus).

All the filtered notifications, if any, were displayed to the user when she completed her main task, in the form of a single e-mail message.

Users were asked to perform a simple task (alphabetically sorting a list of names) belonging to a shared project, and they were told that this had to be their main focus of attention. They were also instructed that they were involved in two other projects (planning a conference and planning their participation to an English class) and that notifications of other people's activities concerning the same projects could interrupt them.

Notifications pop-up windows were displayed in the low-right corner of the screen. Users were told to behave in the most normal and spontaneous way when reacting to an interruption. For example, they could choose to access their e-mail application to visualize the full text of the e-mail and eventually reply to it or just ignore the notifications, proceed with the primary job and then process all e-mails once their primary job was fully accomplished, according to their personal attitude and current state.

All the interruptions were simulated by the experimenter in a *Wizard of Oz* modality; to be able to monitor a subject's on-screen activity, a RealVNC server was installed on the subject's computer. The experimenter, through a client application, watched the subject's task execution and simulated interrupting events in real time.

At the end of the test, a NASA-TLX survey was submitted to the users in order to evaluate their total subjective workload [13]. We analyzed the difference in the mean workloads between the three groups. The mean workload expressed by users in the no filter situation was particularly high (mean = 57.55 in a scale from 0 to 100). Mean workload did not significantly decrease in the priority filter situation (mean = 48.68, T = 0.8371, p = 0.430), while a significant difference was noted between no filter situation and the context filter situation (mean = 36.96, T = 3.3575, p = 0.012).

The context filter emerged as the best choice for our notification policy. The priority filter (which featured only one more interruption than the context filter) performed particularly bad, especially with users that had no previous experience at working in shared ambients, and was therefore discarded.

3 The Integrated Collaboration Environment

Our prototype ICE supports the coordination of personnel activities (professors, students, etc.) within a University Department. The ICE currently includes a calendar management application, a document sharing tool and a process management component which handles the workflows of two University projects.

3.1 Architecture

The ICE is based on the SynCFr environment for the synchronization of applications [14], which supports the sharing of context information among applications, based on the Publish and Subscribe pattern. The context information includes: (i) business data and (ii) synchronization information (concerning, e.g., the events occurring within the applications). The integration of a software component in the environment is performed by wrapping it with an adapter which addresses interoperability issues and enables the component to subscribe for the relevant context information, and to publish the one it generates; see [14].

Our ICE extends SynCFr with the integration of a set of components supporting the user's collaboration and the management of workspace awareness. In particular:

- The *User Agent* manages the identities and the notification preferences of the users registered in the ICE. Moreover, the agent tracks the sphere of activity in their focus of attention, while they operate within the environment; see Section 4.2.
- The *Group Manager* supports the users in the definition of the spheres of activity and of their associated collaboration groups. The spheres can be private or shared with other registered users. We assume that, within an organization, a set of public spheres is defined to organize projects and other similar activities; moreover, any registered user can create her own spheres, to integrate the management of personal commitments with the workplace ones.
- The *Notification Manager* mediates the delivery of notifications to the user, according to the notification preferences stored in the User Agent.

3.2 Notification management policies

The policies applied in our ICE are aimed at deciding whether a notification should be immediately delivered or it should be deferred. According to the results of the tests described in Section 2.2, the criterion used to steer the deferral of notifications is their relevance to the sphere of activity the user is focusing on. In particular, the ICE offers the following policies, which the user can explicitly select:

- The default policy is the *context filter*.
- The user can however set as her notification preference the *no filter* policy.
- Furthermore, the user can keep the *context filter* as a default, but she can apply the *no filter* policy to one or more specific spheres of activity.

In the management of deferrals, two main factors should be taken into account: on the one hand, as noticed in [10], the burst of user activity on a task typically lasts a short time, after which she can be interrupted with less disruptive effects. Moreover, users should be enabled to select themselves the latency to be applied in the deferral. On the other hand, the users work in multitasking (see, e.g., [15]); in order to support workspace awareness effectively, at each focus shift they should be informed about the deferred notifications concerning the new focus of attention.

Given such requirements, our ICE enables the user to select the maximum amount of time a notification can be deferred. While the user operates in the ICE, she is notified about all the deferred notifications as soon as one of them reaches the deadline. Moreover, the environment delivers all the deferred notifications at each focus shift.

At the actual stage of development, when a set of deferred notifications has to be delivered, it is reported in a single e-mail message, in a format supporting the user in the inspection of a possibly long list of messages. The message is an interactive web page in which the notifications are grouped by sphere of activity. For this purpose, the page contains a set of clickable tabs, one for each sphere, among which the user can switch by means of a click. In order to highlight the notifications concerning the user's focus of attention, the corresponding tab of the page is presented as the front one. Each tab includes a list of message headers, available as links. For each message, the following information is displayed:

– *subject*, including the application that generated the notification and the object;
– *sender*, i.e., the user who originated the notification;
– *date*.

4 Technical details

The management of the notification policies is based on three elements:

1. The association of the events generated by the ICE applications to their reference spheres of activity; see Section 4.1.
2. The recognition of the user's *focus of attention*, i.e., of the sphere of activity she is working at, at any given moment; see Section 4.2.
3. The mediated notification management; see Section 4.3.

4.1 Contextualization of events

For each application integrated in the ICE, the adapter wrapping the component is in charge of tagging the events it generates with the sphere of activities they belong to, or with the list of users involved in the event, depending on the kind of component.

Specifically, if the component explicitly manages contexts (e.g., process management components do that), the adapter can tag such events appropriately. However, most collaboration tools only support the sharing of objects with sets of users; e.g., documents in GoogleDocs. In this case, the wrapper tags the event with the list of users sharing the object. If the same users participates to more than one sphere of activity, the event is implicitly associated to all such spheres, and thus ambiguously tagged.

4.2 Analysis of the user's behavior

From the viewpoint of the notification management, the user's behavior in the ICE is summarized by two context variables:

– The user's *activity status* specifies if the user is active, idle, or off line.[2]
– The *focus of attention* stores the systems's hypotheses on which sphere of activity the user is working at: this is a list of spheres, representing alternative hypotheses, and is empty when the user is off line.

The User Agent associated to a user U receives the context information about U's activities available in the Cross-Application Context and it analyzes such information in order to update her focus of attention (henceforth, F):

[2] This is sensed by the Instant Messaging application embedded in the ICE.

- When the User Agent receives the notification that the user is active, it initializes F with all the user's spheres of activity. Moreover, it sets the history (H) of the discarded hypotheses to the empty list. H is a buffer of discarded hypotheses, which might need to be rescued, given the new evidence about the user's behavior.
- Each time the Agent receives a new piece of context information describing a user action (e.g., she has uploaded a document in the shared space), it analyzes the reference groups (henceforth, RG) of the event.[3] Then, it updates F and H accordingly. We assume that, if there is no evidence of a focus shift, then the user is continuing to work within the same sphere of activity (continuity assumption). Thus, F and H are updated as follows:
  - If $F \cap RG \neq \emptyset$, we hypothesize that the new evidence contributes to restrict the focus of attention. Let F' and H' denote the updated values of F and H, respectively. Then, $F' = F \cap RG$. Moreover, $H' = F \cup RG - F'$. This means that the history is cleaned; then, it is set to the hypotheses just discarded from the focus of attention plus those provided by the new event which were not included in F' because they were not consistent with the continuity assumption (they introduced new focus hypotheses). For instance, suppose that H is $\{G3\}$, that the focus is $\{G1, G2\}$ and that the new event is tagged as $\{G1, G4\}$. Then $F' = \{G1\}$ and $H' = \{G2, G4\}$.
  - If $F \cap RG = \emptyset$, we assume that the user has shifted to a new sphere of activity. Thus, $F' = RG$.

Notice that the ICE components publish events concerning both the actions performed by the user and those triggered within the applications she uses. For instance, GoogleDocs can be polled to retrieve events of type `[Document` X `uploaded by user` Y `at time` T`]` each time a user saves a new copy of a document X, or the document is automatically saved by the application. Thus, the User Agent receives a regular flow of evidence while the user is active in the ICE. When the flow of activities stops, the Agent sets the focus of attention to the empty list.

4.3 Notification Management

The Notification Manager handles the notifications directed to each user registered in the ICE by filtering them according to her preferences. Each time an event concerning the user is published in the Cross-Application Context, the Notification Manager operates as follows:

- If the user is off line, it stores the event in an internal buffer; when the user is on line again, it discards all the events older than 24 hours; then, it merges the other ones (cleaned from redundancies) into a message, structured as described in Section 3.2, and sends the message to the user by e-mail.
- If the user is on line, the notifications are delivered, or deferred, depending on her policy preferences:
  - If the user has selected the *no filter* policy, the Notification Agent notifies the user by generating an Instant Message via the IM application.

[3] This is done by retrieving the spheres of activity to which a list of users belongs.

- Otherwise (*context filter*), it reads the user's focus of attention (*F*), and calculates the intersection between *F* and the tagging information of the event (*RG*). If $F \cap RG \neq \emptyset$, or one of the reference groups in *RG* belongs to the *non filtered* spheres, then the Notification Manager sends the notification to the user; otherwise, it defers it.

If the user's is online but the focus of attention is empty, this means that the user is working outside the ICE. Thus, the Notification Manager defers all the notifications, except for those concerning the *non filtered* spheres.

5 Related Work

The notification management approach presented in this paper is based on the mediated notification management model, which is largely used and has been identified as one of the best performing methods; see [6]. In particular, our approach extends previous work on priority-based notification with the management of parallel notification contexts, representing different priorities for the user. Different from the work in [10], where the notifications are filtered on the basis of their features (e.g., the sender, priority of a message, etc.), we base the management of notifications on the sphere of activity to which they belong. Specifically:

– We introduce a *context filter* policy, which delays the delivery of the notifications belonging to spheres of activity out of the user's focus of attention. This policy implements a context-dependent notion of priority, suitable for the environments supporting the management of parallel activity spheres and multiple collaborations.
– We introduce a context-dependent model for the presentation of the notification to the user, in order to support the inspection of the awareness information concerning the various spheres of activity she is involved in.

This differs from the awareness support offered by e-collaboration environments such as BSCW [16], or MyWebDesktop [17], which support the management of parallel collaboration groups, but only filter the notifications on a subscription basis.

Our work strictly relates with the ecology of collaborations proposed by Mark and Su in [1]. However, our activity spheres are more general than the *working spheres* introduced in [1] and can be used to represent any kind of collaboration the user is involved in: both work and personal ones. Moreover, while Mark and Su focus on whether the user can be interrupted, depending on her working sphere, our work aims at steering the notification management.

6 Conclusion

We have described an Integrated Collaboration Environment (ICE) supporting e-collaboration within multiple spheres of activity; e.g., different projects users are involved in at work, their social activities, and so on. Specifically, we have focused on the notification management issue and we have defined a novel notification policy supporting the context-dependent delivery of messages to the user. We based the definition of our notification policies on the results of a user test carried out within our lab; the test evaluated the *context filter*, which defers notifications on the basis of their relevance to the user's focus of attention, as the policy most suited to the user's needs.

The next step of our work is the evaluation of our ICE prototype with users. In particular, we will focus on the proposed notification management policies, in order to evaluate the impact of the ambiguity in the identification of the focus of attention on the selection of the notifications to be deferred.

Currently, we do not analyze the user's activities in detail, e.g., to identify different phases in the execution of a task; e.g., see [15]. This analysis might be part of our future work, in order to investigate the adaptation of notification deferrals to the user's attention level. In our future work, we will also deal with privacy issues; see [18–20].

References

1. Mark, G.: Considering Web 2.0 technologies within an ecology of collaborations. In: Proc. of SociUM: Adaptation and Personalisation in Social Systems: Groups, Teams, Communities, Corfu, Greece (2007) 50–59
2. Grimes, A., Brush, A.: Life scheduling to support multiple social roles. In: Proc. CHI 2008, Florence, Italy (2008) 821–824
3. Collanos: Products overview: Team enabling professionals, http://www.collanos.com/en/products (2008)
4. ActiveCollab: Project management & collaboration, http://www.activecollab.com/ (2008)
5. Scupelli, P., Kiesler, S., Fussell, S., Chen, C.: Project View IM: a tool for juggling multiple projects and teams. In: Proc. CHI'2005, Portland, Oregon (2005) 1773–1776
6. McFarlane, D.: Comparison of gour primary methods for coordinating the interruption of people in Human-Computer Interaction. Human-Computer Interaction **17**(1) (2002) 63–139
7. Bailey, B., Konstan, J., Carlis, J.: The effects of interruptions on task performance, annoyance, and anxiety in the user interface. In: Proc. INTERACT'01, Tokyo (2001) 593–601
8. McFarlane, D., Latorella, K.: The scope and importance of human interruption in human-computer interaction design. Human-Computer Interaction **17**(1) (2002) 1–61
9. Czerwinski, M., Horvitz, E., Wilhite, S.: A diary study of task switching and interruptions. In: Proc. CHI 2004 Conference on Human Factors in Computing Systems, Vienna (2004)
10. Horvitz, E., Apacible, J., Subramani, M.: Balancing awareness and interruption: Investigation of notification deferral policies. In: Proc. 10th Int. Conf. on User Modeling, Edinburgh, UK (2005) 433–437
11. Dourish, P., Bellotti, V.: Awareness and coordination in shared workspaces. In: Proc. CSCW '92, Toronto, Canada (1992) 107–114
12. Brush, A., Bargeron, D., Grudin, J., Gupta, A.: Notification for shared annotation of digital documents. In: Proc. SIGCHI conference on Human factors in computing systems (CHI '02), Minneapolis, USA (2002) 89–96
13. Hart, S., Stateland, L.: Development of NASA-TLX (Task Load Index): results of empirical and theoretical research. In Hancock, P., Meshkati, N., eds.: Human Mental Workload. North-Holland, Amsterdam (1988) 239–250
14. Ardissono, L., Goy, A., Petrone, G., Segnan, M.: SynCFr: Synchronization collaboration framework. In: Proc. of 5th Conference on Internet and Web Applications and Services (ICIW 2009), Venezia, Italy, IEEE (2009) to appear
15. Iqbal, S., Horvitz, E.: Disruption and recovery of computing tasks: field study, analysis, and directions. In: Proc. of CHI 2007, San Jose, CA (2007) 677–686

16. Horstmann, T., Beltley, R.: Distributed authoring on the Web with the BSCW shared workspace system. StandardView **5**(1) (1997) 9–16
17. MyWebDesktop.net: Your personal and collaborative desktop on the internet, address = https://www.mywebdesktop.net/, year =
18. W3C: Platform for Privacy Preferences (P3P) Project, http://www.w3.org/P3P/ (2006)
19. Kobsa, A., Schreck, J.: Privacy through pseudonymity in user-adaptive systems. ACM Transactions on Internet Technology **3**(2) (2002) 149–183
20. Brar, A., Kay, J.: Privacy and security in ubiquitous personalized applications. In: Proc. 10th International Conference on User Modeling (UM'05) Workshop 9, Edinburgh (2005) 47–54

A General Framework for Personalized Text Classification and Annotation*

Andrea Baruzzo, Antonina Dattolo, Nirmala Pudota, and Carlo Tasso

University of Udine, Via delle Scienze 206, 33100 Udine, Italy
{andrea.baruzzo, antonina.dattolo, nirmala.pudota, carlo.tasso}@dimi.uniud.it

Abstract. The tremendous volume of digital contents available today on the Web and the rapid spread of Web 2.0 sites, blogs and forums have exacerbated the classical information overload problem. Moreover, they have made even worse the challenge of finding new content appropriate to individual needs. In order to alleviate these issues, new approaches and tools are needed to provide *personalized* content recommendations and classification schemata.

This paper presents the PIRATES framework: a Personalized Intelligent Recommender and Annotator TEStbed for text-based content retrieval and classification. Using an integrated set of tools, this framework lets the users experiment, customize, and personalize the way they retrieve, filter, and organize the large amount of information available on the Web. Furthermore, the PIRATES framework undertakes a novel approach that automates typical manual tasks such as content annotation and tagging, by means of personalized tags recommendations and other forms of textual annotations (e.g. key-phrases).

1 Introduction

In the context of Semantic Web and Web 2.0 environments, finding an appropriate content is regarded not only as a problem of information overload but also as a problem of Web personalization [1], which deals with personalizing content retrieval and access with respect to a specific user model. Moreover, this large volume of data makes impractical or even impossible several manual activities such as extracting *small* portions of relevant information from available contents, or classifying contents according to a specific model of user interests [2]. As a consequence, the gap between the performance of traditional information retrieval tools (e.g. search engines) and the user satisfaction in their use continues to grow. In order to alleviate this issue [3], more sophisticated approaches and tools become necessary for providing *personalized* content recommendations and classification. Furthermore, in a world of collaborative publishing we have to take into account e-Learning, knowledge management and Web 2.0 as typical

* The authors acknowledge the financial support of the Italian Ministry of Education, University and Research (MIUR) within the FIRB project number RBIN04M8S8.

application environments. Indeed, we can discover new relevant information by looking the *community* of people that, for example, share a common set of documents or use the same tags to label them. In this wider setting, automatic text classification remains a significant research field with several challenges such as:

– *Associating rich and precise semantics to information contents.* For describing an object, people tend to assign to it a very small number of tags, based on their knowledge background; of consequence, same tags, used by different users, do not share a common semantics [4, 5].
– *Adapting information retrieval strategies to an evolving user model*, providing run-time malleability to end-users [6]. Certainly, continuously updating a user profile is more difficult than building a single static representation, and requires the availability of some forms of user feedback to keep synchronized the model.
– *Finding relationships between contents and using a uniform method to share and reuse tagging data* amongst users or communities [7]. The topicality criteria alone may not be sufficient to relate contents when there is no shared semantics for a tag.

Our main goal in building the PIRATES framework is to empower social bookmarking tools, allowing users to easily add new contents in their personal collection of links, automatically supporting them when categorizing by means of keywords (tags) in a personalized and adaptive way. This work is a first step towards the generation and sharing of personal information spaces described in [8]. We have designed PIRATES keeping in mind several applications where it can provide innovative adaptive tools enhancing user capabilities: in e'learning for supporting the tutor and teacher activities for monitoring (in a personalized fashion) student performance, behavior, and participation; in knowledge management contexts (including for example scholarly publication repositories and digital libraries [9]) for supporting document filtering and classification and for alerting users in a personalized way about new posts or document uploads relevant to their individual interests; in online marketing for monitoring and analyzing the blogosphere where word-of-mouth and viral marketing are nowadays more and more expanding and where consumer opinions can be listen.

The paper is organized as follows: Section 2 illustrates the overall architecture and operation of PIRATES; Section 3 describes a typical interaction session and Section 4 concludes the paper.

2 The PIRATES framework

PIRATES (Personalized Intelligent Recommender and Annotator TEStbed) is a general framework for text-based content retrieval and categorization and exploits social tagging, user modeling, and information extraction techniques. Rather than proposing a rigid classification toolset, we have developed a testbed platform for integrating (and experimenting with) various tools and techniques, providing an interactive environment where users can customize the way they

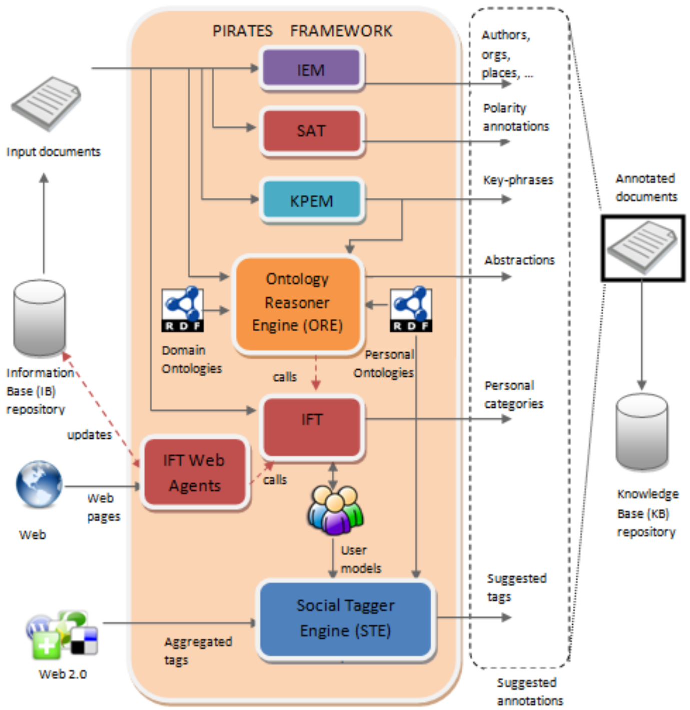


Figure 1. Overall architecture of PIRATES.

retrieve and classify information on the Web. The main feature of PIRATES concerns a novel approach that automates in a personalized way some typical manual tasks (e.g. content annotation and tagging). The framework operates on a set of input documents stored in the Information Base (IB) repository and suggests for these some personalized tags and other forms of textual annotations (e.g. key-phrases) in order to classify them. The original documents are then annotated with these tags, forming the Knowledge Base (KB) repository. Personalization is achieved exploiting user profiles (which represent the user interests), personal ontologies, personal tags, etc., as discussed in Section 3. Furthermore, PIRATES provides several mechanisms of user feedback that helps to provide personalized adaptive information.

The PIRATES architecture is illustrated in Figure 1. On the left-hand side, all the possible input sources are shown: single textual documents, specific IB repositories which can be contained within an e-learning knowledge management environment, and the Web, with specific (but not exclusive) focus on Web 2.0 portals, social networks, etc.. The right-hand side shows the suggested annotations and the resulting KB repository. The main modules of PIRATES are:

- `IEM` (*Information Extraction Module*), which is based on the GATE platform [10] to extract named entities, adjectives, proper names, etc. from input documents, contained in the IB.
- `SAT` (*Sentiment Analysis Tool*), which is a specific plug-in for personalized sentiment analysis (typically to be activated for online marketing applications), that is capable of mining consumer opinions in the blogosphere and classify them according to their polarity (positive, negative, or neutral) [11].
- `KPEM` (*Key-Phrases Extraction Module*), which implements a variation of the KEA algorithm [12] for key-phrases extraction. KPEM identifies n-gram key-phrases (typically n between 1 and 4) that summarize each input document. This information is provided to the user, and is also given as input to the subsequent modules.
- `ORE` (*Ontology Reasoner Engine*), which suggests new *abstract concepts* by navigating through ontologies, classification schemata, thesauri, lexicon (such as WordNet), etc. An abstract concept is identified by looking for a match between the annotations found by the other modules (IEM, KPEM, IFT, and STE) and the concepts stored in ontologies. When a match is found, ORE navigates through the ontology, looking for the common parent node which represents the more abstract term to suggest as annotation. ORE also assists users in creating personal ontologies with techniques similar to those described in [13].
- `IFT` (*Information Filtering Tool*), which evaluates the relevance (in the sense of topicality) of a document according to a specific model of user interests represented with semantic (co-occurrence) networks [14].
- `IFT Web Agents`, which continuously monitor the Web (and the blogosphere) looking for new information, cooperates with IFT to filter contents according to the user model, and updates the IB repository. IFT and its Web agents form together the Cognitive Filtering module discussed in [8].
- `STE` (*Social Tagger Engine*), which suggests new annotations for a document relying on *aggregated tags*, i.e. the user's personal tags (tags previously exploited) and the more popular tags used by the community of people that classify the same document in social bookmarking sites such as Del.icio.us[1], Faviki[2] or Bibsonomy[3]. This social information is integrated with content-based analysis techniques as discussed in [15].

3 A typical usage scenario

In this section we provide a typical scenario that illustrates a use case for our framework. Consider a user interested to read scientific publications in the area of software engineering. He trains the IFT tool providing the training data (e.g. 2-3 relevant papers in the field, some keywords and a short textual description for the argument) in order to setup the user model. After training, the IFT

[1] http://delicious.com
[2] http://www.faviki.com/pages/welcome/
[3] http://bibsonomy.org

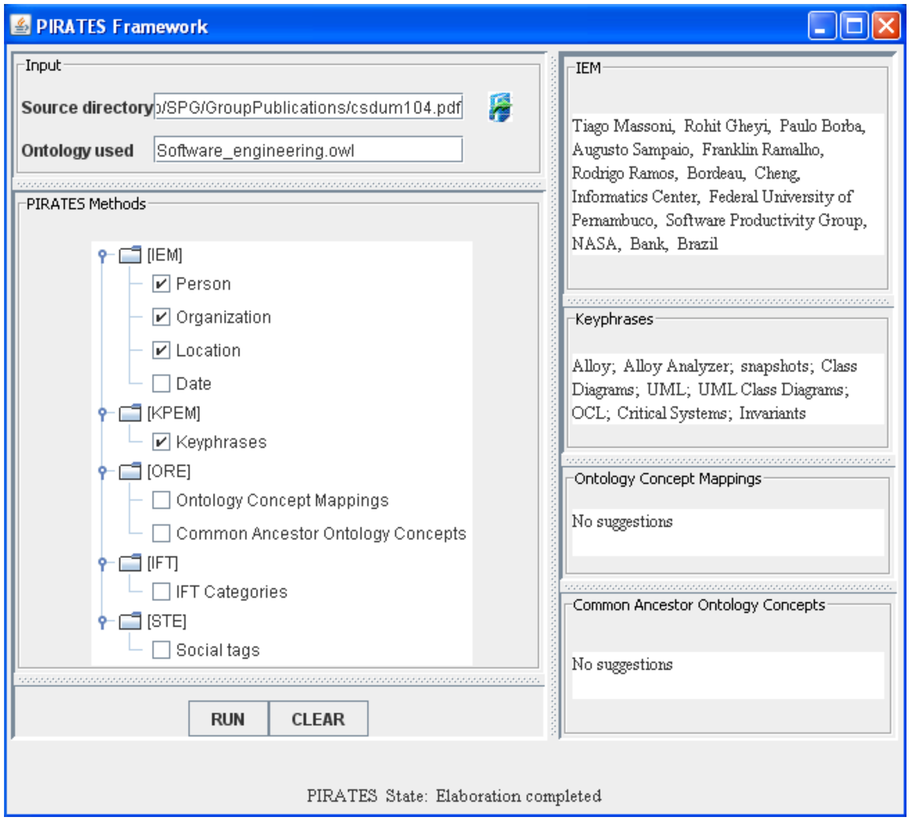


Figure 2. The PIRATES user interface running our example

agents periodically monitor the Web (in our case especially Web 2.0 sites such as Del.icio.us, Bibsonomy, CiteseerX[4], etc.), download new content and scrap selected data from them to filter out irrelevant information (e.g. ads and navigational links). When a relevant content (with respect to the user model) is retrieved, the agents add it to the IB repository and informs the user with a notification (e.g. an e-mail message). This information retrieval workflow has been already discussed in [14, 16], so in the rest of the section we concentrate on the classification features added by the PIRATES framework. Indeed, PIRATES aims expressly to support the user in *organizing* the IB repository, easing the work of classifying new contents by means of personalized tag suggestions.

Suppose now that an IFT agent notifies (among the others) the paper "A UML Class Diagram Analyzer"[5]. In order to classify this new content, the user can enable some PIRATES annotator modules, as illustrated in the left side of

[4] http://citeseerx.ist.psu.edu/

[5] http://twiki.cin.ufpe.br/twiki/pub/SPG/GroupPublications/csduml04.pdf.

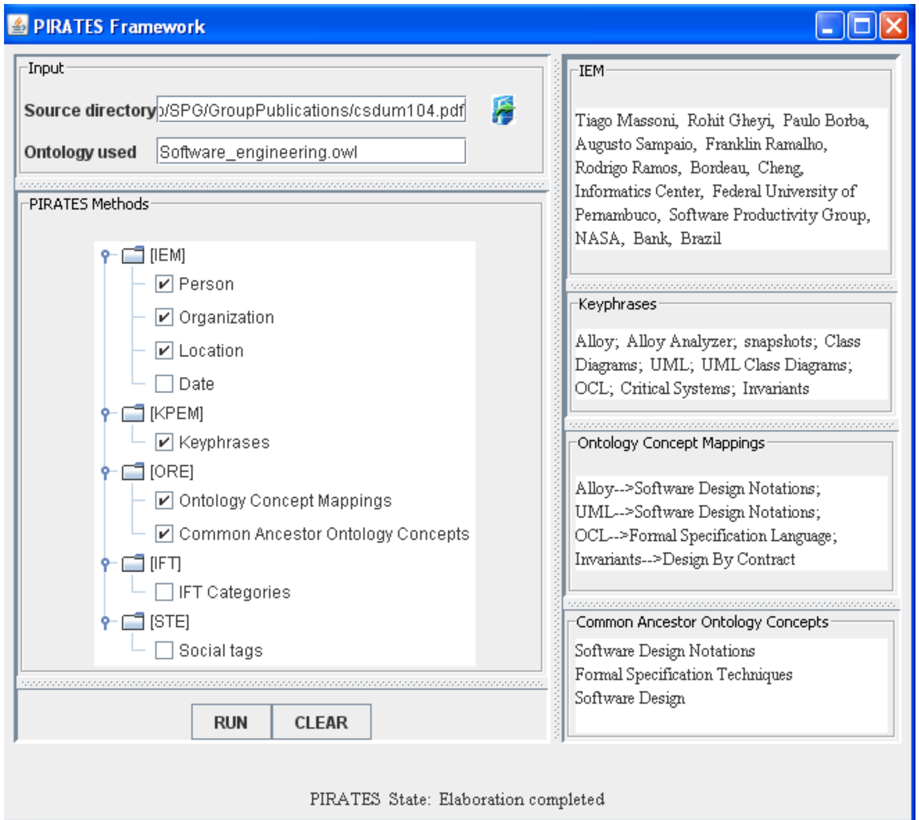


Figure 3. The PIRATES user interface running our example

Figure 2. Let us assume that he enables only IEM and KPEM modules in order to extract, respectively:

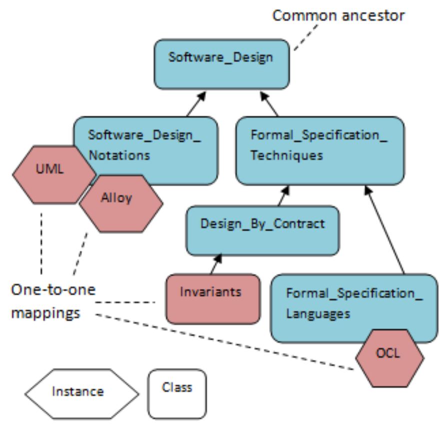
- person's names, organizations, and places (using IEM);
- keyphrases, i.e. n-grams long three terms at maximum (using KPEM).

With these settings, the framework produces the tag recommendations showed in the right side of Figure 2. In particular, the suggested tags concern persons such as the authors (`Tiago Massoni`, `Rohit Gheyi`, and `Paulo Borba`) and the people acknowledged in the paper (`Bordeau`, `Chang`, `Augusto Sampaio`, `Franklin Ramalho` and `Rodrigo Ramos`), locations (`Brazil`), and organizations cited in the text (the `Informatics Center` of the `Federal University of Pernambuco`, the `Software Productivity Group`, and the `NASA`). As keyphrases, KPEM provides many terms related to Alloy specification language (`Alloy`, `Alloy Analyzer`, `snapshots`), to UML (`UML`, `UML Class Diagrams`, `OCL`) and to the specification of dependable systems (`Critical Systems`, `Invariants`).

Tags suggested by IEM + KPEM + ORE modules

IEM	***Orgs***: Federal University of Pernambuco, Informatics Center, Software Productivity Group, NASA ***Places***: Brazil ***People***: Paulo Borba, Tiago Massoni, Rohit Gheyi, Franklin Ramahlo, Rodrigo Ramos, Augusto Sampaio
KPEM	***Keyphrases***: Alloy Alloy Analyzer UML OCL Critical Systems UML Class Diagrams Invariants Snapshots
ORE	***Ontology tags***: Software Design Notation, Formal Specification Language, Design by Contract, Formal Specification Techniques, Software Design

(a) IEM, KPEM, and ORE outputs



(b) Ontology reasoning

Figure 4. Personalized annotations proposed by PIRATES

The tag suggestions provided so far are extracted by the text present in the input document: no personalization is present at all. Suppose now that the user enables also the ORE module which exploits (in our example) a *personal* ontology[6] in the field of software engineering (see left side of Figure 3).

ORE implements a navigation strategy, taking in input the key-phrases extracted by other annotators (KPEM in this case). For four out of the suggested key-phrases (i.e. `Alloy`, `UML`, `OCL`, and `Invariants`), ORE identifies a corresponding one-to-one match in the ontology (see Figure 4(b)). Starting from these nodes, ORE uses a spreading activation algorithm to find common ancestors representing more abstract subjects. Then both one-to-one ontology mappings and common ancestors are provided by PIRATES as potential tag recommendations, as summarized in Figure 4(a). The ontology navigation process highlighted by the spreading activation algorithm is depicted in Figure 4(b). In conclusion, the ORE module recommends five new tags which are not present in the text (i.e. `Software Design Notation`, `Formal Specification Language`, `Design by Contract`, `Formal Specification Techniques`, and `Software Design`)[7].

[6] We exploit an extended version of the existing domain ontology available from http://www.seontology.org/.

[7] Note also that tag `Design by Contract` was not already present nor in the input document, nor in the original ontology, but it was added to the ontology by means

These tags represent *abstractions* of the key-phrases extracted by the other annotators available in PIRATES.

4 Conclusions

We believe that the presented framework is a promising approach to automatic, personalized classification of Web contents. It is a first step in the direction of automatically organize document repositories into personal concept maps, moving from information to knowledge. The development of PIRATES has been planned in an incremental fashion, interleaved with experimental evaluation. Several modules have been already developed and integrated in a testbed environment: IEM with the sentiment analysis plug-in [16], KPEM with key-phrases extraction capabilities, and the Cognitive Filtering comprising an extended version of IFT capable to monitor Web 2.0 sources (specifically newsgoups, forums, and blogs). The integration of these modules is currently being evaluated. Prototyping and integration of ORE, SAT, and STE within PIRATES are ongoing processes, and evaluation experiments are planned. Moreover, we are working specifically on integrating the PIRATES modules in a Web-based version of the environment, which let us validate each module thoroughly. Finally, we have also planned to implement the conceptual map editor described in [8] in order to completely validate the framework.

References

1. Brusilovsky, P., Tasso, C.: Preface to special issue on user modeling for web information retrieval. User Modeling and User-Adapted Interaction **14**(2-3) (2004) 147–157
2. Gauch, S., Speretta, M., Chandramouli, A., Micarelli, A.: User profiles for personalized information access. In: The Adaptive Web. (2007) 54–89
3. Bunt, A., Carenini, G., Conati, C.: Adaptive content presentation for the web. In: The Adaptive Web. (2007) 409–432
4. Katakis, I., Tsoumakas, G., Vlahavas, I.: Multilabel text classification for automated tag suggestion. In: Proc. of the ECML/PKDD 2008 Discovery Challenge. (2008)
5. Marchetti, A., Tesconi, M., Ronzano, F.: Semkey: A semantic collaborative tagging system. (2007)
6. Lonchamp, J.: A platform for cscl practice and dissemination. In: ICALT '06: Proc. of the Sixth IEEE International Conference on Advanced Learning Technologies, IEEE Computer Society (2006) 66–70
7. Kim, H., Yang, S., Jung, J., Kim, K., Breslin, J., Decker, S., Kim, H.: Combining tags and the semanticweb for linked tagging data (2008)
8. Casoto, P., Dattolo, A., Ferrara, F., Pudota, N., Omero, P., Tasso, C.: Generating and sharing personal information spaces. In: Proc. of the Workshop on Adaptation for the Social Web, 5th ACM Int. Conf. on Adaptive Hypermedia and Adaptive Web-Based Systems. (2008) 14–23

of a user feedback mechanism provided by PIRATES. This is where personalization comes from.

9. Omero, P., Polesello, N., Tasso, C.: Personalized intelligent information services within an online digital library for medicine: the bibliomed system. In: IRCDL '07: Proc. of the Third Italian Research Conference on Digital Library Systems. (2007) 46–51
10. Cunningham, H.: Gate, a general architecture for language engineering. Computers and the Humanities **36** (2002) 223–254
11. Casoto, P., Dattolo, A., Tasso, C.: Sentiment classification for the italian language: A case study on movie reviews. Journal of Internet Technology **9**(4) (2008) 365–373
12. Frank, E., Paynter, G., Witten, I., Gutwin, C., Nevill-Manning, C.: Domain-specific keyphrase extraction. In: IJCAI '99: Proc. of the Sixteenth International Joint Conference on Artificial Intelligence, Morgan Kaufmann (1999) 668–673
13. Speretta, M., Gauch, S.: Using text mining to enrich the vocabulary of domain ontologies. Web Intelligence and Intelligent Agent Technology, IEEE/WIC/ACM International Conference on **1** (2008) 549–552
14. Tasso, C., Asnicar, F.A.: ifweb: a prototype of user model-based intelligent agent for document filtering and navigation in the world wide web. In: Adaptive Systems and User Modeling on the WWW, 6th UM Inter. Conf. (1997)
15. Tasso, C., Rossi, P., Virgili, C., Morandini, A.: Exploiting personalization techniques in e-learning tools. In: SW-EL'04: Proc. of the Workshop on Applications of Semantic Web Technologies for Adaptive Educational Hypermedia. (2004)
16. Pudota, N., Casoto, P., Dattolo, A., Omero, P., Tasso, C.: Towards bridging the gap between personalization and information extraction. In: IRCDL '08: Proc. of the Forth Italian Research Conference on Digital Library Systems. (2008) 33–40

A Personalized Tag-Based Recommendation in Social Web Systems

Frederico Durao and Peter Dolog

IWIS — Intelligent Web and Information Systems,
Aalborg University, Computer Science Department
Selma Lagerlöfs Vej 300, DK-9220 Aalborg-East, Denmark
{fred,dolog}@cs.aau.dk

Abstract. Tagging activity has been recently identified as a potential source of knowledge about personal interests, preferences, goals, and other attributes known from user models. Tags themselves can be therefore used for finding personalized recommendations of items. In this paper, we present a tag-based recommender system which suggests similar Web pages based on the similarity of their tags from a Web 2.0 tagging application. The proposed approach extends the basic similarity calculus with external factors such as tag popularity, tag representativeness and the affinity between user and tag. In order to study and evaluate the recommender system, we have conducted an experiment involving 38 people from 12 countries using data from Del.icio.us, a social bookmarking web system on which users can share their personal bookmarks.

Key words: personalization, recommendation, tags, bookmarks, similarity

1 Introduction

Collaborative tagging systems have become increasingly popular for sharing and organizing Web resources, leading to a huge amount of user generated metadata. Tags in social bookmarking systems such as *del.ici.ous*[1] are usually assigned to conceptualize, categorize, or sharing a resource on the Web so that users can be reminded of them later and find their bookmarks in an easy way. Invariably, tags represent some sort of affinity between user and a resource on the web. By tagging, users label resources on the Internet freely and subjectively, based on their sense of values [11]. In this sense, tags from social bookmarking systems represent a potential mean for personalized recommendation because through them it is possible to identify individual and common interests between unknown users. Nevertheless, although huge amount of tag data is available, to compute an individual preference in order to perform efficient recommendation is still a challenging task. In this paper, we propose a tag-based recommender system which recommends bookmarks by calculating the similarity of their tags. The

[1] http://delicious.com

proposed approach besides basic similarity takes into account external factors such as *tag popularity*, *tag representativeness* and the *affinity between user and tag*. We utilize a cosine similarity measure between tag vectors to calculate basic similarity of the pages. We measure tag popularity as a count of occurrences of a certain tag in the total amount of web pages. We utilize term frequency measure to compute tag representativeness for a certain web page. The tag affinity between a user and a tag is calculated as a count of how many times the user utilized the tag at different web pages. We propose a formula which considers all these factors in a normalized way and gives a ranking of web pages for particular user.

The goal of this study is to analyze whether tags can be utilized to generate personalized recommendations. This assumption can be assessed by running an experiment whereby users expresses their satisfaction about the received recommendations. Based on this, we conducted an experiment involving 38 people from 12 countries using data from del.icio.us to evaluate the efficiency of the proposed approach and social aspects such as the purposes behind the tagging activity. The contribution of the paper is therefore:

- The proposed recommendation approach based on similarity, tag representativeness, popularity, and affinity; and
- Findings from the evaluation which show that the approach performs well in a non-controlled environment with people from different domains and intentions.

The paper is organized as follows: In Section 2 we discuss related work. In Section 3 we introduce the motivation of this work. Section 4 describes the factors of similarity that we will analyze. Section 5 presents the experiment and the achieved results. Section 6 addresses a discussion about the findings from the experiment and Section 7 presents the conclusion and future works.

2 Related Work

Tags have been recently studied in the context of recommender systems due to various reasons. [8] argues for a solution where tagging from social bookmarking provides a context for recommender systems in terms of context clues from tags as well as connectivity among users to improve the collaborative recommender system. Similar to our approach, [9] constructed a web recommender based on large amount of public bookmark data on Social Bookmarking system. For means of personalization, they utilize folksonomy tags to classify web pages and to express user's preferences. By clustering folksonomy tags, they can adjust the abstraction level of user's preferences to the appropriate level. In spite of the proximity with our study, the [9] experiment did not measure the efficiency of the recommendations in terms of user satisfaction what could have provided us a parameter for comparison. [13] extends a content based recommender system by deriving current and general personal interests of users from different tags according to different time intervals. However, unlike our approach, the similarity of the tags is given by of two Naive Bayes classifiers trained over different timeframes: one classifier predicts the user's current interest whereas the other classifier predicts the user's general interest in a bookmark. The two classifiers are trained with a subset of the bookmarks created by a user. The tags of each bookmark, converted into a "bag of words", are used as training features. The bookmarks are recommended in case both of the two classifiers predict a bookmark as interesting. The effectiveness of the recommendations however is totally dependent on the quality of the subset of bookmarks used for training the classifiers.

[5] shows the benefits of using tag based profiles for personalized recommendations of music on Last.fm. Similar understanding over the product items as subject of recommendations is considered as another factor in addition to the similar tags when personalizing recommendations given by a tag based collaborative recommender system in [14]. The purpose of tags vary as well as tagging itself may be influenced by different factors. For example, [10] studies a model for tagging evolution based on community influence and personal tendency. It shows how 4 different options to display tags affect user's tagging behavior. [2] studies how the tags are used for search purposes. It confirms that the tags can represent different purpose such as topic, self reference, and so on and that the distribution of usage between the purposes vary across the domains. Other works such as [12] and [7] coined the term *emergent semantics* as the semantics which emerge in communities as social agreement on tag's meaning based on its more frequent usage instead of the contract given by ontologies from ontology engineering point of view. However, the approaches based on emergent semantics are characterized by the power law which gives a long tail of the tags of which semantics have not emerged yet. Therefore, [3] looks at grounding of the tag relatedness with a help of WordNet.

In this paper we look at, how multiple factors such as similarity, tag popularity, tag affinity to a user and tag representativeness can be used together to achieve recommendations. We also wanted to see the personalized recommendations in an open context with users of different background.

3 Running Example or Motivating Scenario

Tags in social bookmarking systems allow users to express their preference by sharing their bookmarks. Tags are personalized piece of information which can be utilized to identify common interests between users. Compared to traditional collaborative rating [6], tags can reflect the user's preference to a given resource in a meaningful way [11]. Based on these premises, we investigate the feasibility of using tags as one approach for the generation of *personalized recommendations*. Along this article, the word *resource* will be used as generic term to refer to document, video, image, text, files or any sort of asset which can be tagged and referenced by URI.

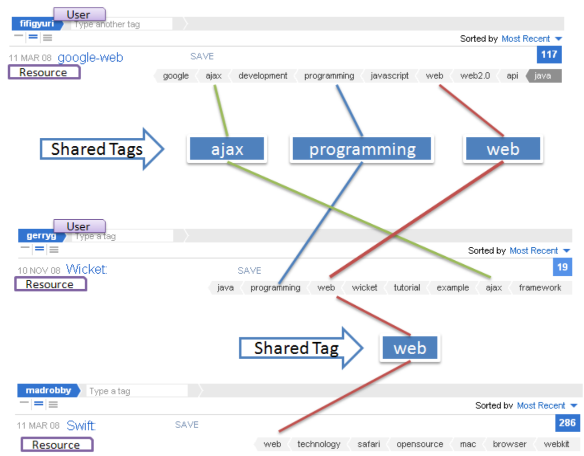


Fig. 1. User x Bookmarks x Tags

Let us now look at a scenario which explains our approach and ideas behind it. According to Figure 1, the resources **Google-Web** and **Wicket** share tags *ajax,programming* and*web* whereas resources *Wicket* and *Swift* only share tag **web**. Considering exclusively the similarity between tags, the resources **Google-Web** and **Wicket** have higher probability of being about the same content than **Wicket** and **Swift**. Based on this fact, their authors should be noticed about the existence of similar resources around. Furthermore, the notification could be prioritized (or emphasized) if the similar tags correspond to the most frequent tags of the authors or they are very representative for the resource they describe. This scenario was presented for illustrating how tag similarity can be computed for means of personalization. Although similarities can be found when the tags are syntactically identical, a number of pessimist scenarios may take place and must be considered such as: *resources which have similar tags but incorrect spelling* - since tags are informal and free writing, no syntax control is assured. For instance, tags "programming" and "programing" looks the same except for the fact the second one is missing the letter *m*; *resources whose tags are syntactically different but similar semantically* - this is a case of synonymy and to overcome it some semantic assistance is needed either by use of domain ontologies or looking up for synonymies in dictionary. For instance, tags "work" and "labor" looks different but share the same meaning. The obstacle is that generic dictionary sometimes is not enough to provide the correct meaning of specific terms in a given *context*; and *resources which share same tags with different meanings* - this is the well known case of polysemy. For instance, the tag "windows" can be about the operational system or the house artifact.

4 The Approach

In order to generate personalized recommendations, we propose an extension method for the calculation of basic similarity between tags. We combine the cosine similarity calculus with other factors such as *tag popularity*, *tag representativeness* and *affinity user-tag* with the purpose of reordering the original raking in recommendation and generate personalized ones.

We define the document score as:
$Ds = \sum_{i=1}^{n} weight(Tag_i) * \sum_{i=1}^{n} representativness(Tag_i)$, where n is the total number of existing tags in the repository.

We define the tag user affinity as:
$Affinity_{(u,t)} = card\{r \in Documents \mid (u,t,r) \in R, R \subseteq U \times T \times D\}/card\{t \in T \mid (t,u) \in R_u, R_u \subseteq U \times T\}$, where t is a particular tag, u particular user, U is a set of users, D set of resources and T set of tags.

Finally, similarity is computed as:
$Similarity_{(D_i,D_{ii})} = [Ds_{D_i} + Ds_{D_{ii}} * cosine\_similarity(T_{D_i}, T_{D_i}i)] * Affinity_{(u,t)}$, where Ds is the document score and T is set of tags of a particular document.

Informally, each one of the factors in the above formulas is calculated as follows:

– **Cosine Similarity** — Our tag similarity is a variant on the classical cosine similarity familiar from text mining and information retrieval [1] whereby two items are thought of as two vectors in the m dimensional user-space. The similarity between them is measured by computing the cosine of the angle between these two vectors.
– **Tag Popularity** — Also called *tag weight*, is calculated as a count of occurrences of one tag per total of resources available. We rely on the fact that the most popular tags are like anchors to the most confident resources. As a consequence, it decreases the chance of dissatisfaction by the receivers of the recommendations.
– **Tag Representativeness** — It measures how much a tag can represent a document it belongs. It is believed that those tags which most appear in the document can better represent it. The *tag representativeness* is measured by the *term frequency*, a broad metric also used by the Information Retrieval community.
– **Affinity between user and tag** - It measures how often a tag is used by a user. It is believed that the most frequent tags of a particular user can reveal his/her interests. This information is regarded as valuable information for personalization means. During the comparison of two resources, the similarity is boosted if one of the resources contains top tags of the Author from the other resources around.

Further, we have set empirically that for one tag represent the user's preference, its frequency of use must be 70% closer to the most frequent tag of the user. In the case on which there are no tags to satisfy this condition, it is assumed the user does not have a clear preference.

5 Experimental Evaluation

In our evaluation, we opted to measure the degree of satisfaction of users about the received recommendations. The user's feedback will allow us to evaluate the quality of recommendations produced from our framework. Although, we recognize that *precision and recall* are metrics which could be used to evaluate the effectiveness of the system, we believe that user's participation provides more precious feedback for means of personalization. In this sense, we invited users by sending a number of invites in various mailing lists from different natures, not only related to technology. We explained the purpose of the experiment and also we outlined easiness of the participation aim at attracting more users not related to technology. To our surprise, within less than 1 month 44 participants had accepted to participate voluntarily. In spite of 44 initial positive replies, only 38 participants joined until the end. Finally, we had 38 participants from 12 countries interested in many different subjects. Data for our experiment was collected from *del.icio.us* in November 2008 comprising 5542 tags and 1143 bookmarks.

Methodology. We have created a *del.icio.us* user account for each participant on which he/she was invited to add at least 10 bookmarks with minimally 3 tags each as suggested. Each participant received the top 5 most similar recommendations to their bookmarks based on the tags assigned to them. Then the participants were asked to select which items of the recommended set matched to their bookmarks. As soon as the participants finished their contribution, the overall results were shared with the participants as well as the reflections and findings.

5.1 Expected Results

Considering that the experiment took place in non-controlled environment (as del.ici.ous is) with diverse audience (people from technology, health, education, biology, etc), we did not expected 100% of acceptance of the recommendations. Some reasons for this are: i) *diversity of culture and background* - Since the participants are from many different countries and have distinct backgrounds, it increases significantly the disparity between tags i)*Syntax of tags* - As previously introduced, the tags assigned by the participants were not under any syntax control. Users could have written their tags in many different (and personalized) ways, for instance, the tag *web2.0* can be also tagged as *web20*, *web2_0* or *web_20* and iii) *difficulty to identify user's preferences through the tags* - if users bookmark web resources of different domains (e.g. sports, education, engineering), hardly any some tags will predominate over others, which increases the difficulty of precisely indentifying user's main preferences.

Based on the reasons addressed previously, we consider the result as satisfactory if more than half of recommendations are accepted (or selected) by the participants. If 80% of the recommendations are accepted, we claim the results as excellent (and unexpected), on the other hand, bellow of 50%, we understand that the proposal has to be reviewed and improved with the findings achieved from the experiment analysis.

5.2 Results from the experiment

Figure 2 provides a clear picture of how many items were accepted by each participant among the five recommended. For instance, the participant 9 accepted 2 recommendations amongst the 5 suggested in the set.

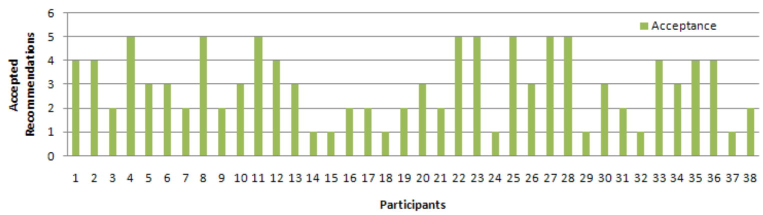


Fig. 2. Amount of recommendations accepted by each participant.

The graph shows that at least one recommendation from each set recommended was accepted. Moreover, it shows that 8 participants were satisfied with the whole set of recommendations and 7 participants accepted only 1 item from whole the set which was recommended. Due to the graph distribution, it is possible to preliminary argue that the acceptance is well balanced. However, in order to evaluate the overall results more properly, we stipulated a threshold by calculating the *arithmetic mean* of the acceptance, which was *2.971*. In the following, we calculated the *standard error of the mean* in order to verify the variance of the mean and consequently perform more concrete argumentation on top of the results obtained. The standard error is given by $se = \frac{s}{\sqrt{n}}$, where s is the sample standard deviation (i.e. the measure of the dispersion of the data set), n is the size (number of observations) of the sample. More about the standard deviation can be found at [4]. Finally, the standard error of the mean obtained was *0.23*, which allow us to judge our results based on the stipulated threshold (2.971) without significant variance.

Figure 3 shows the histogram of the accepted recommendations. It shows the frequency of the accepted items against the amount of participants. Figure 3 shows that 22 participants (or 58% of all) accepted 3 or more recommendations (above the threshold) from the five that was suggested. However, 16 (or 42% of all) participants accepted only 2 or 1 recommendations, below the stipulated threshold. Focusing only on the 16 participants who accepted between 2 or 1 items, 7 of them accepted only 1 recommendation from the whole set. This means that 7 participants together rejected 80% of the recommendations that were sent to them (i.e. 35 sent and 28 rejected). In order to investigate this particular inconvenience, we analyzed the tags assigned to these rejected items.

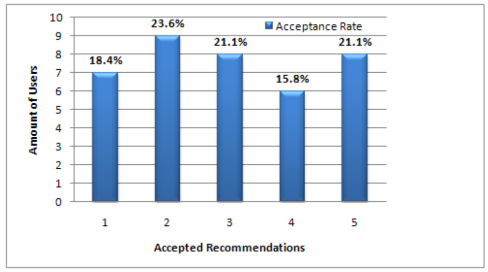


Fig. 3. Frequency of recommendations accepted

We figured out that although the recommendations had been generated correctly (considering the tag syntax), most of them was really out of context and far from the user's interest. We turn out with some findings: i) some tags did not show clearly any relatedness with the resource domain; ii) ambiguity problems, more particularly synonymy, when the tags share same syntax but are semantically different; and iii) impossibility of identifying user's preference due to the low number of tags of some users.

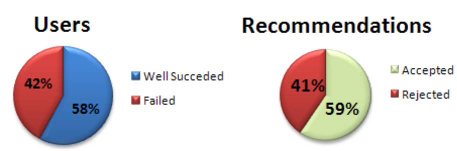


Fig. 4. Final results

Figure 4 summarizes the overall results. As already pointed, the pie chart on the left shows that 58% of the participants received a set of recommendations above the stipulated threshold while 42% received a set of recommendations bellow of it. The pie chart on the right shows the overall rate of well succeeded recommendations in which 59% of the whole recommendations was accepted and 41% was rejected. In summary, the final result cannot be considered excellent but satisfactory, since 58% of the overall set of recommendations was above the threshold and 59% of the overall recommendations was accepted. In spite of achieving satisfactory results, it is not possible to affirm the proposed solution is ready for large usage. Improvements to overcome the ambiguity problems are needed and further experiments must be performed again in order to provide more insights about the system evolution. On the other hand, the results indicate the research is on the right track.

6 Discussion

From the evaluation, we realized that the proposed recommender approach performs satisfactorily well even in a non-controlled environment with users from different domains and backgrounds. Based on the results, we understand the multifactor approach can be utilized to generate personalized recommendations. However, it is quite important to discuss the problems found from the unsuccessful recommendations. The ambiguity problems such as synonymy and polysemy can be attenuated by using WordNet dictionary since it has been employed for computing semantic similarity [3]; however, they are generic and do not cover particular meanings from specific domains. Focusing on the problem of (or lacking of) relatedness between tags and resource, we believe that a viable solution is to capture the purpose why users are tagging as studied by [2]. If the purpose is asked explicitly, then we have a usability problem, i.e. one additional step to simply assign a tag to a resource. On the other hand, to infer the purpose of a tag in a given resource relies on long observation about the users tagging activity. Moreover, if the inference is uncertain, a number of bad recommendations can be processed. Concerning the difficulty of identifying the user's preference using tags, we understand that the factor *time* should be taken into account. The user's preference changes along the time and these changes can be reflected in the tags as well.

7 Conclusion and Future Works

This paper introduced a tag-based recommender system which generate personalized recommendations. The efficiency of the system rely on cosine similarity calculus with additional factors such as *tag popularity*, *tag representativeness* and *affinity between user and tag*. A experiment involving 38 people from 12 countries using data from del.icio.us was conducted to evaluate the efficiency of the system for means of personalization.

The overall results showed that approximately 60% of the recommendations succeeded and the proposed recommender system requires improvements. As a future work, we propose to perform semantic similarity to overcome ambiguity problems (as mentioned in the pessimist scenarios) and investigate the purpose of the tags when they are assigned to a resource. Finally, comparisons with other approaches must be addressed since the current evaluation methodology only assesses user's satisfaction using the specific algorithm.

8 Acknowledgment

The research leading to these results is part of the project "KiWi - Knowledge in a Wiki" and has received funding from the European Communitys Seventh Framework Programme (FP7/2007-2013) under grant agreement No. 211932.

References

1. Ricardo Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley, May 1999.
2. Kerstin Bischoff, Claudiu S. Firan, Wolfgang Nejdl, and Raluca Paiu. Can all tags be used for search? In James G. Shanahan et al., editor, *Proc. of the 17th ACM Conference on Information and Knowledge Management, CIKM 2008*, pages 193–202, Napa Valley, California, USA, October 2008. ACM.
3. Ciro Cattuto, Dominik Benz, Andreas Hotho, and Gerd Stumme. Semantic grounding of tag relatedness in social bookmarking systems. In Amit P. Sheth et al., editor, *The Semantic Web - ISWC 2008, Proc. of the 7th Intl. Semantic Web Conference, ISWC 2008*, volume 5318 of *Lecture Notes in Computer Science*, pages 615–631, Karlsruhe, Germany, October 2008. Springer.
4. Tony F. Chan and John Gregg Lewis. Computing standard deviations: accuracy. *Commun. ACM*, 22(9):526–531, 1979.
5. Claudiu S. Firan, Wolfgang Nejdl, and Raluca Paiu. The benefit of using tag-based profiles. In Virgílio A. F. Almeida and Ricardo A. Baeza-Yates, editors, *Fifth Latin American Web Congress (LA-Web 2007)*, pages 32–41, Santiago de Chile, November 2007. IEEE.
6. Rong Jin and Luo Si. A study of methods for normalizing user ratings in collaborative filtering. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 568–569, New York, NY, USA, 2004. ACM.
7. Peter Mika. Ontologies are us: A unified model of social networks and semantics. *J. Web Sem.*, 5(1):5–15, 2007.
8. Reyn Nakamoto, Shinsuke Nakajima, Jun Miyazaki, and Shunsuke Uemura. Tag-based contextual collaborative filtering. *IAENG Intl. Journal of Computer Science*, 34(2), 2007.
9. Satoshi Niwa, Takuo Doi, and Shinichi Honiden. Web page recommender system based on folksonomy mining for itng '06 submissions. In *ITNG '06: Proc. of the Third Intl. Conference on Information Technology: New Generations*, pages 388–393, Washington, DC, USA, 2006. IEEE.
10. Shilad Sen, Shyong K. Lam, Al Mamunur Rashid, Dan Cosley, Dan Frankowski, Jeremy Osterhouse, F. Maxwell Harper, and John Riedl. tagging, communities, vocabulary, evolution. In Pamela J. Hinds and David Martin, editors, *Proc. of the 2006 ACM Conference on Computer Supported Cooperative Work, CSCW 2006*, pages 181–190, Banff, Canada, November 2006. ACM.
11. Gene Smith. *Tagging: People-powered Metadata for the Social Web (Voices That Matter)*. New Riders Press, December 2007.
12. Steffen Staab. Emergent semantics. *IEEE Intelligent Systems*, 17(1):78–86, 2002.
13. Pavan Kumar Vatturi, Werner Geyer, Casey Dugan, Michael Muller, and Beth Brownholtz. Tag-based filtering for personalized bookmark recommendations. In *CIKM '08: Proceeding of the 17th ACM conference on Information and knowledge mining*, pages 1395–1396, New York, NY, USA, 2008. ACM.
14. Shiwan Zhao, Nan Du, Andreas Nauerz, Xiatian Zhang, Quan Yuan, and Rongyao Fu. Improved recommendation based on collaborative tagging behaviors. In *IUI '08: Proc. of the 13th Intl. conference on Intelligent user interfaces*, pages 413–416, New York, NY, USA, 2008. ACM.

Using Asynchronous Client-Side User Monitoring to Enhance User Modeling in Adaptive E-Learning Systems

David Hauger
hauger@fim.uni-linz.ac.at

Institute for Information Processing and Microprocessor Technology,
Johannes Kepler University, Linz, Austria
http://www.fim.uni-linz.ac.at

Abstract. Traditionally, user modeling in adaptive hypermedia systems is based upon monitoring requests of resources on the server. This information, however, is relatively vague and may lead to ambiguous or uncertain conclusions. One approach to improve granularity and accuracy in user models is monitoring "client-side" user interactions in the browser. Despite their steadily growing importance, especially along with the emergence of Web 2.0 paradigms, up to now such interactions are hardly being monitored. This paper shows how, by means of Web 2.0 technologies, additional information on user interactions can be retrieved, and discusses the benefits of this approach for user models (like accuracy, granularity and immediate availability of data out of the request-response cycle) as well as for subsequent adaptations.

Key words: user monitoring, interaction, Web 2.0, user modeling, asynchronous adaptive hypermedia

1 Introduction

Traditional user modeling techniques of adaptive hypermedia systems (AHS) monitor requests of resources on the server to make assumptions on users. However, the fact that a document has been requested might be an insufficient source of information to determine whether it has been read. After servers have transmitted a hypermedia document, there is usually no further feedback to tell whether it is still open or which of its sections is being read. If a browser's request to the server (which refers to pages as a whole) is the only source of information available, then one cannot make assumptions in relation to the document's comprising parts or sections. Therefore, documents have to be treated as atomic items.

Additional information may be retrieved by calculating the time between requests (e.g. in Knowledge Sea [1]) or by adding semantic meta data (e.g. by mapping pages to concepts as in AHA! [2]). More and more elaborate algorithms analyze the limited available data to increase granularity and accuracy of user models, while still lacking information on user interactions inside the browser.

In contrast to these approaches, the one put forward in this paper aims to improve the monitoring process itself, and suggests a way to asynchronously monitor client-side user interactions, which user models and consequently AHS should benefit from. Although, especially with the emergence of Web 2.0, the importance of asynchronous client-server communication and client-side interactions have been increasing, they are hardly being used in terms of user modeling at the moment.

2 Client-Side User Monitoring

Instead of exclusively monitoring requests on server-side, interactions could as well be monitored directly within the browser.

2.1 Limitations of Traditional Approaches

As long as observation is reduced to server-side monitoring, user interactions that do not cause browser requests can not be observed, which limits possible assumptions on user behavior. Some of the questions that can not be answered due to the lack of information on client-side interactions are [3]:

- Whether a user has spent time on the page
- Whether a user has had a look at the whole page
- Whether a user has read a page
- Whether a user is interested in the page
- Whether a user has understood a page
- Why a user has skipped a page

Moreover, although Brusilovsky [4] stated that "the user can prefer some nodes and links over others and some parts of a page over others" already in 1996, current approaches are still not able to tell these differences for text nodes, as reading or copying text does not result in new requests. Retrieving more fine-grained information on which parts of a page have been read would consequently lead to better results than regarding a whole page as an atomic unit and mapping it to a concept.

2.2 Anticipated Benefits through Client-Side User Monitoring

As client-side monitoring provides more information on the users' interactions, it allows additional assumptions and increases the level of certainty for conclusions. This might help to improve both models and resulting adaptations.

Improving Accuracy of Existing User Models One evident example where user models can be improved is the assumption of a user "having read a page". Traditional systems make use of the information that a document has been requested. Some more advanced systems consider the time of a subsequent request to identify the "time spent reading" [1]. However, a user might have opened several documents in different browser windows or tabs at the same time. Switching between them leads to wrong values for the "time spent reading" as the user is always assumed to be reading the most recently requested page. Moreover, there is no information on whether the user has seen the whole page (e.g. scrolled down if the document is larger than the browser window). In most cases the document being read will have the focus, which can be checked by client-side monitoring. Being displayed within the active browser window is a stronger indicator for determining which page is being read, than supposing that the active document is always the most recently requested one. Moreover, it can be assumed that a document is being read if the user is interacting with it (periods of inactivity can be identified). Monitoring scroll events allows for checking whether all parts of a hypermedia document have been visible within the browser window and can be used to calculate the time spent on each segment of the page.

Improving Granularity of Existing User Models Hijikata [5] showed that text tracing, link pointing, link clicking and text selection may be used to identify interest. Of all these interaction types only link clicking causes a server-side event. The others are not being monitored by traditional systems. Therefore the systems usually take the amount of requests (sometimes with respect to the assumed time spent reading) to determine interests. Using client-side interaction-based user monitoring, all types of interactions can be used to set up a more detailed user model. Moreover, it is possible to define more granular user model attributes; e.g. not only read/unread, but also the percentage of a document or concept that has been read.

Independence of the Request-Response Cycle If the transmission of data depends on subsequent requests, information is lost, if users leave a page. When using browser history or parallel windows for navigation, incorrect data is transmitted. Asynchronous monitoring is able to overcome these restrictions by retrieving data out of the request response cycle, and additionally results in a model always being up-to-date. Moreover, if changes in the user model require immediate adaptations, this can be achieved as well. As the communication takes place asynchronously, the system might offer help, highlight important elements or provide any other (non-distracting) adaptation while the user is still interacting with the page [6].

3 State of the Art

Attempts concerning client-side monitoring have already been made by Goecks and Shavlik [7] who used JavaScript to log mouse and scrolling activities and summed it up into a "level of activity". Mapping this value to a set of keywords extracted from the visited page has been used to determine the "interests of the user". Hofmann et al. [8] extended this approach by sending information on all events to the server (especially timestamps).

Due to technical boundaries (workarounds like refreshing iframes, Java applets or flash movies have hardly been used) both monitoring and adaptations were traditionally bound to a strict request-response cycle. In order to overcome these limitations, custom browsers have been developed (like “AVANTI” [9], which monitored the time spent reading and client-side interactions in order to set up a user model and determine disabilities); or browser plugins (like the agent-based approach in [7] monitoring the time spent on a page, mouse events, scrolling and status bar changes and using this information as input for a neural network). While effective, the necessity to install additional software limited the applicability of the approach.

Within “The Curious Browser” [10], the browser itself was monitoring and storing user interactions and compared this information to explicit ratings of users. Analyzing the results, Claypool et al. discovered that the amount of scrolling, the time spent on a page and combination of these two pieces of information is strongly correlated with the explicit ratings. Individual scrolling methods and mouse clicks were ineffective in terms of determining strong explicit interests, but helped (when missing) to identify the least interesting content. These results prove that client-side interactions may be used to get valid information on users' interests.

With the emergence of Web 2.0 and asynchronous web technologies like AJaX (running in the background, not requiring additional plugins) some limitations for continuous and unobtrusive monitoring of user interactions are now addressed. Using these technologies not only for presentation, but also for user monitoring has already been suggested for help systems by Putzinger [6]. He used mouse and keyboard events to determine the “locus of attention”, especially focusing on input elements. The gained information has been used to adaptively provide help for the user.

As Web 2.0 technologies extend the range of what is technically possible, it is now up to researchers to answer questions that could not be addressed earlier. This includes increasing granularity and accuracy of user models [3]. The current challenge lies in discovering how client-side interactions can be monitored and processed to retrieve more semantic information about users and their behavior.

4 Asynchronous Client-Side User Monitoring

In order to monitor user interactions inside the browser without requiring additional software, client-side scripting languages are required. Although there are several possibilities, for the system currently being developed JavaScript has been selected as the most appropriate choice, as it is widely supported and used. AJaX is used to send the collected data to the server as it is unobtrusive, not bound to the request-response cycle and natively supported by modern browsers.

4.1 Monitoring Client-Side Events

JavaScript already provides a set of events that can easily be monitored. The current system uses various mouse and keyboard events, scrolling, window resizing, window blur, window focus and document ready.

However, some of these events occur too frequently for continuous monitoring, which is especially true for mouse moves and scrolling events. They are an indicator for interest and need to be monitored. For a single mouse move the browser may generate a large amount of JavaScript events, which would cause a lot of network traffic if every event would be sent to the server. Moreover, the area of the mouse move is of interest and not the exact route and all mouse positions. Therefore, these events need to be preprocessed and aggregated (e.g. all mouse-move or scrolling events within a small period of time are combined to a single one). In order to achieve this, the library developed along with the approach put forward in this paper supports the definition of custom events. They can also be used for decreasing the level of detail (e.g. if exact positions are not required), adding supplemental information (e.g. based on the event context) and for creating specialized monitors. Additionally, custom events are necessary, because not everything that is monitored on the client side directly causes a JavaScript event. In the case of selecting text (which needs to be monitored e.g. to identify text tracing), different events and document states have to be monitored to be able to tell, when a "text selected" event occurred. Combining this event with additional mouse or keyboard events can show if a piece of text has been copied for further processing. Furthermore, custom events may use their own triggers, e.g. timers for events of a temporal basis.

The data provided by JavaScript events may be both too detailed and/or insufficient. In order to determine parts of a page a user has been interacting with, exact mouse positions hardly serve, if the content is not bound to fixed positions or dimensions. Different font- or window sizes may change the absolute positions of content. In order to overcome these limitations the concept of page fragments is being introduced.

4.2 Monitoring Page Fragments

In order to be able to not only use a general "level of activity" within a page, but to treat parts of a page differently, events need to be mapped to semantic meta information, and/or their spatial location needs to be identified.

One approach is to analyze the text a user is currently interacting with. For example the event triggered on text selection contains information on the currently selected text, which may be used for keyword analysis. For other types of events, like mouse-over events, analyzing keywords may be less effective. These events are better suited to indicate activity within an area, than to assume the user is particularly interested in the words being hovered above. Furthermore, mapping events to keywords does not differentiate between parts of a page using the same keywords and makes it therefore difficult to tell what percentage of a page a user has read. Despite being useful for some types of events, exclusive use of this approach is not able to address some of the difficulties mentioned in previous sections.

Monitoring all HTML DOM elements and mapping events to them brings about problems as well. If a hypermedia document is highly structured in terms of markup (an element may contain a single letter), the result might be too detailed and consequently not useful. For some interaction types the exact position is only an approximation for the locus of attention. A user does not need to interact with every single element while reading it. Therefore, monitored areas must not be too small or detailed. On the other hand, within unstructured documents a single HTML element may contain the whole content, which results in treating the page as a single unit again.

Splitting hypermedia documents manually and providing semantic meta data for each section might be effective, but requires additional authoring effort and might therefore cause acceptance problems. The solution to be chosen should offer the possibility of manual sectioning, but provide its advantages also for unstructured contents.

The suggested solution tries to meet these demands by introducing "page fragments". By default, a page is vertically split into a number of k sections, each representing $1/k$ of the height of the document. Changing the width of a browser window may change the height of the document, but the height of the fragments will change as well. For $k = 5$ the third fragment will represent the center of the page; everything between 40% and 60% of the total height of the page. As this percentage will not change through resizing or using different font sizes (although the absolute postitions might change), this approach might be better suited for locating events. For some items, like images or spatial layout elements, resizing influences exact positioning more strongly; but nevertheless, the approximation should be sufficient.

One benefit of this approach is the independence of the structure of the content. Even unstructured documents may be segmented this way. For structured contents Therefore, no additional adaptive authoring effort is required.

The actual size of a fragment depends on the height of the document and on the value chosen for k. The higher the value for k, the higher the level of detail. However, if k is too high and/or there is little content on a page the height of one fragment might be less than the line height, which means that clicks on the top of a word are treated differently from clicks on the bottom of a word. Determination of optimal values of k for different circumstances is part of the ongoing work. As each fragment represents a fixed percentage of the page and each fragment is monitored separately, it becomes easier to tell how much of a page has been read; based on how many fragments are regarded as "having been read". Although the claim something has been read still remains an assumption, having gone through the whole page, spending a reasonable amount of time on all of its parts and interacting with them should be a stronger indicator for reading than simply requesting a document and possibly spending time on it.

Although using fragments instead of absolute positions to retrieve spatial information results in a loss of precision, this approximation should be sufficient for most applications, as e.g. mouse positions themselves are only an approximation for the locus of attention.

Within relatively static documents a fragment represents not only some percentage of the page, but also the actual content, which is important to determine interests. However, if parts of a page are adaptively included, the text within a fragment might shift, which requires a different technique to identify the actual content a user has been interacting with. Moreover, if semantic information is already available, it would be interesting to use it. Therefore, in addition to the predefined fragments, "custom fragments" can be defined, i.e. any HTML element can be specified as an additional fragment. Such custom fragments may be images (if they should be treated differently), sections automatically detected by means of topographical information (e.g. using headlines as separators) or elements that are already mapped to concepts or semantic meta information. When assembling hypermedia documents from multiple sources (e.g. due to conditional fragment inclusion), fragments can, during assembly, automatically be denoted and semantically characterized on the basis of their source, their "function" in the assembly, etc..

Generally, custom fragments are treated like predefined ones, so events are mapped to them as well (they can be mapped to more than one fragment). For instance, the time spent on images can be automatically calculated without requiring further modifications. Moreover, events can be defined for single fragments only, e.g. to specify the required reading time for each image separately (based on the actual content) and to trigger an event, if the time is exceeded and the image is regarded as visited.

5 Proposed Adaptations and Applications

Having monitored the user's interactions, the next step is to use this information within an AHS.

5.1 Using Existing Adaptation Techniques

The acquired data can be used directly within existing AHS, for instance, by replacing values in the user model. As an example, the values for "having read a page" or "knowing a concept" may be redefined within the user model by something more accurate, like "having spent at least x seconds on each page fragment". In this case, existing adaptive content might be used as is, while still profiting from more adequate data. Secondly, the level of certainty for attributes within existing user models may be increased. A third example on how to make use of the available information (using existing adaptation engines) is the definition of new and more granular adaptation rules like "probability of having read at least half of the page is greater than 80%". This helps authors of adaptive content to use the same authoring process as before, while still benefiting from user modeling based on client-side interaction monitoring.

5.2 Social Navigation Support

"Social navigation" has been defined as "moving 'towards' a cluster of other people, or selecting objects because others have been examining them"[11]. In

Knowledge Sea II [1] page visits and the time spent reading have been used to determine the most interesting documents.

Using the activity information on single page fragments, social navigation support can be provided not only on the level of documents and links, but also to highlight sections of a document. Fragments with a high number of interactions and reading time might be the most attractive and relevant ones. As users tend to follow highlighted links [12], highlighting these “popular” fragments might help readers to get a quick overview on the most interesting parts of a content.

Moreover, social navigation support may help authors to improve their contents by noticing, which parts of a document users found interesting. If within a block of important content users tended to spend time on the first lines only and skip the rest, the content might be too difficult, already known or not interesting enough. Important sections that tended to be skipped might need to be improved.

5.3 Identifying Learning Styles

One example where information on custom fragments opens new perspectives is the identification of learning styles. If images are treated as special fragments the system may identify the time spent on pictures and the interactions performed on them. These values may be compared to the average number of interactions / time spent on other parts of the page for the current user, which should allow for conclusions on the user's preferred learning style.

Having included this information into the user model, course material may be adapted accordingly. Contrary to traditional systems asking for learning style preferences, this approach might help to identify the learning style automatically by monitoring user interactions.

5.4 Retrieving Additional Information by Analyzing Usage Patterns

Perkowitz and Etzioni [13] have addressed the identification of usage patterns based on access logs. This approach can be extended by additional interaction data. If users stay on a page for a short time this might be because the content is too difficult, irrelevant or already known. Till now it has not been possible to tell the differences, because from a request-based point of view the users act in the same way. However, they might differ in their way of interacting with the system. A user not interested in the content might stay at the top of the page, read the headline and move on. Novice learners may start reading and scroll back again and give up before reaching the end of the page. Experts might have a quick glance at the whole page reading just the headlines and move on to an advanced topic. These are just some ideas on what might be found when analyzing interaction data. As there is more data available, new challenges arise concerning the analysis of this data. Finding patterns might be an important further step in improving AHS user models.

6 Ongoing Work and Future Perspective

A JavaScript library for asynchronous client-side user monitoring as described in section 4 has been developed. The implementation aims to be both easy to use and simple to extend with respect to client-side observation as well as server-side data processing. Existing courses can easily be provided with the core functionality of the library, while still allowing experts to add advanced features. Ongoing work on prototypes aims to provide and improve sample implementations according to the proposals in section 5. In order to provide the results of this work for a larger audience, the system is currently being integrated into a version of AHA! [14] embedded in the Sakai e-learning platform [15].

Evaluation work planned for the immediate future aims to determine an appropriate value k for the number of fragments a page will be divided into. It is presumed that in order to have comparable data, k should be constant within a system. However, for different types of systems a different k might fit best, depending on the actual content and its length. For example, in an AHS presenting slides (generally fitting in one screen), a lower k might be sufficient, in contrast to one providing e-books, where a page may contain a whole section or chapter. Two approaches will be used to determine the optimal k within specific contexts. The first one will start with a high amount of fragments. Values for neighboring fragments will be compared in order to find out, how many significantly different sections this number could be reduced to. A second approach would allow users to manually adjust the value for k, so that it subjectively results in the most appropriate user model.

Another challenge is to determine suitable weights for interaction types when combining different types of interactions to a "level of activity" for specific page fragments. A model capable of theoretically characterizing but also quantifying different types of activities will need to be selected and applied. The quantified "level of activity" resulting from said model will have to be verified with real end users' subjective perceptions of activity.

At a more general level, to evaluate the overall approach put forward in this paper, one would have to ascertain that: (a) it results in a more fine-grained and significantly more accurate user model and, (b) that this higher level of detail and the improved accuracy result in better adaptations.

For the tests, a version of AHA! based on traditional modeling techniques will be compared with an extended version analyzing client-side user interactions. To check the accuracy, a summative evaluation will be done through user studies. Students will work on their AHA! courses and will have to evaluate the two user models and decide which one they find more appropriate. Care will be taken to ensure that users do not just select the model presenting the values most favorable to their person (e.g. showing higher values for knowledge for a large number of concepts). Alternatively, only one model could be used, presented and evaluated and the average evaluation of both models would then be compared. Another scenario would be a self-evaluation of students and a comparison of these results to both models.

Acknowledgments

The work reported in this paper is funded by the "Adaptive Support for Collaborative E-Learning" (ASCOLLA) project, supported by the Austrian Science Fund (FWF; project number P20260-N15).

References

1. Farzan, R., Brusilovsky, P.: Social Navigation Support in E-Learning: What are the Real Footprints? In: Third Workshop on Intelligent Techniques for Web Personalization (ITWP '05). At 19th Int. Joint Conf. on Artificial Intelligence. (2005)
2. De Bra, P., Calvi, L.: AHA! An open Adaptive Hypermedia Architecture. The New Review of Hypermedia and Multimedia **4** (1998)
3. Hauger, D.: Fine-grained user models by means of asynchronous web technologies. In Hartmann, M., Krause, D., Nauerz, A., eds.: ABIS 2008 - Adaptivity and User Modeling in Interactive Systems, Würzburg, Germany (2008) 17–19
4. Brusilovsky, P.: Methods and techniques of adaptive hypermedia. User Modeling and User-Adapted Interaction **6**(2-3) (1996) 87–129
5. Hijikata, Y.: Implicit user profiling for on demand relevance feedback. In: IUI '04: Proc. of the 9th int. conf. on Intelligent user interfaces. (2004) 198–205
6. Putzinger, A.: Towards asynchronous adaptive hypermedia: An unobtrusive generic help system. In Hinneburg, A., ed.: LWA 2007: Lernen - Wissen - Adaption, Halle, Germany (September 2007) 383–388
7. Goecks, J., Shavlik, J.W.: Learning Users' Interests by Unobtrusively Observing Their Normal Behavior. In: Int. Conf. on Intelligent User Interfaces - Proceedings of the 5th int. conf. on Intelligent user interfaces. (2000) 129–132
8. Hofmann, K., Reed, C., Holz, H.: Unobtrusive Data Collection for Web-Based Social Navigation. In: Workshop on the Social Navigation and Community based Adaptation Technologies. (2006)
9. Stephanidis, C., Paramythis, A., Karagiannidis, C., Savidis, A.: Supporting interface adaptation: The avanti web-browser. In Stephanidis, C., Carbonell, N., eds.: Proceedings of the 3rd ERCIM Workshop "User Interfaces for All". (1997)
10. Claypool, M., Le, P., Wased, M., Brown, D.: Implicit interest indicators. In: Intelligent User Interfaces. (2001) 33–40
11. Dourish, P., Chalmers, M.: Running out of space: Models of information navigation. Proceedings of HCI'94 (1994)
12. Farzan, R., Brusilovsky, P.: AnnotatEd: A social navigation and annotation service for web-based educational resources. New Review in Hypermedia and Multimedia **14**(1) (January 2008) 3–32
13. Perkowitz, M., Etzioni, O.: Adaptive web sites: Concept and case study. Artificial Intelligence **118**(1) (2001)
14. De Bra, P., Ruiter, J.P.: AHA! Adaptive Hypermedia for All. In: Proceedings of the WebNet Conference. (2001) 262–268
15. Sakai: collaboration and learning for educators, by educators, free and open source. http://sakaiproject.org (2009)

Customized Edit Interfaces for Wikis via Semantic Annotations

Angelo Di Iorio[1], Silvia Duca[1], Alberto Musetti[1], Silvia Righini[1], Davide Rossi[1], Fabio Vitali[1]

[1] Department of Computer Science, University of Bologna, Mura Anteo Zamboni 7, 40127 Bologna, Italy

{ diiorio@cs.unibo.it, silvia.duca@unibo.it, musetti@cs.unibo.it, righini@cs.unibo.it, rossi@cs.unibo.it, fabio@cs.unibo.it}

Abstract.
Authoring support for semantic annotations represent the wiki way of the Semantic Web, ultimately leading to the wiki version of the Semantic Web's eternal dilemma: why should authors correctly annotate their content? The obvious solution is to make the ratio between the needed effort and the acquired advantages as small as possible. Two are, at least, the specificities that set wikis apart from other Web-accessible content in this respect: social aspects (wikis are often the expression of a community) and technical issues (wikis are edited "on-line"). Being related to a community, wikis are intrinsically associated to the model of knowledge of that community, making the relation between wiki content and ontologies the result of a natural process. Being edited on-line, wikis can benefit from a synergy of Web technologies that support all the information sharing process, from authoring to delivery.
In this paper we present an approach to reduce the authoring effort by providing ontology-based tools to integrate models of knowledge with authoring-support technologies, using a functional approach to content fragment creation that plays nicely with the "wiki way" of managing information.

Keywords: mediawiki, metadata, editor, template, semantic, Web 2.0

1. Introduction

As a first approximation, editing a wiki page is a very easy thing to do: we find the page that needs new content and click on the “Edit” button; either a WYSIWYG editor or a text box (if we are unlucky) are shown and off we go with writing down our stream of content. Then we click on the “Save” button and the page is shown updated.
This approach works perfectly as long as the content of the page is made of free-flow text and no structured data, so that the act of writing it resembles as much as possible the kind of writings possible on traditional desktop word processors.

Wiki content is often expected[1] to assume a more precise organization: e.g. a given structure, the presence of one or more predetermined elements, a specific sequence of parts, a given hierarchical organization, etc.. Some wiki applications have introduced the concept of template to help the page author to remember all the appropriate parts and the expected structure of the page. These templates are usually associated to page categories (chosen by the content author at the beginning of the editing process) and can be classified in two categories, called *creational* and *functional* [3]:

1. creational templates create new whole pages as a copy of an existing page, adding expected fragments (e.g., section headings) and boilerplate text to be substituted by the content author (e.g.: the words "insert description of page here"),
2. functional templates create a parameterized environment where named procedures can be invoked with actual parameter values to generate a completely different content fragment, such as infoboxes in Mediawiki[2] that use a functional syntax in the edit page to generate a structured name-value table in the rendered page.

In previous papers [2] and [7] we have shown how light constraints and user-defined rule-based annotations can help authors to verify the content correctness of a page, after the editing action. Such a process does not impose an excessive burden in actions, requirements and controls that would spoil or ruin the easy-going and ultimately successful "wiki way" of editing content [1].

In this paper we examine how content authors can be further helped in generating structured content in a manner that is compatible, again, with the aforementioned "wiki way" of editing.

Our approach is based on Semantic Web technologies to automatically generate either forms or templates. These interfaces are expressively equivalent but structurally independent of the expected content of the rendered page. Users do not deal with wiki code or complex syntaxes, rather they manipulate consistent (semantic) data in a transparent way.

More important, such an automatic process produces customized interfaces that can be easily tailored for each user or class of users. Authors use their own interfaces and can select, organize and customize their content with little effort.

The core of our proposal is the exploitation of ontologies for describing both customized interfaces and content. Ontologies are used to provide an abstract description of the concepts underlying the wiki page, and yet are separated from the details of the templates and forms, described through a separate ontology.

In this paper we introduce two separate but complementary tools, called Gaffeform and TinPP, which can be used to enrich MediaWiki with ontology-based functionality for the editing and presentation of structured data.

---

1 Throughout the paper we will often use the term "expect" whenever we are tempted to use (or some readers would not be surprised to find) the term "require" to refer to the behavior of wiki authors and readers. This is to stress the fact that the full adoption of the *wiki way*, i.e., the philosophy of use and content generation introduced by wiki software, is generally speaking not compatible with impositions, rules and constraints on final users, and adherence to such rules should always be considered as voluntary and optional rather than imposed by the software.

2 http://www.mediawiki.org/

Gaffeform uses a three-ontology model to generate arbitrary and customized forms for the editing of classes and class properties within MediaWiki. TinPP uses the same three ontologies to generate MediaWiki templates that can be used for rendering properties within the wiki page and along with the free-flow text of the same page. Together, they can be used to generate custom editing and rendering interfaces for MediaWiki functional templates. The end result is to generate structured content without ever forcing the user to adopt a given template nor to learn how to edit MediaWiki templates.
The paper is structured as follows: in section 2 we describe existing approaches at using semantic Web technologies in wikis. In section 3 we discuss the issues connected to generating custom interfaces through a three-ontology model, and in section 4 we introduce Gaffeform and TinPP, the tools generated to showcase our approach.

2. Related works

Combining wikis and semantic technologies is not a new idea in the literature. Semantic wikis are enhanced wikis that allow users to decorate pages with semantic data and to create a shared knowledge-base. Semantic wikis provide users with sophisticated searching and analysis facilities, keeping the open editing philosophy that characterizes traditional wikis.
Some semantic wikis allow users to load a predefined ontology in the wiki platform and automatically create pre-populated pages, that users can further modify. Makna [4] adopts such an approach and produces advanced wiki pages for the manipulation of the concepts of the ontology. While Makna is a complete wiki clone, WikiFactory[8] is a server-side application for the automatic transformation of OWL ontologies into wiki pages, deployable on multiple wiki platforms. Both these projects rely on the strong distinction between roles: the content-author knows the wiki domain but does not have any technical skills, while an ontology expert masters OWL-related tools to map ontology concepts into wiki pages. The system is in charge of translating such an abstract description into actual pages.
Kawawiki [5, 6] extends such an approach providing to the users an editing environment to customize and configure the final pages. Kawawiki takes an input RDF templates – a simplification of the RDF full language – describing both domain concepts and wiki instances. More important, Kawawiki automatically produces forms for populating the wiki pages after their initial deployment. These forms are described in the input RDF templates file and can be further customized by users. However their customization requires users to master a quite complex syntax.
Similarly, SemanticForms [11] *automatically* generates forms for letting users insert structured data in wiki pages. Semantic Forms is an extension of MediaWiki, as the tools presented in this paper. Semantic Forms does not take as input an OWL/RDF file but a MediaWiki template source code. It works on the top of structured content, that could also be created through automatic processes, to provide users a simplified interface for the authoring and customization of content.

WikiTemplate [10] is a closely related project, that allows users to access a page in two modes: when a page is viewed it is formatted according to its view template; when it is edited a set of editable text area is supplied, each corresponding to an area of the template. Common users can freely edit each field of that form, while *tailors* can combine different pieces into complex forms that better match both the structure of the content and the practices of the community.
The idea of combining atomic assets to generate customized interfaces is actually rooted in the early days of the Computer Supported Cooperative Work research. ObjectsLens [9] and its predecessor Information Lens [12] aimed at providing users with an environment for building their own collaborative applications. They adopted a modular approach: users could customize and combine atomic building blocks (forms, input fields, tables, messages) without following pre-constrained paths. The systems were, in fact, characterized as semi-formal systems, "processing some information in formally specified ways and other information in ways that are not formally specified". A very similar approach will be later adopted by the aforementioned wiki-related tools to provide users simplified authoring interfaces, without twisting and forcing the (semantic) structure of the content.

3. Generating Editing Interfaces from OWL ontologies

The creation of semantic content for wikis is still an open issue. Even more difficult is the creation of personalized workflows that allow users to create content on the basis of their preferences **and skills**. In fact, most of the applications described in the previous section assume that users know how to master languages like OWL and/or tools like Protégé [13]. That solution allows designers to create powerful semantic wikis but it does not solve issues related to authoring and usability.
This paper proposes a different approach, aiming at simplify the production of structured data by providing users with *customized interfaces*. The key idea is to let wiki administrators personalize forms and templates through a very simple and dynamic interface and to automatically produce end-users 'tools' that will actually allow authors to insert new semantic data.
The first relevant aspect of our approach is the strong distinction between roles involved in the creation of structured content. In particular, we identified three different actors with very different skills and very different tasks to complete:

1. **Anna:** the final wiki author. She writes the content and add metadata. She knows the topic of the wiki and the content of the pages. On the other hand, she does not have any technical expertise on semantic technologies or Web languages.
2. **Andrea:** the interface and usability expert. He knows how to map domain concepts to interface widgets and organize forms and templates. On the other hand, he does not have any knowledge of semantic technologies. Moreover, his tasks are independent from the actual content of the wiki.

3. **Pino:** the semantic expert. He knows tools and technologies related to the Semantic Web. He is not, however, an expert of the content domain, nor expert of Web technologies.

Note that the same user can play different roles at different times. In fact, each role represents a given category of skills (and goals) rather than a physical person accessing the wiki.

The contribution of this work is the design and implementation of a framework for simplifying the authoring and customization process of semantic data *by exploiting different capabilities of these roles*. The architecture and workflow we propose is shown in figure 1.

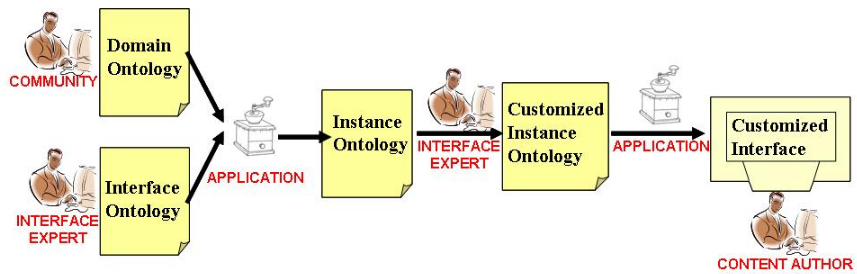


Figure 1: the overall architecture.

The framework takes as input two different ontologies - respectively called *Domain Ontology* and *Interface Ontology* – and automatically produces an intermediate ontology, the *Instance ontology,* describing a basic interface to handle structured data for the input domain. This ontology is transformed into a *Customized instance ontology*, that is finally instantiated into the actual interface shown to the users. Let us briefly discuss each of these ontologies, along with the process for creating them and their usage.

The *Domain Ontology* is actually created by Pino, reflecting ideas and comments of the users about the domain the wiki is related to. This ontology will finally describe properties of each page and relations among them, as domain entities will be mapped into actual wiki entities. There is no pre-defined structure for this ontology, that can be used to model any domain.

The *Interface Ontology* describes graphical interfaces. In particular, it describes objects like text areas, text fields, buttons or checkboxes that can be composed into complex interfaces. Constraints and relations among these objects can also be expressed with the ontology, as well as simple static interfaces. This ontology is created by Andrea though a graphic environment or with the help of Pino, the expert of semantic languages and tools. Basically, such an ontology describes the capabilities of the system and the potential interfaces users can use.

The *Instance Ontology* is automatically created by an application that merges the domain and interfaces ontologies. Domain-related entities are transformed into interfaces widgets that allow users to insert actual data. These widgets might be simple text areas whose content is written in the wiki template syntax, as well as sophisticated and dynamic tools.

Although such a description is enough to automatically generate the final interface, expert users (Andrea) can further customize it. This process is manual, possibly though ad-hoc tools. What is important is that the complexity of managing ontologies is hidden to Andrea, that only masters high level concepts related to interfaces and forms. The result of such a customization is a new ontology, just called *customized instance ontology*, describing the final refined interface.
The last step of the process is completely automatic and consists of mapping the abstract description into an actual template editing interface. Such an instantiation is performed by the system aggregating atomic widgets initially described in the interface ontology. One of the most important aspects of this approach is its *modularity*. Each actor - a human user or a system module – has to perform a specific and well-defined task. In particular, content authors are not required to master semantic tools or Web technologies: their input is filtered into ontology descriptions instantiated *automatically* in the wiki. A second advantage is the independence between the interface to modify the content and the content itself: multiple interfaces, in fact, can be created from the same domain ontology. Different users with different skills and preferences can then customize their interfaces to manipulate shared knowledge-bases. The declarative approach is a final important aspect: modifications to the interface are applied by changing their abstract description and translated into the final wiki without writing any line of code. Changes and updates are automatically propagated without any further intervention of the developers. All these points contribute to make such an approach a good solution to simplify the authoring and customization process of wiki templates, without burdening users with complex structures and tasks. In fact, the complexity of the architecture exists but it is hidden to the final users that only work according to their own skills.

4. From model to implementation: Gaffeform and TinPP

Two prototypes, called Gaffeform and TinPP, have been implemented following the aforementioned ontology-based approach. They are both presented in this section. Gaffeform is an Ajax-based editing environment for customizing forms within Semantic Media Wiki[3]. TinPP is a Java application for automatic generation of wiki templates, that provided preliminary studies and results for the overall Gaffeform framework.

4.1. TinPP

TinPP is a Java library, developed as a MediaWiki extension, for automatic generation of templates from OWL ontologies. The basic idea is to automatically map ontology *classes and properties* into wiki *categories and fields* of MediaWiki templates. In terms of the general architecture described in the previous section, TinPP takes as input a Domain Ontology, created by Pino on the top of multiple inputs from the domain experts, and a very simple Interface Ontology that describes

3 http://semantic-mediawiki.org/

the final interface as a plain wiki textarea, supposed to be manually modified by the content authors. Even the customization process – the generation of the Customized Interface Ontology – is very simple as it only consists of changing the order of fields in the template.

In fact, the focus of TinPP was not to provide users a sophisticated customization environment such as the complete Gaffeform infrastructure. Rather, to test the automatic generation of 'wiki data' from OWL ontologies and the extensibility of our approach. The results were very promising and allowed us to create a library later integrated in the overall framework.

The actual output is a wiki template, written in the MediaWiki syntax. The TinPP module saves the template directly in the wiki installation, so that the new template is available to any user about to start a new page.

Another interesting aspect of TinPP is the support for both *forced* and *unforced* template editing. The template creation process can be configured to force a choice between available categories and, ultimately, ontological classes. Or, it can be left to the community the possibility of suggesting specific templates or not using templates at all.

4.2. Gaffeform

Gaffeform is an integrated system that implements the general architecture described in section 3. It provides users with a dynamic environment to create structured wiki content in a simple and personalized way. In particular, it allows users to create forms from OWL ontologies, to customize these forms and to insert actual data though these customized and user-friendly interfaces.

The application is composed of three modules, delivered with different technologies:

- a *MediaWiki extension*: a module integrated in MediaWiki, and written in PHP, that adds Gaffeform functionalities to pages and interfaces of the basic installation;
- an *Ontology manager*: a module that processes OWL ontologies to produce final forms. Actually, this manager uses internally the Jena OWL processor. It is a Web-service, written in Java, that provides onotology-related tools to the MediaWiki extension;
- an *Ajax-based interface*: a client-side module that allows users to actually insert data through the forms generated by the Gaffeform engine.

The workflow integrating these three modules follows the schema described so far. The MediaWiki extension provides users an interface to upload the Domain Ontology and the Interface Ontology: these two operations are actually presented in different panels and are available to different users.

The input ontologies are automatically transformed into a simple and general form, by aggregating some widgets pre-loaded in the platform.

The first time a user edits a page, the MediaWiki extension shows such a basic form. The author can then organize a new form adding dynamic behaviors, moving buttons, changing fields order and so on. Modifying the structure and the parameters of the ontologies, the author is able to customize and change any detail of the interface. Figure 2 shows a basic form and a possible personalization.

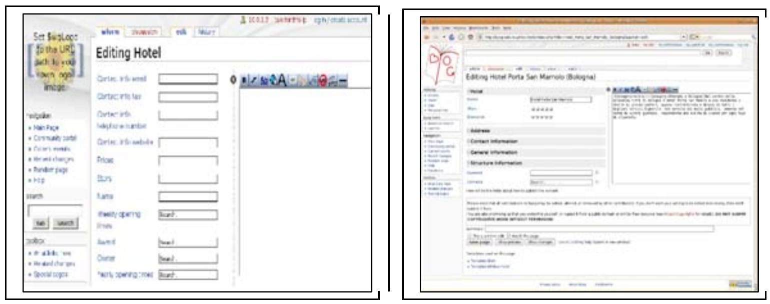


Figure 2: a general and personalized forms produced by Gaffeform

The current implementation does not provide users any simple interface to modify the ontologies but it still require the manual modification of these files. We just started to implement this new module that completes the architecture.

What is important is that any form is generated *automatically* by Gaffeform, integrating Javascript widgets and form objects described in the Interface Ontology. There is no manual generation of code but users adopts a declarative approach to aggregate new forms.

The forms produced by Gaffeform allow final users to produce both plain wiki templates and semantic data – in line with the domain ontology - to be used for advanced searching and analysis. Figure 3 shows a final page with a template (infobox in the right side of the page), whose data were submitted through the forms shown in figure 2.

Figure 3: The final wiki page with data inserted though the customized form

The current implementation of Gaffeform is a Web 2.0 application that simplifies the creation of MediaWiki templates. On the other hand, it still requires a manual intervention on the Interface Ontology. The next step of our research will be the implementation of a graphic tool to customize forms in a simple and fast way, using widgets that will be integrated in the Interface Ontology as well.

A systematic users evaluation of Gaffeform is another very important item in our agenda. Preliminary results showed us that customized and automatically-generated interfaces help users in creating semantic content. We plan to perform larger experiments and evaluations, and to study different trends for different domains and classes of users.

5. Conclusions

In this paper we have presented two separated technologies to help content authors in generating structured content in wiki pages in an easy way through the use of fully customizable form interfaces based on Semantic Web technologies. Although the application of semantic wiki technologies in the field of wikis is not new, the idea of associating them to template editing and to the customization of the interface for the editing of structured information is unique and represents a relevant innovation to the field of interface adaptation. We believe that both Gaffeform and TinPP represent useful approaches for extending the ease of use of wiki editing to structured content as well, and custom interfaces through the use of a three layer ontology can be exploited in a number of different directions, too.

References

1. Cunningham, W. and Leuf, B.. The Wiki way . Addison-Wesley, New York, 2001.
2. Di Iorio A., Zacchiroli S. "Constrained Wiki: an Oxymoron?". In the Proceedings of the ACM Symposyum on Wikis 2006, ACM Press, Odense, Denmark, 2006, pp. 89-98.
3. Di Iorio A., Vitali, F., Zacchiroli S., "Wiki Content Templating" in the Proceedings of the World Wide Web Conference 2008, Beijing, China, 2008.
4. Dello, K., Paslaru, E.. Creating and using semantic web information with makna. Proceedings of the First Workshop on Semantic Wikis, Max Vòlkel and Sebastian Schaffert Editors, 2006.
5. Kawamoto, K., and Kitamura, Y., Tijerino, Y. Kawawiki: A semantic wiki based on RDF template. Proceedings of the 2006 IEEE/WIC/ACM international conference on Web Intelligence and Intelligent Agent Technology, Hong Kong, China - 18-22 Decembe 2006.
6. Kawamoto, K., and Kitamura, Y., Tijerino, Y. Kawawiki: A template-based semanticwiki where end and expert users collaborate. Proceeding of the 5th International Semantic Web Conference, Athens, GA, USA - 5-9 November 2006.
7. Di Iorio A., Rossi D., Vitali F. and Zacchiroli S. "Where are your Manners? Sharing Best Community Practices in the Web 2.0". In Proceedings of ACM SAC 2009 (the 24th Annual ACM Symposium on Applied Computing), Track on Web Technologies.
8. Di Iorio, A., Presutti, V. and Vitali, F. WikiFactory: an ontology-based application to deploy domain-oriented wikis. In Proceedings of the European Semantic Web Conference, Budva, Montenegro, 2006.
9. Lai, K.-Y., Malone, T. W., and Yu, K.-C. 1988. Object lens: a spreadsheet for cooperative work. ACM Transactions on Information Systems 6, 4, 332{353).
10. Haake, A., Lukosch, S., and Schummer, T. 2005. Wiki-templates: adding structure support to wikis on demand. In WikiSym '05: Proceedings of the 2005 international symposium on Wikis. ACM Press, New York, NY, USA, 41-51.
11. Koren, Y. 2008. Semantic forms: Creating Forms for MediaWiki. http://www.mediawiki.org/wiki/Extension:Semantic_Forms
12. Malone, T. W., Grant, K. R., and Turbak, F. A. 1986. The information lens: an intelligent system for information sharing in organizations. In CHI '86: Proceedings of the SIGCHI conference on Human factors in computing systems. ACM, New York, NY, USA, 1(8).
13. Stanford Center for Biomedical Informatics Research, What is Protégé?, (2007), http://protege.stanford.edu/overview/index.html

Visualising web server logs for a Web 1.0 audience using Web 2.0 technologies: eliciting attributes for recommendation and profiling systems

Ed de Quincey[1], Patty Kostkova[1] and David Farrell[1]

[1] City eHealth Research Centre (CeRC), City University, London, UK
{Ed.de.Quincey}@city.ac.uk

Abstract. Web server logs have been used via techniques such as user profiling and recommendation systems to improve user experience on websites. The data contained within server logs however has generally been inaccessible to non-technical stakeholders on website development projects due to the terminology and presentation used. We describe a process that uses visualisation to enable these stakeholders to identify questions about site usage including user profiling and behaviour. The development of this tool utilising Web 2.0 technologies is described as well as feedback from the first stage of user evaluation on a real-world multi-national web development project called e-Bug. The potential for this process to elicit user attributes and behaviour that can be incorporated into automated user profiling systems is also discussed.

Keywords: Visualisation, Web Server Logs, User Profiling, Web 2.0 Technologies

1 Introduction

Research into online user behaviour has been aided by the relative ease of collecting feedback data using implicit methods such as web server logs [1, 2, 3], compared to explicit methods such as usability testing [4, 5], tagging [6] and ratings [7]. The data stored in server logs has been used to create a number of recommendation [8, 9, 10] [11, 12] and profiling systems [13, 14].

This has had a dramatic impact on the user experience e.g. Amazon [15] but apart from deliberate or accidental releases of server log data (e.g. NetFlix Prize[1], AOL), the information contained within the logs has been generally hidden from the users of a website and more importantly from non-technical stakeholders of a web development project. This means that few people outside of the server log analysis or web development communities fully understand the information that is stored in web logs and the user behaviour that it can explain.

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

There have been several commercial attempts (Google Analytics[2], Sawmill[3], WebTrans[4]) that have tried to make server logs, and therefore user behaviour, more accessible to site owners. However, these applications analyse generic features of sites that do not answer specific questions that certain stakeholders will have and do not help them identify trends in user behaviour due to the sheer volume and technical nature of the information presented [16].

A potential solution to this problem is to use techniques from the field of Software Visualisation (SV) to make the data contained within server logs more accessible to non-technical stakeholders in website development projects. Using these methods utilises the innate pattern matching ability [17] of the human cognitive system to identify trends in user behaviour which might be missed by the current automated profiling and recommendation systems. Once identified, non-technical stakeholders, such as content providers, can adapt content and the site design to fit user behaviour [16]. This human expertise could then potentially be integrated into current automated recommendation and profiling systems.

This paper describes the process of developing and using visualisation techniques to disseminate site usage information to non-technical stakeholders, in order to identify potential attributes for user profiling and recommendation systems. An ongoing multinational project in e-Health, called e-Bug (www.e-bug.eu), has been used as a test-bed and feedback from project stakeholders is detailed. The future possibilities of this technique are discussed as well as general implementation issues from using Web 2.0 technologies.

2 Background Information

2.1 Visualisation and Metaphors

Visualisation is concerned with making large amounts of information more comprehensible for the user by using a visual representation. Software Visualisation has been successfully used by software engineers to “make software more visible” [18] by representing the significant features of code using a visual metaphor. A well known example of a visualisation is the London Underground Tube Map[5] which is a representation of a complex, real world artifact that can be understood immediately and navigated simply. A detailed taxonomy of SV has been produced by Brice et al. [19] and also the related fields of Information Visualisation, Visual Analytics [20][21] and Metaphors used in interface design [22] contain a number of related and relevant techniques.

[2] http://www.google.com/analytics/
[3] http://www.sawmill.net/
[4] http://www.webtrans.co.uk/
[5] http://www.tfl.gov.uk/gettingaround/1106.aspx

2.2 The e-Bug Project

e-Bug is a European Commission funded project that aims to reduce inappropriate antibiotic use and improve hygiene through improving the education of young people in seventeen participating countries. e-Bug combines traditional methods of classroom delivery with online, browser-based (Flash) games to teach a pupils in junior and senior schools about microbes, hand and respiratory hygiene, and antibiotics. Example lessons and media are available on the e-Bug website[6] alongside games that can be used alongside the pack or standalone [23].

Currently the server logs from the e-Bug project are analysed using a proprietary application called Sawmill. This produces standard reports that cover information such as visits, hits, content viewed, visitor demographics and systems and referrers. These reports are produced monthly and uploaded onto the e-Bug website[7]. It was found however that although the project partners expressed a high degree of interest in the website statistics during meetings, the format that the reports were currently in were not easily accessible to non-technical users. This was mainly due to the terminology used and the statistics presented not answering specific questions that the project partners had regarding the users of the site [D. Farrell 2009, pers. comm.]. It was decided therefore that the server logs from the e-Bug project website would make a suitable test-bed to use visualisation techniques to analyse and present the statistics in a way that reduced the confusion and elicited potential attributes for user profiling.

3 Method for server log visualisation

A User Centred Methodology (UCD) [24] was used to develop a prototype application that would visualise the statistics that were currently calculated by the Sawmill application e.g. visits during particular months/years, geolocations of visits.

Sketching has been used previously to create code visualisation software [25] and so the same approach was used initially to explore potential metaphors and representations that could be used. An example sketch is shown below in Figure 1.

[6] http://www.e-bug.eu
[7] http://www.e-bug.eu/ebug_secret.nsf/England-Project-General/eng_eng_p_wp_gn_stats

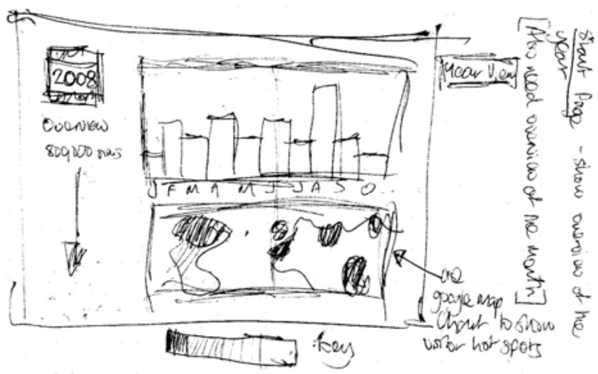


Fig. 1. Example sketch illustrating the weather map metaphor and bar charts

At this stage two potential metaphors were identified: a weather map metaphor and a timeline metaphor. After discussion with members of the project team it was decided to begin by developing the weather map metaphor as this would support one of the main features that was missing from the current reports: accurate geographical distribution of the users of the site.

3.1 Web 2.0 Technologies for Visualisation

Having identified possible interface designs for the application, an online prototype system was developed and suitable technologies explored for creating the map metaphor. The following figure shows the first version of the prototype.

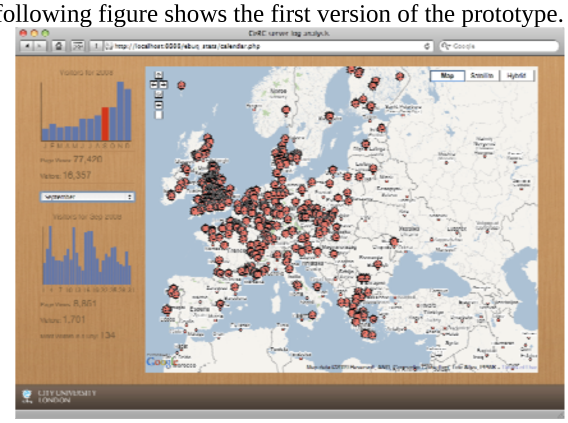


Fig. 2. Visualisation of visitors in September 2008 with each red icon representing a visitor

The interface incorporates two main visualisations. An area on the left hand side of the screen that shows the number of visitors and page views in a particular year and month (and their daily distribution) using simple bar charts. The area on the right contains a map with individual visitors denoted by their location with a marker (in this case the e-Bug logo). Users can select particular months from the drop down menu on the left and navigate the map using the navigation icons and the mouse pointer.

The map was created using the Google Maps API, which uses JavaScript to make asynchronous calls (AJAX) to display the map and the markers. The data for the markers is stored in an XML file that is generated by a PHP page parsing a CSV file that is created using Sawmill[8]. The CSV file contains paired values of a users' hostname and the number of page views that came from that IP address. PHP is then used along with the GEOIP Lite Open Source reverse geolocation database[9] to calculate a longitude and latitude for each hostname. These are then saved in an XML file in the following format:

```
<marker lat="40.6333" lng="-7.8333"/>
```

The bar charts were created using the Google Charts API, which creates dynamic images based on parameters passed in the querystring, for example:

```
<img
src=”http://chart.apis.google.com/chart?cht=p3&chd=t:60
,40&chs=250x100&chl=Hello|World” alt=”Bar chart” />
```

The parameters were determined using PHP pages and CSV files that contain monthly and daily totals of visits and page views.

4 Evaluation

This prototype was then uploaded to the e-Bug website and feedback was elicited from members of the e-Bug project team from seventeen European countries, as well as researchers involved in similar projects at UK Universities as part of the UCD process. The evaluation was in the form of an email with a set of open-ended questions that respondents were asked to answer regarding the interface. The main focus of this exercise was to ascertain whether the information that was being represented was clear enough, whether appropriate metaphors were being used and also whether there were any other statistics that users would be interested in. As this is an ongoing project, feedback has so far been received from nine respondents.

The majority of respondents reacted positively to the interface and the visualisation and a number of them were able to give detailed feedback, indicating that they were able to understand what the page was showing and what it did not. The main recurring points from this feedback are detailed below:

- Add representation that shows “magnitude of visitors” as it is difficult to gauge repeat visitors, number of pages viewed and markers that overlap.

---

[8] The data from Sawmill was used rather than the raw server logs due to the fact that Sawmill filters out certain web crawlers as well as using custom filters that have been created to remove certain IP addresses.

[9] http://www.maxmind.com/app/geolitecity

- Add specific place markers to the map that do not appear (unless at a higher zoom level).
- Add specific evaluation areas/overlays onto the map[10].
- Show the density of visitors in each area i.e. show visitors per 100,000 population to get more meaningful comparisons.
- Add in a view of popular pages downloads and where they originate from.
- Highlight returning visitors.
- Add in a view that shows the times of day that various pages are being accessed e.g. if the games are being viewed outside of school hours this could indicate that students are playing them at home.
- Ability to compare months and countries.

One of the most interesting points noted by the stakeholders however was the fact that the data being represented itself is a potential area of confusion. For example, a number of users gave the general impression that they did not know the difference between a visitor and a hit. It became clear that the target users of this application do not posses the same knowledge that experts in the field take for granted and further investigation into this area is being conducted.

Following on from this, a second version is currently being developed to take into account the feedback and also to tackle some of the issues that have been raised with regards to the interface and the information that users would like displayed. A screenshot(s) from the second iteration of the software is shown below:

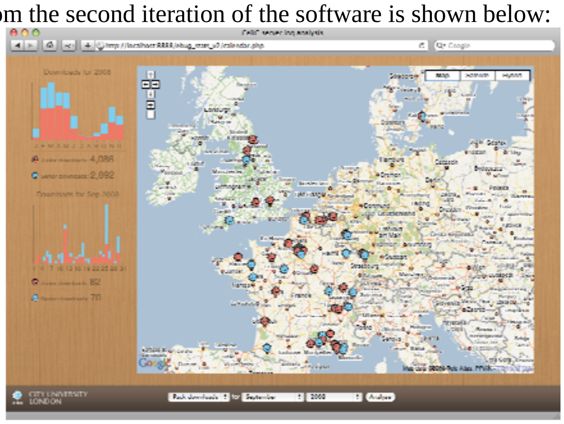


Fig. 3. Version 2 of the software visualises different types of file downloads, represented with two different colours

As well as markers and statistics for visitors, information regarding pack downloads (educational resources for teachers in Word and PowerPoint files) has been included and split into "Junior" and "Senior" versions.

This version of the application also uses an updated visitors' visualisation that takes into account the number of page views from a particular users. The well-known temperature scale visualisation used on weather maps has been utilised to be able to differentiate between the levels of activity in various regions.

[10] this can potentially be achieved using the Google Polylines' API

5 Discussion

5.1 Potential for use in User Profiling

Initial feedback has already indicated that visual representations of the data have allowed the non-technical stakeholders in the project to start to identify user types and user behaviour. One particular interest is whether pupils are accessing the games pages at home or at school and whether the tool can identify whether it is a student viewing the website or a teacher. By geographically representing visitors in relation to the location of target schools, along with the time they are accessing the site can potentially achieve this simple user profiling task.

This example and others detailed in Section 4 indicate that providing non-technical stake holders with a visual representation of the server logs has allowed them to communicate requirements for further analysis which can either be integrated into the filters used in the Sawmill application or into the visualisation tool. Without the use of visualisation techniques, it is doubtful that these questions regarding the users of the site and their behaviour would have been raised.

Further investigation of user profiles and understanding of national profiling differences is a subject of our ongoing research.

5.2 Strengths of Web 2.0 Technologies for Visualisation

There are a number of advantages with using Web 2.0 technologies such as the various Google API's and AJAX such as being able to create richer and more interactive online interfaces but the main advantage relates to being able to utilise users' pre-existing skills and experience. The majority of users have prior experience with interfaces such as Google Maps and in the same way that the Desktop has become the standard metaphor used for operating systems, maps and markers and the various methods of interaction that Google has developed have become a standard in this area. Being able to "piggy-back" on to that frees the user from the interface and allows them to focus on the visualisation, even though this application is a bespoke solution.

An associated advantage is that Google is a global organisation and so is its software. The potential users of this software are from a diverse set of countries with a number of different languages and levels of expertise. With Google being even more popular in Europe than the US [26], and its projected market share expected to take over the number one position from MapQuest by the end of the year [27], means that the chances of a user having had previous exposure to the Google Maps interface, and therefore the interface of this application, is quite high. This also has follow on advantages for issues such as localisation and internationalisation.

The other advantage is the increased speed in development. Being able to harness pre-existing API's allows for rapid prototyping and the ability to demonstrate a working concept to users to elicit feedback almost immediately and also allows for faster changes and incremental versions.

Finally, the fact that Web 2.0 technologies are also designed to be accessible via a number of different browsers and platforms also allows for speedier access and dissemination of the information which is vital for cross-nation projects such as e-Bug.

5.3 Limitations of Web 2.0 Technologies for Visualisation

One of the main problems with the Google Maps marker metaphor is the problem of occlusion, something that is common when using 3-D visualisations. If a user visits the site numerous times or downloads numerous pack or pages it is difficult to represent that with numerous markers on the map as they will overlap with one another. This can partially solved with the colour coding of markers but the accuracy of the geolocation database and the fact that numerous visitors can originate form the same area means that the markers often overlap. To improve this a method for clustering the markers so that close "neighbours" are represented by one marker and for this information to be presented textually once a user clicks on a clustered marker are being investigated.

A related problem is the amount of data that can be represented using these tools and the limitations of the browser. During testing of the application it was found that once around five thousand markers were placed on the screen using the standard Google method, the browser would slow down and become unusable. For this prototype this problem was solved by filtering out duplicate markers and also non-European hits (as this was not required at this stage in the site's development). However once the site it is launched and publicised further this year, there will be an increase in visitors and therefore an increase in markers. Clustering methods are therefore currently being investigated.

One final problem that was highlighted from user feedback was that relying on users having had prior experience on Google Maps means that for those who have not, or those who do not realise that this is a Google Maps interface, have initial problems with the interface. Adding extra methods of navigation or instructional video/instructions are currently being piloted.

6 Conclusion

The process of identifying appropriate visualisations to allow non-technical users to start to identify site usage from server logs is important for successful web site development and evaluation. The process presented in this paper has provided a number of insights into the potential of using Web 2.0 tools and metaphors for visualisation and dissemination of information. Although at an early stage, the tool is already providing insights into a number of usage patterns on the site which are enabling non-technical stakeholders of the e-Bug project to start to identify distinct user profiles and most importantly to start to be able to utilise the data stored in server logs more readily.

Future work will include an investigation into pre-existing taxonomies that exist of software visualisation [19] to see which might be relevant for representing web server log data and also which can be supported by Web 2.0 technologies. Also, current visualisation techniques from the biological sciences will be studied to see if any of these are appropriate e.g. spread of user activity being represented in a similar way to disease spread.

Following on from this, the tool will be used in an investigation into user behaviour on the e-Bug website in order to see whether researchers can identify usage trends visually and what are the attributes of these trends e.g. time of day a user visits plus geographical location might indicate whether they are a pupil or a teacher. This will then feed directly into the development and tailoring of content for the site and the potential for incorporating this into an automated profiling system will be investigated.

References

1. Dupret, G. E. and Piwowarski, B.: A user browsing model to predict search engine click data from past observations. In Proceedings of the 31st Annual international ACM SIGIR Conference on Research and Development in information Retrieval (Singapore, Singapore, July 20 - 24, 2008). SIGIR '08. ACM, New York, NY, 331-338 (2008)
2. Joachims, T.: Optimizing search engines using clickthrough data. In Proceedings of the Eighth ACM SIGKDD international Conference on Knowledge Discovery and Data Mining (Edmonton, Alberta, Canada, July 23 - 26, 2002). KDD '02. ACM, New York, NY, 133-142 (2002)
3. Joachims, T., Granka, L., Pan, B., Hembrooke, H., Radlinski, F., and Gay, G.: Evaluating the accuracy of implicit feedback from clicks and query reformulations in Web search. ACM Trans. Inf. Syst. 25, 2 (Apr. 2007), 7 (2007)
4. Nielsen, J.: Usability inspection methods. In Conference Companion on Human Factors in Computing Systems(Boston, Massachusetts, United States, April 24 - 28, 1994). C. Plaisant, Ed. CHI '94. ACM, New York, NY, 413-414 (1994)
5. Spool, J. and Schroeder, W.: Testing web sites: five users is nowhere near enough. In CHI '01 Extended Abstracts on Human Factors in Computing Systems (Seattle, Washington, March 31 - April 05, 2001). CHI '01. ACM, New York, NY, 285-286 (2001)
6. Kipp, M. E. and Campbell, D. G.: Patterns and inconsistencies in collaborative tagging systems: An examination of tagging practices. Available from http://dlist.sir.arizona.edu/1704/01/KippCampbellASIST.pdf (2006)
7. Anand, S. S., Kearney, P., and Shapcott, M.: Generating semantically enriched user profiles for Web personalization.ACM Trans. Interet Technol. 7, 4 (Oct. 2007), 22 (2007)
8. Schafer, J. B., Konstan, J., and Riedi, J.: Recommender systems in e-commerce. In Proceedings of the 1st ACM Conference on Electronic Commerce (Denver, Colorado, United States, November 03 - 05, 1999). EC '99. ACM, New York, NY, 158-166 (1999)
9. Schafer, J. B., Konstan, J. A. and Riedl, J.: Ecommerce recommendation application, Data Mining and Knowledge Discovery, 5(1/2):115–153 (2001)
10. Sarwar, B. M., Karypis, G., Konstan, J. A. and Riedl, J.T.: Item-based collaborative filtering recommendation algorithms, Proc. of the Tenth Int. WWW Conf., pp. 285–295 (2001)
11. Ali, K. and van Stam, W.: TiVo: making show recommendations using a distributed collaborative filtering architecture. In Proceedings of the Tenth ACM SIGKDD international Conference on Knowledge Discovery and Data Mining (Seattle, WA, USA, August 22 - 25, 2004). KDD '04. ACM Press, New York, NY, 394-401 (2004)

12. Herlocker, J. L., Konstan, J. A., Terveen, L. G., and Riedl, J. T. 2004. Evaluating collaborative filtering recommender systems. ACM Trans. Inf. Syst. 22, 1 (Jan. 2004), 5-53 (2004)
13. Kostkova, P., Diallo, G. and Jawaheer, G.: User Profiling for Semantic Browsing in Medical Digital Libraries, Proceedings of ESWC 2008, The Semantic Web: Research and Applications, 5th European Semantic Web Conference, Tenerife, Canary Islands, Spain, June 1-5 (2008)
14. Danilowicz, C. and Indyka-Piasecka, A.: Dynamic User Profiles Based on Boolean Formulas, Lecture Notes in Computer Science, Volume 3029/2004, pages 779-787 (2004)
15. Linden, G., Smith, B. and York, J.: Amazon.com Recommendations Item-to-Item Collaborative Filtering, IEEE Internet Computing, January-February (2003)
16. Drott, M. C.: Using Web server logs to improve site design. In Proceedings of the 16th Annual international Conference on Computer Documentation (Quebec, Quebec, Canada, September 24 - 26, 1998). SIGDOC '98. ACM, New York, NY, 43-50 (1998)
17. Sinha, P.: Recognizing complex patterns. Nat Neurosci 5:1093–1097 (2002)
18. Petre, M. and de Quincey, E.: A gentle overview of software visualisation. Computer Society of India Communications. August, 6-11. ISSN 0970-647X (2006)
19. Price, B. A. Baecker, R. M. and Small, I. S. A Principled Taxonomy of Software Visualization, Journal of Visual Languages and Computing, 4, 1993, 211—266 (1993)
20. Thomas, J.J. and Cook, K.A.: Illuminating the Path: The R&D Agenda for Visual Analytics. National Visualization and Analytics Center. p.3-33 (2005)
21. Tufte, E.R.: The Visual Display of Quantitative Information. Graphics Press (1983)
22. Erickson, T.: Working With Interface Metaphors, 65-73, edited by B. Laurel, The Art of Human-Computer Interface Design, AddisonWesley Publishing Company Inc, Reading, Massachusetts, USA. (1990)
23. Kostkova, P. and Farrell, D.: e-Bug online Games for Children: Educational Games Teaching Microbes, Hand and Respiratory Hygiene and Prudent Antibiotics Use in Schools across Europe, In the Proceedings of the ESCAIDE 2008 conference, Berlin, November 2008 (2008)
24. U.S. Department of Health & Human Services: Step-by-Step Usability Guide. Available: http://www.usability.gov/. Last accessed 4th February 2009. (2009)
25. Craft, B. & Cairns, P.: Using Sketching to Aid the Collaborative Design of Information Visualisation Software-A Case Study, In Human Work Interaction Design: Designing for Human Work, Springer Boston, Volume (221), p. 103—122 (2006)
26. Lipsman, A.: Google Holds Top Spot in European Site Rankings, According to comScore World Metrix. Available: http://www.comscore.com/press/release.asp?press=988. Last accessed 1st March 2009. (2006)
27. ABI Research: The Google Maps vs MapQuest Online Mapping Portal War Is Driving Map 2.0 Innovations. Available: http://tinyurl.com/dlwj7d. Last accessed 5th March 2009. (2009)

New Generation of Social Networks Based on Semantic Web Technologies: the Importance of Social Data Portability

Liana Razmerita[1], Martynas Jusevičius[2], Rokas Firantas[2]

Copenhagen Business School, Denmark[1]
IT University of Copenhagen, Copenhagen, Denmark[2]
liana.razmerita@cbs.dk, martynas@itu.dk, rokas@itu.dk

Abstract. This article investigates several well-known social network applications such as Last.fm, Flickr and identifies social data portability as one of the main technical issues that need to be addressed in the future. We argue that this issue can be addressed by building social networks as Semantic Web applications with FOAF, SIOC, and Linked Data technologies, and prove it by implementing a prototype application using Java and core Semantic Web standards. Furthermore, the developed prototype shows how features from semantic websites such as Freebase and DBpedia can be reused in social applications and lead to more relevant content and stronger social connections.

1 Introduction

Social networking sites are developing at a very fast pace, attracting more and more users. Their application domain can be music sharing, photos sharing, videos sharing, bookmarks sharing, professional networks, and others. Despite their tremendous success social networking websites have a number of limitations that are identified and discussed within this article. Most of the social networking applications are "walled websites" and the "online communities are like islands in a sea"[1]. Lack of interoperability between data in different social networks applications limits access to relevant content available on different social networking sites, and limits the integration and reuse of available data and information. This may result in a growing dissatisfaction of the user community and a reduced usability of the websites. More research combining social networks and Semantic web is required to address the above mentioned limitations. Research combining social networks and Semantic Web is an interdisciplinary field, atracting researchers from both social and information sciences. Current research is mostly related to extraction of semantic data from existing social applications, its representation and its analysis. For example, there is work done in extracting ontologies from user contributed folksonomies (collaborative tagging systems) and integrating ontologies together with folksonomies [2, 3]. Others propose approaches to development and evolution of lightweight ontologies in a similar collaborative way [4, 5]. Researchers seem to agree that folksonomies and (lightweight) ontologies share more common properties than differences and will be

further integrated, and thus community-based bottom-up development approach will prevail over top-down controlled engineering efforts.
Much of the current research for representing simple user profiles is based on the Friend of a Friend (FOAF) project[1] – a project aimed at "creating a Web of machine-readable pages describing people, the links between them and the things they create and do". FOAF is currently an important source of RDF data available on the Web which has already been used for social network analysis [6-8].
A related initiative is Semantically-Interlinked Online Communities (SIOC) project2, which provides an ontology for describing items and relationships from Internet discussion methods (such as blogs, forums, and mailing lists) to facilitate interconnection of these methods via publishing of metadata [9, 10]. Many recent papers show growing interest in portability issues among social network applications – they are being called "fundamental problems", and semantic technologies (mainly FOAF) are being proposed to solve them [11].

There is theoretical work done combining Semantic Web (SW) and social networks, especially in analysis of social networks and extraction of knowledge [12]. However, creation of new end-user semantic social applications as well as their design and implementation are not well explored. Existing social network applications do not employ SW technologies, although most of the standards infrastructure is already in place. Most of them are "walled" websites, which provide limited means for users and developers to control, publish, and access social data. This limits possibilities for reuse and integration, which are the driving forces behind Web 2.0 as well as Semantic Web, and results in growing dissatisfaction in the user community.
This article proves through the implementation of a prototype that Semantic Web technologies can be used to build a next generation of social networks that overcome limitations of social networks applications and enable new features currently not exploited by them.

2 Study of Social Networks

We analyzed applications which we personally use and which we think reflect the current state of the art in social networking: Last.fm, Flickr, Facebook, LinkedIn. These social networking applications have a number of technological limitations as summarized bellow:

- It is not possible to export/import profile data from one application into another
- It is not possible to export/import social relationships from one application into another
- There is usually less data available in machine-readable formats than the application contains
- Application Programming Interfaces (APIs) are based on variety of custom formats and protocols, some of them non-standard (such as FQL and FBML in Facebook)

1 http://www.foaf-project.org/
2 http://www.sioc-project.org/

Our observations fit well with statements by initiatives such as *Open Social Web*[3], *Social Network Portability*[4], *DataPortability*[5], *OpenID*[6], *OpenSocial*[7], born as a result of a growing dissatisfaction in user communities.

2.1 Semantic Social Networks

Freebase claims to be an open database of the world's information. It acquires structured data spanning different domains such as music, people, and locations from various sources such as Wikipedia and MusicBrainz[8]. The data is aggregated and identical or related concepts are linked together. In addition, users in the community can add, edit, and even upload data. Topics in Freebase are organized by types which are grouped into domains. An important feature is that users not only can fill already predefined types with instance data or edit it, but also create their own types and define their properties, i.e. they can create new schemas and extend Freebase's domain model using the same interface. However, it provides an open but proprietary API for its data and encourages its use in applications and mashups.

DBpedia is a community effort to extract structured information from Wikipedia and to make this information available on the Web. It provides an RDF dataset extracted from Wikipedia, which contains mostly free text but also structured information such as categories, lists, info boxes, links to external pages etc. DBPedia makes it possible to ask complex queries (such as "*German musicians who were born in Berlin*") over a SPARQL query interface. DBPedia is a prime example of Linked Data publishing and can be browsed using semantic browsers. It is interlinked with other semantic datasets such as Geonames[9], MusicBrainz etc.

3 Social network applications with semantic technologies

3.1 Creation of metadata

Folksonomies are the primary sources of metadata on Web 2.0, however they have issues with consistency, ambiguity, synonymity. A next step beyond Web 2.0 is Semantic Web. It has been observed how folksonomy tags evolve into *property*:*value* triple-tags, which serve the same purpose as *subject property object* triple statements in RDF. This phenomena has even been called "poor man's RDF" [13]. And thus

3 http://opensocialweb.org/
4 http://microformats.org/wiki/social-network-portability
5 http://www.dataportability.org/
6 http://openid.net/
7 http://code.google.com/apis/opensocial/
8 http://musicbrainz.org/
9 http://www.geonames.org/ontology/

folksonomies move towards becoming lightweight ontologies. Social networks will have to provide more sophisticated means to directly create RDF metadata, and collaborative tagging will evolve into lightweight ontology development and integrate into collaborative modeling of the social network domain.

3.2 User interface

Much work is left on the issue on how to present semantic data to the user in applications, not to mention editing it [14]. There exists a number of semantic browsers, such as Tabulator[10], Disco[11], OpenLink RDF Browser[12], Objectviewer[13], Zitgist[14]. They are able to render generic RDF data for human users and navigate through different data sources through RDF links, just as conventional Web browsers navigate through HTML links. However, this kind of presentation most likely too advanced for mainstream Web users (see Figure 1).

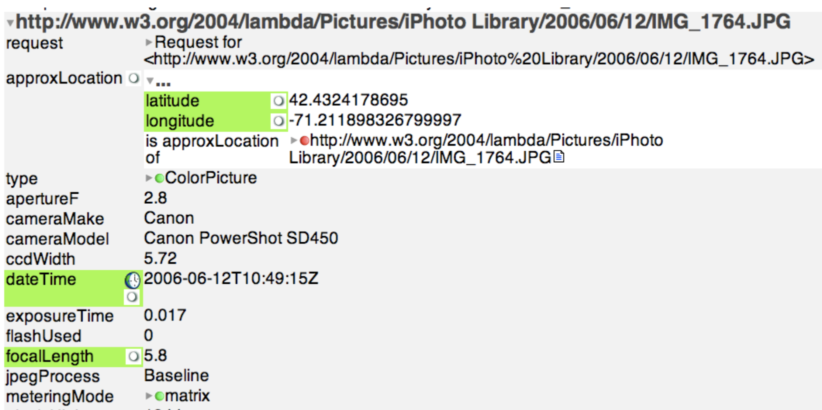


Figure 1 Tabulator view

We can assume that a Semantic Web application interface visualizes its domain ontology so that each class and instance would have its own page, linked to others through class-instance and instance-instance relationships. This approach, which we call *generic*, is used in many semantic websites, and probably best illustrated by Freebase. Another approach, which we call *specific*, is used by conventional Web applications, as well as social networks. Every type of information (such as a car, a user, or an event) has its own specific user interface. Each new type has to get a new interface, the same interface cannot be used for different types, and interfaces have to be fixed when the schema changes. This approach is obviously not feasible on the

10 http://www.w3.org/2005/ajar/tab
11 http://www4.wiwiss.fu-berlin.de/bizer/ng4j/disco/
12 http://demo.openlinksw.com/rdfbrowser/index.html
13 http://objectviewer.semwebcentral.org/
14 http://dataviewer.zitgist.com/

Semantic Web, where ontologies are meant to be extended, reused, and integrated from different sources. If social networks should become extensible semantic applications, it is likely that they should adopt this generic approach.

3.3 Domain model

Social network applications (Last.fm, Flickr, LinkedIn etc.) are usually developed for different application domains (music, photos, business). However, they share a common property: the domains are fixed and non-extensible. Users are encouraged to contribute and improve application data, but this is only limited to instance data for predefined types. Semantic applications such as Freebase take a different approach and allow users to edit the domain model itself: not only fill in instance data, but extend and edit types, add new ones, and define properties in the underlying ontology. Following this approach, social network applications would empower users to express their identities by creating or reusing concepts and relationships relevant to them, and share them with others. The domain model could be left to the community to control and to develop it in a direction which is currently of most interest to it, keeping it relevant over time. People would connect through things they have in common, achieving object-centered sociality [10]. This may be achieved in the future by integrating lightweight ontology development into the means of user collaboration and content contribution. To implement this approach, applications need to be modelled with semantic-enhanced languages such as: RDF/OWL, which offers more expressivity than object-oriented and relational models, is based on formal semantics and therefore interpreted unambiguously by different agents. Furthermore, they need to reuse FOAF and SIOC ontologies, which currently are state of the art representations of social networks on the Semantic Web, as well as other relevant ontologies.

Most of the current SW applications are also static and fixed in the sense that ontologies are known and mapped manually at design time [15]. Although semantic technologies are designed with extensibility and openness in mind, current programming languages and tools are not able to fully exploit it. It is expected that future semantic applications will be using multiple ontologies, discover them and integrate on request.

3.4 Publishing and reusing data and metadata

Large amounts of meaningfully interlinked RDF data available on the Web are crucial for achieving the Semantic Web vision. However, many social networks do not offer interfaces and APIs to access application data. Others make the contents of the website (such as lists of users, songs, or pictures) available via simple read-only REST interface in a software-processable data format, usually a custom schema of XML, Atom, or RSS. Some provide full APIs with add/update methods, invoked via various interfaces such as REST, XML-RPC, SOAP, Atom, or OpenSocial. A variety of publishing formats (especially non-standard) make reuse difficult. We argue that semantic social networks should publish their data in RDF, designed specifically for distributed knowledge representation. Furthermore, all resources in social networks (including non-information, "real-world" resources) should be given URIs, distinguished from URIs of representations that describe them, and published as Linked Data[15]. APIs should be replaced by SPARQL endpoints, which would allow running remote structured queries against application data. Semantic data representation and advanced interfaces would help to overcome portability issues of proprietary APIs and interconnect social networks with different data sources, enable use of semantic browsers, and facilitate semantic mashups.

4 Prototype application design and implementation

The application prototype is a social network that allows users to browse events (such as concerts and conferences) and places of interest (such as venues and hotels) and find those that are most relevant to them. The prototype features a generic user interface. Users are able to browse OWL ontology classes and their instances and see properties with values, as represented in Figure 2.

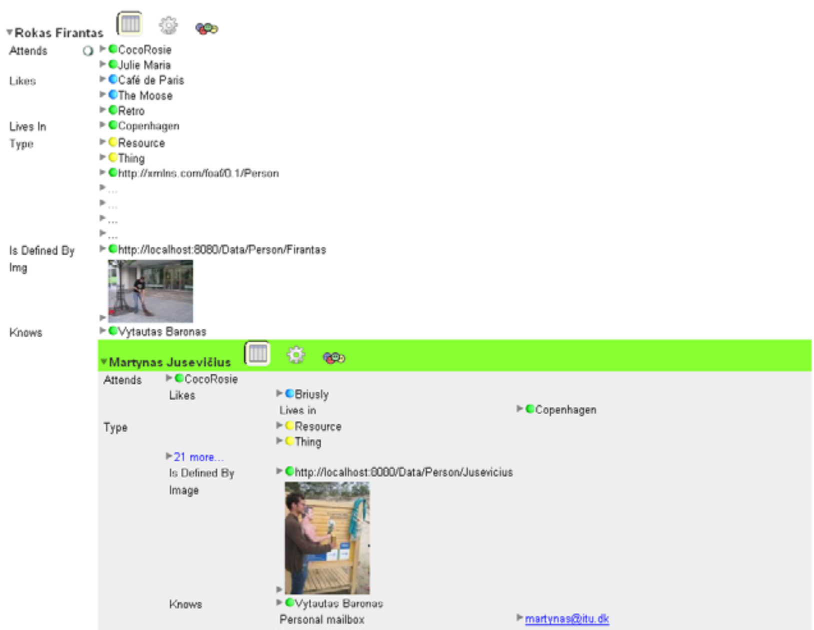


Figure 2 Browsing the ontology in a Tabulator view

The application is able to import a list of friends from an external FOAF source. It provides Linked Data access by serving interlinked RDF/XML which can be

15 http://linkeddata.org/

visualized and browsed using semantic browsers such as Tabulator or reused in other applications. It provides a SPARQL endpoint, which allows running structured queries. It also implements a semantic mashup: when a page of an instance is requested, the application queries remote DBPedia SPARQL endpoints, retrieves its description and homepage address in real-time and presents it to the user in the same fashion as local properties and values. Prototype is based on a RESTful Web framework which treats HTTP resources as first-class objects and follows a Model View Controller (MVC) pattern and W3C standards. Within the prototype the Model is the ontology layer, the Java code is the Controller and Views are generated by integrating SPARQL queries results and transforming them into XHTML using XSLT.The application domain is modelled as a RDF/OWL ontology, stored in a RDF triple store, accessed using Jena[16], and queried using SPARQL.

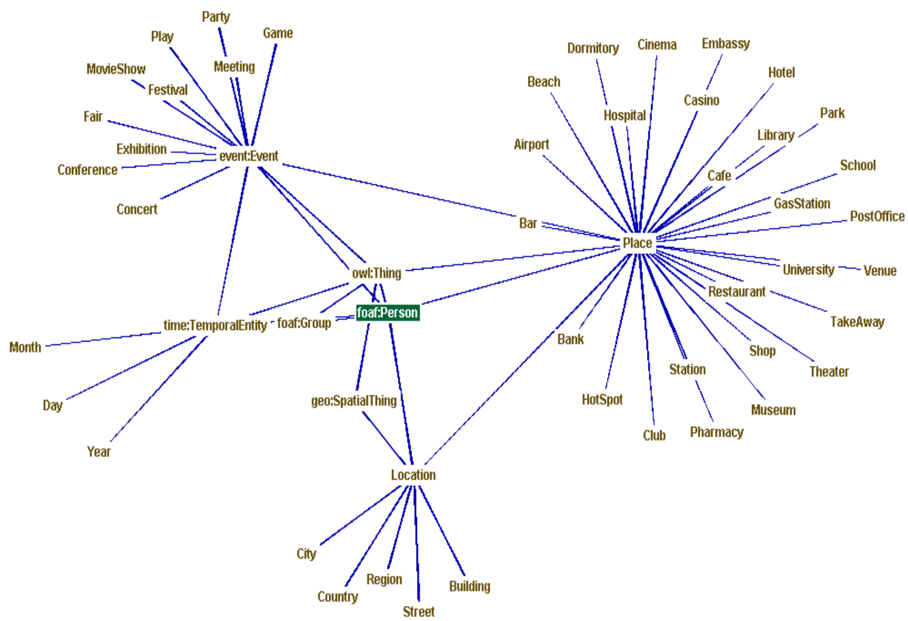


Figure 3 Domain ontology graphical view

The domain ontology is based on FOAF and SIOC. It is implemented in OWL and extends classes such as foaf:Person and adds a number of new classes such as: Place and Event as represented in Figure 3. FOAF and SIOC classes and properties were reused. Views become representations of REST resources (XHTML, RDF). They join several SPARQL XML results and transform them directly to output XHTML using XSLT, or serve raw RDF/XML for the Linked Data interface, depending on the HTTP *Accept* header. Controller dispatches requests to resources which have explicit

16 http://jena.sourceforge.net/

URIs, implement HTTP methods, can be related to domain instances using *foaf:topic*, and return view representations.

Most of the current object-oriented languages are statically typed and do not allow classes be changed or extended at run-time. I.e. it is not easy with the existing tools to map Event class in OWL to an Event class in Java so that it could be changed or extended at run-time. Tools such as RDFReactor[17] and Elmo[18] generate object-oriented Java code from our ontologies, but this code is static and not extensible at run time and therefore it was not used.

5 Conclusions and future work

Social data portability issues are leading to dissatisfaction in both user and developer communities. They are caused by limited amounts of social data published openly and lack of tools to import it, as well as formats and APIs of limited interoperability. Social networks would benefit from Semantic Web technologies. FOAF, SIOC, and Linked Data can solve portability issues and enable data reuse.

New generation of social applications could also take advantage of the advanced means to model data that SW technologies provide. Semantic data representation and advanced interfaces would help to overcome portability issues of proprietary APIs and interconnect social networks with different data sources, enable use of semantic browsers, and facilitate semantic mashups. Domain model could be collaboratively developed by users of the application. This approach requires a new generic user interface based on classes, instances, and properties. It could lead to more up-to-date and relevant content, which would in turn facilitate social connections through points of common interest. Other interesting directions that we have not yet pursued include AJAX-enabled application interface, a form-based interface for SPARQL, and dynamic, run-time object-ontology mapping tools.

References

[1] U. Bojars, J. Breslin, G.,, V. Peristeras, G. Tummarello, and S. Decker, "Interlinking the Social Web with Semantics," *IEEE Intelligent Systems,* vol. 23, pp. 29-40, 2008.
[2] L. Specia and E. Motta, "Integrating Folksonomies with the Semantic Web," in *Lecture Notes In Computer Science*, 2007, pp. 624-639.
[3] Z. Xu, Y. Fu, J. Mao, and D. Su, "Towards the semantic web: Collaborative tag suggestions," in *Collaborative Web Tagging Workshop at WWW 2006*, Edinburgh, UK, 2006.
[4] S. Angeletou, M. Sabou, L. Specia, and E. Motta, "Bridging the Gap Between Folksonomies and the Semantic Web: An Experience Report," in *ESWC 2007, Workshop on Bridging the Gap between Semantic Web and Web 2.0*, 2007.

17 http://ontoware.org/projects/rdfreactor/
18 http://www.openrdf.org/about.jsp

[5] P. Mika, "Ontologies are us: A unified model of social networks and semantics," *Journal of Web Semantics,* vol. 5, pp. 5-15, 2007.

[6] T. Finin, L. Ding, L. Zhou, and A. Joshi, "Social networking on the semantic web," *The Learning Organization,* vol. 12, pp. 418-435, 2005.

[7] L. Ding, L. Zhou, T. Finin, and A. Joshi, "How the Semantic Web is Being Used: An Analysis of FOAF Documents," 2005, pp. 113c-113c.

[8] J. C. Paolillo, S. Mercure, and E. Wright, "The social semantics of livejournal foaf: Structure and change from 2004 to 2005," 2005, pp. 69–80.

[9] J. G. Breslin, A. Harth, U. Bojars, and S. Decker, "Towards Semantically-Interlinked Online Communities," 2005, pp. 500–514.

[10] J. Breslin and S. Decker, "The Future of Social Networks on the Internet: The Need for Semantics," *IEEE Internet Computing magazine,* vol. 11 pp. 86-90, 2007.

[11] P. Mika, "Flink: Semantic Web technology for the extraction and analysis of social networks," *Web Semantics: Science, Services and Agents on the World Wide Web,* vol. 3, pp. 211-223, 2005.

[12] L. Ding, T. Finin, and A. Joshi, "Analyzing Social Networks on the Semantic Web?," *IEEE Intelligent Systems (Trends & Controversies),* vol. 8, 2004.

[13] B. Andersen, "Meta Tags: The Poor Man's RDF?," http://weblog.scifihifi.com/2005/08/05/meta-tags-the-poor-mans-rdf/, 2005.

[14] A. Ankolekar, M. Krötzsch, T. Tran, and D. Vrandecic, "The two cultures: Mashing up Web 2.0 and the Semantic Web," *Web Semantics: Science, Services and Agents on the World Wide Web,* vol. 6, pp. 70-75, 2008.

[15] L. Razmerita, A. Angehrn, and A. Maedche, "Ontology-based User Modeling for Knowledge Management Systems," in *UM* Pittsburgh, USA, 2003, pp. 213-217.

Balanced Recommenders: A hybrid approach to improve and extend the functionality of traditional Recommenders

Javier G. Recuenco[2], David Bueno[1]

[1]Departamento de Lenguajes y Ciencias de la Computación. Universidad de Málaga
[2]AbyPersonalize
javier.recuenco@abypersonalize.com, bueno@lcc.uma.es

Abstract. The authors present a possible approach for a new general purpose recommender architecture, one which complements the current proven and tested techniques (User Model, Collaborative Filtering, Content Based Filtering), used in some everyday business scenarios, balancing with newly developed personalization procedures and methodologies. The overall objective is to try to tackle some of the typical shortcomings of traditional recommender systems (Cold start, dilution of the "personal color" in a sea of collective thinking…), by effectively balancing the amount of collective intelligence used against a more "personal affinity" score. This, the authors call PPM (Product Profile Matching), an approach which ignores collective results and relies mainly on the intrinsic affinity between the nature of both the subject and the item. Hence the use of the name "balanced", because of the balance struck between A.I. techniques and Applied Personalization Techniques used to make a better recommendation. The authors also focus on the need for proper self-fulfilling techniques in order to illustrate the paramount importance of improving and extending the control that existing recommender systems give users in order to optimize the user experience. An example based on the author's previous work in the field of TV content recommenders is presented to illustrate the validity of our approach.

1 Introduction

Information overload has become a problem in recent times. Increasingly, system users encounter difficulties in finding the information they need. Recommender Systems [1] have emerged as a way to reduce the amount of information users have to process in order to find something interesting. They have been applied to different areas of knowledge such as personalized newspapers (newsdude) [2], movie recommenders (movielens) [3], personal electronic programming guides ((PTV) [4]) or art recommenders [5]. In the rest of this article, the base element of the recommender system will be referred to as an item. Items may be documents, songs, news, TV programs, goods in a shop, pictures etc…
There are two main techniques used by existing recommender systems: content-based recommendations and collaborative recommendations [6]. In the first case, user recommendations are based on items similar to those s/he may have chosen in the past. An example of this is METIORE [7], which recommends publications; or myTV project [8] which is related to TV programming. In the second case, users are

informed of recommendations based on similar users' preferences. A well-known example of this approach is Amazon.com [9] or Barnes&Noble both of which recommend books purchased by other clients with a similar profile. The Movielens recommender is also based on this technique. Ideally, the best solution benefits from both content and collaborative information. This is called the hybrid model and some interesting and relevant material can be found in [1] [10] [11].

Existing recommender systems have certain limitations, which although they do not hamper the overall usefulness of the system they prevent the "perfect recommendation" from being provided. The "perfect recommendation" is somewhat difficult to specify, but we define it as:

"*The result of ascertaining the exact desires of the individual using a recommender system, taking into account not only the knowledge of the whole network, but the particularities of the user AND the items available, which are relevant to the recommendation process*".

Some of the difficulties of recommenders are well known and are usually dealt with in different ways: "Cold start" is perhaps the best known one; clearly there is no real way for a recommender to provide useful recommendations from the start without an initial recommendation from other users. In Movielens different techniques are developed to select some items (films), shown to the user in order to create an initial model. One of the criteria to show initial items to the users is to rank items according to their particular relevance to these individuals. A good overview of the ranking algorithms is presented in [12] but most of these results are applied to queries made to documentary databases or to the Web like the popular PageRank ranking system [13]. We can also find ranking algorithms for blogs [14] to select the most popular ones according to the number of times they are read, the number of comments made and their voting average. Recent work [15] has tried to solve the cold start problem using the tied Boltzmann machine model, improved with content for collaborative recommendations. Another limitation is slightly more subtle in nature (dilution of the personal color in a sea of collective thinking): In a progressively personal world, where individual tastes are increasingly being better catered for, there is no such thing as the "perfect segment". Our aim for the recommender system is that it should approach as closely as possible the minimum segment size of 1. Segmentation is therefore a compromise between our ability to characterize a specific set of behaviors or attributes in order to define a user and the amount of available information and the real relevance and significance of those attributes connected to our context. So-called "Macrosegments" that can work correctly in a macro context (Women, Man 25-45..) are usually useless in terms of returning finely tuned recommendations. Each individual has a "color" of their own. Let us consider an example from a music recommender system, from the many currently available on the market (Pandora, Last.fm, Strands…) A hard rock music fan may also listen to a Synth Pop artist, and traditional recommenders will therefore associate that individual with a taste for BOTH kinds of music, so there will be a "poisoning" effect on future recommendations due to the apparent "anomaly", because the system does not handle "individual colors" but performs macrocluster mapping. The authors in [16] propose to solve this issue of different user 'faces' using a goal oriented recommendation, which keeps a common model and also a specific partial model for each of the user "goal/objectives". There is a risk that users may end up "belonging" to a specific cluster instead of what should really happen: A distinctive, unique personality should be matched to the shape of well known, well characterized "macroclusters", and the best fit selected. The current approach however could be compared to the process of making a random shape using paper and scissors and then trying to compare it with well known polygonal shapes: Circle, Pentagon, Octagon.., and then deciding which one fits best.

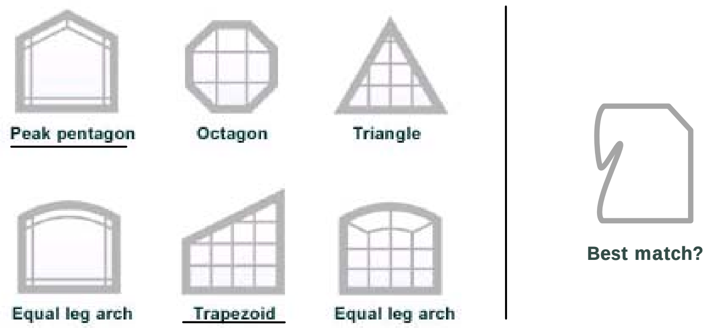


Fig. 1. The proposed shape shares some characteristics with the underlined pre-established shapes, but we cannot make a direct association to any of them beyond some shared characteristics

What we found interesting is that while we aim to achieve perfection in terms of pattern recognition and other such mathematical delicacies, we ignore as "non manageable" the capability to effectively and precisely draw a unique, non clustered, image of our user. This is where user modeling comes to the rescue: the main idea behind user modeling is to produce a "model" that tries to identify the key attributes belonging to a specific domain (in the case of Pandora, musical tastes) which can truly identify the user.

The problem with user modeling is a simple one: The model is produced (as accurately as possible), but it does not provide a suitable technique to ensure that several objectives are achieved. These objectives, fundamental in the overall process to guarantee a perfect personal recommendation experience, are the following:

a) To take the user from a "dummy" experience (i.e., one where they have had no involvement in the recommendation process) to being fully in control (fully tuning all the parameters included in the recommendation process) in a smooth and logical transition,
b) To provide an effective way of interacting with the user in order to engage him/her to produce more and more explicit feedback and profile detail.
c) To provide an effective framework for creating a constant "quid-pro-quo" scenario between provided data and improved responses from the recommender

User modeling provides a framework, but does not resolve the problem entirely. The user is not naturally enticed to cooperate, because there is no real incentive. In most approaches to recommendation engines there is one notable flaw: The systematic relying on Machine Learning and non-explicit feedback from the user to create the user model, where the possibility of truly engaging the user in the construction of their own profile is practically nonexistent. Why does this happen?

Mainly for one reason; most current approaches to recommender systems come from the "hard sciences" i.e., those related with knowledge based on rigid disciplines with fixed definitions such as mathematics and engineering Most of the sciences addressing the question of personalization however deal with "lighter" disciplines like etiology, psychology, marketing and so on , i.e., disciplines with a somewhat laxer approach to definitions and even contradictory solutions for identical problems. Therefore it seems that there is no possible way to rationalize and approach these disciplines systematically so it is best to rely on tried and tested scientific approaches. Unfortunately, although some work is emerging in this area [5], there is as yet very little literature existing on this matter[1]

2 Personalization: A Framework

Given the fact that "personalization" is a fairly vague word, which encompasses a lot of different definitions and approaches, with varying degrees of depth and no common consensus on the definition, we provide a series of basic components for our framework, dealing with the Personalization aspects of our work. Our proposal is a Balanced Recommender System defined as follows:
"A balanced recommender system is a approach which combines a recommender algorithm based on implicit, collective and behavioural data with a user's, explicit, user-centric and specific user model. The system uses additional tools and techniques provided to manipulate, enrich and fine tune the final recommendation."
The specific user model is not a generic one but depends entirely on the type of recommender involved (tv recommender, book recommender etc). Also the overall degree of involvement of the user in the creation of his/her profile has a significant impact on the quality of the final recommendation.
In this work we illustrate how we concluded that there was a need for this new type of recommender and describe the logic used to build our system.

2.1 Current use of the term "Personalization" in Recommender Systems

Supposedly, recommender systems - even the least sophisticated ones- deliver personalization. They deliver "personalized" recommendations, make "personalized" offers and deliver "personalized" messages. In our opinion however, this is not entirely the case. A detailed definition of personalization has been included in a previous reference[1] but for the purposes of this paper we will try to provide a less complex explanation:

"Personalization is a process which basically tries to adapt as closely as possible a product/message to a customer/speaker. The more accurate the analysis, the more accurate will be the recommendations. If we manage to grab the interest of our user

[1] One of the authors of this paper has published a book and several papers on a systematic approach for handling this problem, from which we have taken some definitions and some basic building blocks. Unfortunately, to date it is currently only available in Spanish: "Personalización" – Pearson Financial Times 2004 ISBN 9788420543543

and to obtain/understand their preferences, we will be more successful in our selling or communication proposition."

Therefore, we need to ascertain user preferences on the aspects relevant to our proposal (i.e.,, if we are trying to sell a Chinese cookery book, it is irrelevant to know the customer's hair color, but it is important to know that s/he is fond of cooking). Equally it is vitally important to know which communication channel the user is more receptive to. Adaptation involves a continuous process. Let us imagine for example that our objective is to paint a whole wall black. There is no such thing as "instantaneous wall painting" but rather it must be achieved one brush stroke at a time. In our case, the trivial data (i.e. name, address etc.,) are the equivalent of the brush strokes. When we use the word "personalization", the problem is that the verb "personalize" is like a kind of light switch, either it is on or off. Either you personalize or you do not. What really happens however is that there is a continuous process involved: It may be not be possible to personalize, it may be possible to personalize a little, it may be possible to have more or less accurate personalization, or have a completely tailor-made personalization. Clearly, "real" personalization is the latter of these possibilities, those really relevant to the user. Besides increasing the potential success of every subsequent interaction with our customer, there is another positive collateral effect arising from the use of personalization: After several relevant communications have been made, the customer/user pays more and more attention to our messages, because he has perceived them as relevant, unlike most of the communications they receive, where he/she perceives him/herself as an anonymous receiver. This precision is quite important, as we feel that there are too many unfounded claims of "delivers personalized results". In reality, the results may differ according to the true use of personalization in each scenario.

The recommender presented here is adapted to different systems and has been adapted with new features such as the one presented for the first time in this article (see sections 2.2, 2.3 and 2.4). Basically our proposal is a hybrid recommender that combines different features and separates long term and short term assumptions of the user model as presented in [17] [18]:

- Collaborative recommender (slope one)
- Content based recommender (WNBM, fingerprinting, PPM)
- Social recommender (Tags)

By having multiple sources for recommendations the cold start problem that appears in purely single source systems and especially in collaborative recommenders is avoided. This problem arises when a new item arrives, and no one has evaluated it, making it difficult to know how to recommend it. In our case the content based recommendation can be used initially in conjunction with the top relevance algorithm (see 2.2) and the Product Profile Matching approach (PPM) (see 2.3).

Summarizing, we have different recommenders: content based recommendations, a short term and a long term model, an item-item collaborative recommendation, one based on tags and the PPM. Each of these recommender approaches produces a list of programs and in order to calculate the relevance of each one for the user, we compute a weighted sum, where $\alpha+\beta+\varphi+\delta+\omega=1$. This determines the importance that we give to any of the four recommenders mentioned above (short and long are based on the same content based recomender). See Eq. (1). These parameters have an initial value that is updated for each recommender according to the amount of data available for it (i.e. if the number of user tags grows, this recommender will be given more importance). Besides automatic adjustment, the user can also express his/her preferences using the self fulfilling technique explained in 2.4.

$$R(user, item) = (\alpha R_{u,i}^{short}\ \beta R_{u,i}^{long})\varphi R_{u,i}^{collab.}\ \delta R_{u,i}^{tags}\ \omega R_{u,i}^{PPM} \quad (1)$$

2.2 Top Relevance Algorithm

For the cold start scenario we propose different solutions. One of these is to propose items based on their relevance for the users. As users evaluate the items (i.e., the book, tv program, artist etc) the relevance must take into account the number of evaluations of this item(FO_i), the quality of its evaluations($Av(O_i)$), and the total number of evaluations input into the system($|evO|$). It is important to make a good comb ination of these factora because if not we may find situations like the following in some systems:

$$ranking_i = Av(O_i) \quad (2)$$

At first glance, this approach may *seem* logical, as it means that an item will obtain its relevance according to the average of its evaluations. Let us suppose for the following examples that our items can be evaluated from (1 meaning very bad to 5 meaning very good). With eq. (2) there may be some strange results: if a document has been evaluated 100 times with an average of 4.2 it will be less relevant than one that has been evaluated only once with 5. This solution benefits newcomers and makes the top list very changeable and unstable.

$$ranking_i = Av(O_i) \times FO_i \quad (3)$$

On the other hand we could take equation (3). This would give the older items a better position in the ranking because they have been evaluated many times, even if the evaluations were not particularly good. So, how can we obtain the right solution? We wish to give reflect an appropriate value for well evaluated newcomers but also respect those items evaluated many times. If we analyze the information retrieval experience, a similar problem arises when trying to rank documents according to a query. The algorithm TF-IDF [19] with all its variants [20] tries to solve a similar problem associated to terms in documents. IDF gives more importance to a term if it appears a few times in all documents (similar to our newcomers that have been evaluated several times) whereas TF increases the importance of a term if it appears many times on one document (similar to our many times evaluated items). Therefore, inspired by IDF, the first serious approach to our algorithm was:

$$ranking_i = \frac{\log_2 \frac{|evO|}{FO_i}}{|evO| + 1} \times FO_i \times \text{Av(0i)} \quad (4)$$

This equation, (4) works quite well but the logarithmic function gives much more priority to the newcomers, and if an item has been evaluated many times independently of its evaluations it becomes less relevant because the logarithm approaches zero (these are extreme cases), also the difference between different evaluations is not taken into account. Finally, inspired by the modification of IDF by

Joachim [21], where he states “The second difference is that the square root is used to dampen the effect of the document frequency instead of the logarithm”, we changed the logarithm for a square root and squared the average evaluation in order to clarify the differences. The final equation is the following:

$$ranking_i = \frac{\sqrt{\frac{|evO|}{FO_i}}}{|evO| + 1} \times FO_i \times \mathrm{Av}(\mathrm{O_i})^2 \qquad (5)$$

To clarify with a simplified example let us suppose there are 7 items that users can evaluate in the range 1-5. We have the average of their evaluations $Av(O_i)$, the number of times the items have been evaluated FO_i. and the total number of evaluations done in the system $|evO|=Sum(FO_i)$. In Table 1 we can find on the left how the algorithm (4) sorts the items and on the right how algorithm (5) does so. We can observe that the results on the left may not be entirely accurate because for example an item evaluated 100 times as 2 is ranked better than another that has been evaluated 10 times as 4. The square root of the equation (5) solves this problem and its ranking looks much more realistic. The equation (5) can be used with different goals: 1) To create top lists, i.e. for the *top list of favorites* (items are sorted because users have selected them as favorites) or the *popular items* . (sorts the items according to popular user evaluations). It could be used for example to obtain the most recent and popular selections 2) To tackle the Cold start problem. New users could obtain recommendations of the most popular items in the system as other personalized recommendations cannot be calculated yet or 3) To have initial estimations if the Personal and Explicit Profile (explained in the following section) has not yet been created.

Table 1. Comparison of the Ranking using a) the Logarithmic eq. (left) and b) the Square eq.(right)

$Av(O_i)$	FO_i	Log(eq.(4))	$Av(O_i)$	FO_i	Square(eq.(4))
5	40	2,21336768	5	40	10,5408205
4	40	1,77069415	5	20	7,45348565
5	20	1,55311241	4	40	6,74612512
2	100	1,03307474	4	10	3,37306256
4	10	0,79981639	5	3	2,88672258
5	3	0,41624581	2	100	2,66664009
2	10	0,3999082	2	10	0,84326564

2.3 Product Profile Matching

We understand PPM as a continuous process that involves the following elements:

A) A detailed User Explicit Profile (usually considered the user model), regarding the specific domain that in each system is being covered.

B) A product item, (which we can associate to something called the item model). This would involve the characteristics of the item relevant to the decision making process

C) A complete detailed model of the application of both group A and B, which could predict individual affinity between the specific user profile and the item model, not on a cluster basis but on an individual basis.

It is a continuous process because all three models are subject to continuous improvement, and a possible initial approach could yield some information helpful for improving every model. The key here is relevance. The criteria of inclusion /exclusion of attributes in these models is not to do with how easily they can be obtained but their relevance in the Product-Profile relationship. The design of both the attributes and the relationship must be done independently of the feasibility or any other factors that could hamper the creation of the best possible affinity mechanism model. Compromises can be made later but the model should take into consideration every single cause-effect that could influence the affinity model.

PPM involves a dedicated effort to create a taxonomy that must be addressed in a professional way, by people with knowledge of both business fields (User model – Item Model). Let us imagine a PPM model created for an online bookstore: There should be a clear customer expert behind the creation of the customer model, a librarian perhaps, and some kind of product manager behind the creation of the product mode. The combination of these, perhaps someone from a commercial department, should be behind the affinity model.

Product characterization does not need to be extensive if the relevance prerequisite previously mentioned is fulfilled – the authors have produced a paper on a process of PPM from the Product side [22] in which there is considerable compacting of an exhaustive product characterization (TV content, made by Anytime TV) into a more compact, easy to manage form and they present a taxonomy which would be a perfect product model for a PPM scenario, along with a complete TV user model and a complete Affinity model (More on this in [17] [18]).

How does PPM relate to balanced recommenders? A Proper PPM schema should be included in balanced recommenders for the following reasons:

- It must deal with the "individual" aspects of the recommendation, like the rest of the aforementioned techniques discussed previously.
- It provides a strong initial starting point thereby avoiding the Cold Start scenario (working in conjunction with the aforementioned solutions), translating the responsibility of preventing the cold start to the user providing detailed info on his model (as the product model has been previously covered, as well as the affinity application model).
- It offers a strong model to refine the overall recommender results.

2.4 Self Fulfilling Capabilities

Another key component of a Balanced Recommender system is the existence of Self fulfilling capabilities and a proper Self Fulfilling strategy must be in place. Let us try to develop this. Most recommenders have adopted an approach that we strongly discourage – that of keeping the user away from the underlying algorithm used. This is like telling users "Trust us, we are really smart" – not the best approach for a supposedly "personalized" approach. Users do not have a real sense of being in control and we think that it is important to allow people to decide to what degree outside or collective intelligence should play a part in the suggestion / decision provided by the recommender. As our current proposal involves several different features and it is highly likely that different users will have different degrees of collaboration, the following steps should be taken:

- A step-by-step system should be developed to educate the customer on how to move from a fully automated to a fully (user) controlled contribution for every factor
- The degree of precision in the recommendation should be directly linked to the following factors:
    - How well the user understands the underlying model and how this affects their input and fine tuning,
    - The degree of fulfillment of the data user model proposed, through a clear "tit for tat" proposition – you provide me with better data and a better recommendation will be the result.
    - The degree of precision must not be related to external factors such as intrinsic data quality or the degree of training of the recommender network
- All the contributions involved in the recommendation algorithm should be shown for the customer to fine tune and adjust their preferences once they have received appropriate training on the matter: i.e., they should be informed of how much weight was given to content based evaluations, how much to collaborative, how much to PPM etc…

With some kind of visual metaphor and some easy feedback procedures we are sure that people will have a much better experience with recommenders than has been the case up to now (see Fig. 2).

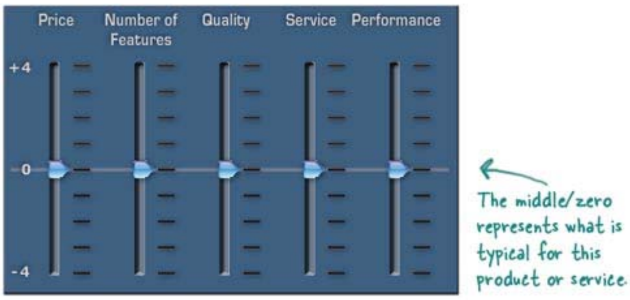


Fig. 2. The user can adjust the recommender parameters

3 Mirotele: A Balanced Recommender at work

Mirotele[2] is a joint venture between the authors in which we have been involved [17] [22] [18] for some time and we are using it to test all our theories at the moment. We have created a whole User model (representing what we consider to be the

[2] http://www.mirotele.com

relevant attributes involved in TV preferences), a whole self fulfilling environment (where advanced users can start to tinker with the data involved in the algorithm once they begin to appreciate how it works). We have developed a whole PPM schema - combining the aforementioned user model with a fingerprinting method for TV Programs [22]. In addition, we incorporate all the improvements made to the existing algorithms mentioned previously. There is as well a whole social networking schema, a cross between a wiki and a folksonomy approach, where the social network is used not go generate, but to filter and gather a huge amount of content related information that has been produced via an automatic information gathering and classification system (using web services from Google, YouTube, and other TV related services). Unfortunately we have not yet collected data on a real implementation for the current version of this paper (April 2009).

4 Conclusions and Future Work

Our current conclusions are basically the following:

- Although our Balanced Recommenders incorporate a hybrid approach, they are not the same as hybrid recommenders. This is not merely a question of semantics. We are mixing personalization techniques and classic recommender techniques. In our recommenders, the knowledge domains are quite different and the result is much more intuitive than in the hybrid approach. Personalization is a science in itself and trivial approaches must be avoided. We have found several personalization techniques like PPM that are appropriate for expanding the current recommender schema. We do not discard the possibility of enriching our schema with other techniques in the future, and have found that “balancing” a purely scientific approach with personalization techniques has produced an extremely good/promising result.
- In the future there will be a systematic shift from current recommender schemas to balanced approaches like the one we present here.
- We foresee a new golden age in the use of recommenders systems as they gradually become important information-organizers, substituting those currently in existence (mostly Search engines).

We are currently working on the implementations of our schemas and algorithms and plan to continue researching the area of balanced recommenders, in particular dealing with the less documented and structured aspects of personalization techniques. At the same time we will continue to improve our tools and attempt to determine as much as possible the correct combination of every factor considered here in order to achieve “the perfect recommendation”. Perhaps it is as elusive as the perfect cocktail, but our ultimate goal is to improve current.

5 References

1. G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions," Knowledge and Data Engineering, IEEE Transactions on, vol. 17, no. 6, pp. 734-749, 2005.

2. D. Billsus and M. Pazzani, "A Hybrid User Model for News Story Classification," Banff, Canada: 1999.
3. N. Good, Schafer J., J. Konstan, A. Borchers, B. Sarwar, J. Herlocker, and J. Riedl, "Combining collaborative filtering with personal agents for better recommendations," 1999.
4. P. Cotter and B. Smyth, "WAPing the Web: Content Personalisation for WAP-Enabled Devices," Springer-Verlag, 2000.
5. H. Cramer, V. Evers, S. Ramlal, M. van Someren, L. Rutledge, N. Stash, L. Aroyo, and B. Wielinga, "The effects of transparency on trust in and acceptance of a content-based art recommender," User Modeling and User-Adapted Interaction, vol. 18, no. 5, pp. 455-496, Nov.2008.
6. M. Balabanovic and Y. Shoham, "Combining Content-Based and Collaborative Recommendation," Communications of the ACM, vol. 40, no. 3 1997.
7. D. Bueno and A. A. David, "METIORE: A personalized information retrieval system," User Modeling 2001, Proceedings, vol. 2109, pp. 168-177, 2001.
8. M. Pogacnik, J. Tasic, M. Meza, and A. Kosir, "Personal Content Recommender Based on a Hierarchical User Model for the Selection of TV Programmes," User Modeling and User-Adapted Interaction, vol. 15, no. 5, pp. 425-457, Nov.2005.
9. G. Linden, B. Smith, and J. York, "Amazon.com recommendations: item-to-item collaborative filtering," IEEE Internet Computing, vol. 7, no. 1, pp. 76-80, 2003.
10. R. Burke, "Hybrid Recommender Systems: Survey and Experiments," User Modeling and User-Adapted Interaction, vol. 12, no. 4, pp. 331-370, Nov.2002.
11. L. Candillier, K. Jack, F. Fessant, and F. Meyer, "State-of-the-Art Recommender Systems," in Collaborative and Social Information Retrieval and Access: Techniques for Improved User Modeling. M. U. o. T. Chevalier, Ed. 2009, pp. 1-22.
12. A. Trotman, "Learning to Rank," Information Retrieval, vol. 8, no. 3, pp. 359-381, Jan.2005.
13. S. Brin and L. Page, "The anatomy of a large-scale hypertextual Web search engine," Computer Networks and ISDN Systems, vol. 30, no. 1-7, pp. 107-117, Apr.1998.
14. Klaus, "Algoritmo de Popularidad Actualizado," in http://blogs.gamefilia.com/markus/10-02-2008/854/algoritmo-de-popularidad-actualizado.
15. G. Asela and M. Christopher, "Tied boltzmann machines for cold start recommendations," in Proceedings of the 2008 ACM conference on Recommender systems Lausanne, Switzerland: ACM, 2008, pp. 19-26.
16. D. Bueno, "Recomendación Personalizada de documentos en sistemas de recuperación de la información basada en objetivos." Ph.D. Universidad de Málaga, 2003.
17. D. Bueno, R. Conejo, and J. G. Recuenco, "An architecture for a TV Recommender System," International Workshop on Personalization in iTV.Euro ITV 2007 Amsterdam: 2007, pp. 117-122.
18. D. Bueno, R. Conejo, D. Martín, J. León, and J. Recuenco, "What Can I Watch on TV Tonight?," 2008, pp. 271-274.
19. J. Rocchio, "Relevance Feedback in Information Retrieval," The SMART Retrieval System: Experiments in Automatic Document Processing, pp. 313-323, 1971.
20. S. Robertson, "Understanding inverse document frequency: on theoretical arguments for IDF," Journal of Documentation, vol. 60, no. 5, pp. 503-520, 2004.
21. T. Joachims, "A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization," 1997, pp. 143-151.
22. J. Recuenco, N. Rojo, and D. Bueno, "A New Approach for a Lightweight Multidimensional TV Content Taxonomy: TV Content Fingerprinting," in Changing Television Environments 2008, pp. 107-111.

Visualizing Reciprocal and Non-Reciprocal Relationships in an Online Community

Kadhambari Sankaranarayanan, Julita Vassileva

Computer Science Department, University of Saskatchewan
Saskatoon, SK, S7N 5C9 Canada
{kas411,jiv}@mail.usask.ca

Abstract. Online communities thrive on their members' participation and contributions. There are numerous ways to visually represent information, current status, power, and acceptance of members in an online community. In this paper we present a design of a visualization representing reciprocal and non-reciprocal relationships among users, which emphasizes and hopefully triggers common bond in the community. Our future goal is to see whether the visualization triggers higher participation in an online community called "WISEtales", which currently is mostly based on common identity. If our hypothesis is confirmed, it will present one of the few examples of successful community whose members associate both by common identity and common bond.

Keywords: Information visualization, social visualization, visual design.

1 Introduction

Designing a visualization tool for an online community is a great challenge in the field of visualization research. During its existence an online community produces huge amount of content and it becomes difficult for the user to navigate and find the information that they are looking for. It also becomes complex to understand the evolution and the type of relationships that exist among members. "Social visualizations are one way to "describe" our online environments and make interaction patterns and connections salient" [1]. Any visualization should evoke meaning beyond direct mapping of data otherwise it is said to be misleading. Social visualizations have some evocative quality [2].

WISETales is an online community for Women in Science and Engineering. This community has been developed by a graduate student, Zina Sahib, as one of the projects of the NSERC/Cameco Chair for Women in Science and Engineering for the Prairies, Dr. Julita Vassileva. This community is specially designed to allow women who are underrepresented in these areas to share their personal stories. This is a virtual channel to share emotion, experience and provide support to other women. It helps women to overcome the generation gap and isolation. Generally women in these fields are very busy and achieving active participation is a great challenge. So to motivate their participation is vital for the existence of the community. In order to overcome this problem, we propose to use a visualization of user relationships that can motivate users to contribute and reach a critical mass of active users.

2 Literature Survey

Our research covers the area social comparison and motivational theories in psychology, organizational theories: common identity and common bond theory, and social visualization.

Theories of Motivation in Psychology

According to the cognitive evaluation theory there are two motivation systems - intrinsic and extrinsic - that corresponds to two kinds of motivators. Intrinsic motivators are: achievement, responsibility and competence, motivators that come from the actual performance of the task or job, the intrinsic interest of the work. Intrinsically motivated individuals perform for their own achievement and satisfaction. Extrinsic motivators are: pay, promotion, status, power, better working conditions, feedback that comes from a person's environment, controlled by others [3].

One of the theories from social psychology that is used to explain human motivation is the social comparison theory [4]. Social comparison consists of comparing oneself with others in order to evaluate or to enhance some aspects of the self. Cognitive and emotional responses to comparison have been extensively studied, but less is known about the effects of comparison on behavior. There is very little guidance about how people compare themselves in an online community. Sun and Vassileva [5] examined the effect of making individual reputation visible in an online system for sharing research papers and found out that displaying reputation increased contributions but some users contributed low quality content simply to achieve higher reputation. A study on the MovieLens movie rating system was conducted [6] by sending email newsletters to users indicating whether their contributions to the community were above or below or about average when compared to others which involved men and women. Women reported being motivated to contribute more ratings when they were told they had rated approximately the same number of movies as others and men were motivated to contribute more when they were told they had rated fewer than others. Members who received a newsletter that encouraged social comparison rated more movies than other members who received a newsletter which didn't encourage social comparison. Upward comparisons were most motivational in this system. However, introducing social comparison into a community might be risky. It could work and increase member participation or it might not work and reduce member's contributions. Competitive and gaming members like to be compared with other members, but others may find it discouraging and de motivating. People who are by nature more competitive (stereotypically, men are believed to be more competitive than women) are more likely to be motivated by the upward social comparison condition. It is arguable if women are less competitive, and especially if women in the science and engineering field are less competitive. They may respond very well to social comparison. However, in this research we would like to experiment with creating a visualization that emphasizes relationships, based on the common bond theory. It is generally considered a bad idea to mix motivations (e.g. extrinsic and intrinsic motivation) in the same system. Similarly, we fear that mixing social comparison with common bond may negate each other and it may be impossible to observe any change in user participation, or it will be hard to attribute the change, if there is any.

Common Identity and Common Bond:
Community design affects how people can interact, the information they receive about one another and the community, and how they can participate in community activities. There are two theories of group attachments that have been linked to design decisions on online communities [7]. The *common identity theory* makes predictions about the causes and consequences of people's attachment to the group as a whole and the *common bond theory* makes predictions about the causes and consequences of people's attachments to individual's group members.
The causes of *common identity* are social categorization, interdependence and intergroup comparison.
Social categorization: it happens when one creates a group identity by defining a collection of people as members of the same social category [8][9]. *Interdependence*: Groups whose members are cooperatively interdependent tend to become committed to group [7]. *Intergroup comparison:* People who define and categorize themselves as members of a group compare themselves with other groups [10] and raising the salience of out-groups intensifies people's commitment to their in-groups. The causes of *common bond* are social interaction, personal information, and personal attraction through similarity.
Social interaction: Social interaction provides opportunities for people to get acquainted, to become familiar with one another, and to build trust. As the frequency of interaction increases, their liking for one another also increases [11]. *Personal Information:* Opportunities for self-disclosure when members exchange personal revealing information about the self becomes a cause or consequence of interpersonal bonds [12]. *Personal attraction through similarity:* People like others who are similar to them in preferences, attitudes and values, and they are likely to work or interact with similar others. Similarity can create common identity as well as interpersonal bonds [7].

Comparison of Common identity and Common bond:
Some identity-based communities shift eventually toward supporting and promoting interpersonal connections among members. For example, Flickr.com was established as an online application for photo management and sharing but it later evolved into a community where people not only share, tag, and comment on photos, but also join groups and interact in its public and private forums[7].

Bond based communities help newcomers to connect with existing members, to join group interactions, and to form lasting relationships with a subset of community members. Bond-based communities care more about people-finding than information finding, making it easy to find and meet specific members through directory or personal profile search page [7]. These communities encourage personal relationships, and their introductory material often encourages participants to post on a wider range of topics [7]. As compared to common identity, in common bond based communities newcomers feel isolated and become confused to see off topic discussions among members. But in our research since all discussions would be based on members stories, newcomers would be able to understand every part of the discussion once the corresponding story is read and it would not be off putting for them.

Reciprocation

In Common Bond based community people develop relationships with other members and that is what ties them to the community which cannot be expected from Common Identity based community. People often help others with the expectation that their help would be compensated or reciprocated, either by those they have helped or by the group as a whole [13] [14]. Thus reciprocation can happen at a dyadic or at community level. In case of common bond there is direct reciprocity and in case of common identity there is general reciprocity. Social psychologists have found that the urge to reciprocate is deeply ingrained [15]. Sellers and buyers on eBay usually reciprocate in their ratings of each other [16] Voting on web sites is sometimes done in the context of reciprocity [17]: if you rate my story highly I will rate yours highly. Networks of reciprocity are highly motivating, and encourage participants to maintain an awareness of the community that surrounds them [18].A community designed on the basis of common identity is said to be more stable when compared to community designed on the basis of common bond [7]. This is because, in common bond based community, if a member leaves the group, the friends associated with that member would also likely leave the group or become passive. This does not occur in community designed on the basis of common identity. WISEtales is designed on the basis of common identity theory, so we can expect that it would be more stable. Representing relationships in a common identity based community encourages common bond. As very little research has been done on the coexistence of identity-based and bond-based attachment, this encourages us explore combining cues that stimulate both kinds of attachment. According to Milgrams [19] and Zajonc[20], visually representing people in an online group formed personal attachment to them even without communicating with each other. Visualization of actual communication flow among community members can create bond between friends of friends by helping people fill in gaps [7]. Making contributions visible in a community as a whole leads to some extent of recognition of the member's contributions. The nature of online interaction means that helpful acts are more likely seen by the group as a whole. The following features encourage reciprocity: ongoing interaction, identity persistence, and knowledge of previous interactions, since they promote the creation and importance of reputation within a community. So visualizing reciprocal and non reciprocal relationships might help members to recognize their current position in the community.

Social Visualization

Visually representing information enables users to see data in context, observe patterns and make comparisons [21]. Visualization techniques are important aids in helping users and researchers understand social and conversation patterns in online interactions [22]. A data portrait of an online community can give overall information about each other and the overall social environment [23]. "Social visualization is defined as the visualization of social data for social purposes" [24]. Social visualization is a sub category of information visualization. It focuses on people, groups, conversational patterns, interactions with each other and relationships with each other and with their community. Social networks are said to be a form of social visualization because they have two types of organization patterns namely social groups and social positions [25]. There are various techniques to represent a group of people in an online community. Most approaches use nodes to represent individuals and arcs between the nodes to represent connections between them. Real social networks have dense interconnections between people.

Vizster is a visualization system for playful end-user exploration, navigation of large-scale online social networks to increase awareness of the community. Heer et al. [21] found out by observing through Vizster visualization that groups of users, spurred by stotytelling of shared memory spent more time in exploring stories and asked deeper analysis questions than other members. Further Vizster's visual community analysis provided help to users who could construct and explore higher-level structures of their online communities. Visualizations provide not just an analysis tool for social science researchers. Heer et al. [21], through the "sense.us" visualization for group exploration of demographic data found out that combining conversation and visual data analysis helps people to explore data broadly and deeply. When visualizing conversations, it should evoke an appropriate intuitive response to represent the feel of the conversation as well as depict its dynamics [26]. Coterie, a visualization tool for Internet Relay Chat (IRC) shows the activity of the participants and also the structure of conversation. It highlights active participants and conveys the vitality of discussion [26]. PeopleGarden is a visualization tool for representing member's participation on a message board. It uses flower and garden metaphor. From this anyone can easily perceive an individual's active role or long-time lurker [26]. The Loom Project is an evocative semantic visualization for Usenet newsgroups. It is used to depict the leaders and provocateurs. There are people who post frequently and are often replied to in a positive way. This visualization distinguishes them from other frequent posters such as trolls (deliberate troublemakers), automatic newsfeeds, and the excessively verbose [26]. IBlogVis [27] is a visualization tool for browsing blog archives. It provides an overview of posted blog articles over time with their length and number of comments received to help users to find the interesting articles in the blog at a glance and to ease exploration and navigation. Social network visualization for blogspace revealed that topic-oriented blogs had more interconnections and reciprocation than most popular blogs [28]. Webster and Vassileva [29] explored in the context of a discussion forum, if a visualization of the reciprocity of a user's relationships with other users would motivate the user to engage in more reciprocal relationships and showed that it indeed does so for active members, though it doesn't increase the level of participation in general. Chin and Chingel's [30] visualization for blogspace show links for suggesting a social relationship among the bloggers. Social visualization is expected to activate social norms of behavior, encourage social comparison and reciprocity. According to Vassileva and Sun [5] motivational visualization effectively increased awareness of community and encouraged social comparison and as a result contribution to the community increased. We propose to incorporate a motivational visualization to increase participation by stimulating social bond among members and evoking reciprocity among between pairs of users, as well as a gentle social comparison in terms of number of reciprocated relationships.

3 Proposed approach

To achieve the goal of increasing active participation, we propose designing a system which incorporates visualization techniques to motivate user participation by evolving their relationships with other members in the online community.

Motivation

Our hypothesis is that an appropriately designed visualization can stimulate motivational and organizational mechanisms that lead to more active contributions by users to their community. Our approach is to encourage intrinsic motivation (according to the cognitive evaluation theory from psychology) and the common bond theory (from organizational studies). The objective of our research would be to model the evolution of relationships based on data from user interactions, for example reading and writing stories or giving comments and to design a visualization of these relationships which will serve as a tool to motivate users to contribute more towards their group. Our visualization would display these relationships between users so that it would be easy for the user to understand his/her current position in the community.

We have chosen the WISEtales community as a test bed for our approach. In bond-based community people engage in direct reciprocity. So the visualization will reveal which are reciprocal and non reciprocal relationships. Reciprocity increases when members interact repeatedly. People help others with the expectation of having their help returned by that individual or the group as a whole [13] [14]. Returning favors is are acts of reciprocation. Yet it is not clear if being aware of the reciprocity of their relationship, and the direction of non-reciprocal relationships (who “owes” favors to whom) will motivate users to reciprocate more frequently and thus contribute more. This is what we would like to test. In this community, reciprocation happens when a member reads a story or post comments to a story submitted by someone else. Other actions, such as posting a story to one’s Facebook profile, forwarding it to a friend or checking the story, author’s profile may also be considered as acts of reciprocation.

Visualization Design

To make the visualization more likeable for women, a flower garden metaphor is used (see Figure 1). Each user is represented in circular node with his/her name written in it. The node is surrounded by arcs (visualized as leafs) corresponding to relationships with other users. Each arc (leaf) has the corresponding user names and different color to indicate reciprocal and non reciprocal relationships. The stronger and thicker the color then the reciprocation is said to have happened between the users. This helps the users to understand how many reciprocal and non reciprocal relationships they and the other users are involved in. The node of the viewer will be highlighted among the other circular nodes, so that he / she can compare his/her relationships with those of the other users. If a user has received lot of comments from a particular user and has not been aware of that before, the visualization will make him/her realize that he/she “owes” that user some attention, and that he/she needs to contribute something to the other user. Also the realization that other users are viewing the same visualization and will be aware of the lack of reciprocation from the user to others will add social pressure to behave according to community norms (a form of social comparison).

Thus a social visualization showing the users' relationships with other users could be motivational, if users become aware about the number and balance of their reciprocal and non reciprocal relationships with other users expressed through the visual effects. They would get an overall idea about the other members' contributions, would be interested to read stories contributed by active members, post comments and also spread the word about interesting stories. The visualization will be dynamic – it will change when new members sign in and when new comments are given and reciprocal actions performed. The visualization is intuitive but not interactive since previous research by [5] showed that interactive features were rarely used. It is not customizable by the users.

One can see in Figure 1 that there are three distinct colors used to represent reciprocal and non reciprocal relationships. The more petals a flower has, the more the active the member is. The dark green color leaf is used to represent reciprocation among users; the medium green color leaf is used to represent comments received from other users and the light green color leaf is used to represent comments given to other users. Viewers perceive colors differently but experimental evidence shows that relationships between colors are universal and are free from individual and cultural differences [31]. According to [31], "People can make consistent evaluation of the magnitude of any given experience of colors based on the type of interaction among colors. People respond to the relationship among colors". The colors chosen for this visualization are of analogous ordering. Such kind of ordering is more lively than monochrome and is stable in arrangement than non analogous ordering or complementary parings. Each member is represented as circular node in brown color.

The person who is engaged in most reciprocal relationships is placed in the center and other members are placed surrounding it. According to [32] "Varying shapes of nodes is used to denote different characteristics of members in the graph; the location of the node is used to denote the valuable marker for understanding the structure in the network. Centrality in a group is a useful indicator that the participant plays a key role in the group [33]. Each leaf has a rounded and a very sharp edge. The sharp edge is placed outside and is rotated to point to the direction of the corresponding individual's node whose name is mentioned on the respective leaf (along the arc connecting the nodes representing the users). The reason is to give an easy navigation and sense of direction for the user to find their relationship partners in the visualization.

Reciprocation between two members is currently calculated by the number of views and comments to each other's story. For example, in Figure 1 it can be seen that Karthik's node is placed in the center as it has a higher number of reciprocated relationships when compared to other nodes. The members with fewer reciprocated relationships are placed surrounding the central person. The other members with very few relationships are placed in the outer circle. All nodes in the graph are created using concentric circle algorithm. Placing the leaves in the corresponding direction of the node is not a trivial task. It is done by using some rotation measures and graphics algorithm to generate the graph.

This visualization does not include any connection lines between nodes. "The fewer the number of lines crossing, the better the sociogram" [32]. This is because lines between nodes increases complexity and decreases the beauty of the visualization. The visualization comes with a key to help users indentify which colors represent which type of relationship.

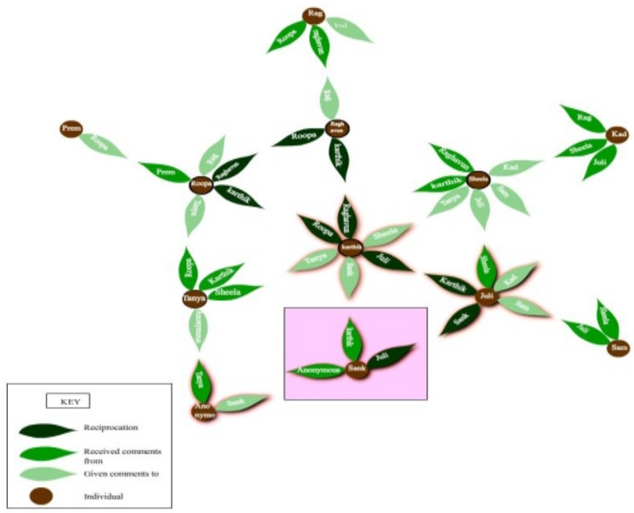


Figure 1: Visualization of reciprocal and non-reciprocal relationships for logged in members of WISETales.

Implementation

The technology used to design the visualization is Flash and Flare. Flare is mainly used for web content visualization and is highly scalable. WISETales is built on Drupal, PHP and MySql technologies. Flash can easily integrate with PHP and MySql. A link to the visualization will be implemented in WISETales website. As soon as the member of WISETales website logs in he/she would be able to click on the link to visualization to see it. In the visualization, the area of the corresponding member who is currently viewing the visualization would be highlighted in pink to show his/her current position in the group. Also when they click on their node all the nodes and leaves that are related to them representing reciprocal and non-reciprocal relationships would also be highlighted in the visualization. User of the visualization can also click on the particular flower to scale to get the information of a particular person clearly.

Prototype Evaluation

A medium fidelity prototype of the visualization using Flash was developed and tested to assess whether the visualization of reciprocal and non-reciprocal relationship conveys the correct information to the user, whether they were able to understand the visualization clearly. The evaluation tool used for the medium fidelity prototype was a questionnaire. The question type used were Scalar-Likert scale because it measures opinions, attitudes and beliefs. Each question asks the user to judge a specific statement on a numeric scale with extremes 4 –indicating agreement and 1 –

indicating disagreement with a statement. I also used open questions to get specific answers and to give room for user suggestions. The questionnaire was implemented using the SurveyMonkey.com tool.

The representative users for the evaluation of the medium fidelity prototype were 12 graduate students from our MADMUC lab at the University of Saskatchewan. The link to the prototype as it ran on a server and a link to the questionnaire were sent to each participant in an email. The most serious concerns users had were related to the scaling of the visualization, as can be seen in Figure 2. We need to work on the scalability, perhaps through creating fish-eye views or a magnifying glass effect.

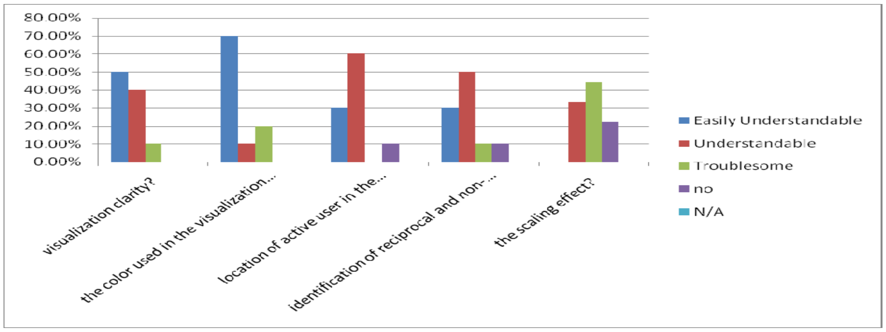


Figure 2: Results of the evaluation of the visualization prototype.

Future evaluation of the visualization

Our hypothesis is that visualizing reciprocal relationships would increase the users understanding of their community, will encourage common bond and will ultimately increase participation. We chose to evaluate the effect of proposed visualization in WISEtales by using three different versions (two control versions and experimental version) of the community with two different groups of users. Fifteen members would participate in each version. The experimental version would have the proposed visualization. The first control version will have no visualization and the second control version will have a different visualization (one developed by Zina Sahib) and based on common identity theory, showing only the type of contributions, not the users. All members would be given a period of one month to use the community with their respective version. In the next two months, the groups will rotate their versions, so that each group gets exposure with each version. The contributions from members in experimental version and members in control version and their reciprocal relationship with other members would be collected and analyzed. A questionnaire will also be used to collect qualitative data about the users understanding of the structure of the community, the importance of individuals in it; as well as their feelings of attachment to particular individual or the community as a whole.

4 Conclusions

We propose to use a motivational visualization aimed at encouraging common bond in a common identity based community and see the effects on user contributions. We want to test if particular visualization design, showing how users are engaged in reciprocal and non-reciprocal relationships with each other could stimulate reciprocation and motivate higher user participation. If our hypothesis turns to be true this may provide empirical evidence about the possibility of successful and stable co-existence of common identity based community and common bond based community within one group.

References

1. Karahalios, K.G 2006 'Social Visualization: Exploring Text, Audio, and Video Interaction'. CHI April 22-27 ACM.
2. Boyd, D., Lee, H-Y, Ramage,D. and Donath, J. Developing legible visualizations for online social spaces. In Proceedings of the Hawaii International Conference on System Sciences. (Jan. 7-10, 2002, Big Island, Hawaii).
3. Theories of Motivation. http://www.analytictech.com/mb021/motivation.htm
4. Festinger, L. A Theory of Social Comparison. Human Relations, 7, 1954.
5. Vassileva,J and Sun,L.(2007) Using Community Visualization to Stimulate Participation in Online Communities. E-Service Journal, 6 (1), 3-40. (special Issue on Groupware).
6. Harper, M.F., Frankowski, D., Drenner, S., Ren, Y., Kiesler, S., Terveen, L., Kraut, R., Riedl, J Talk Amongst Yourselves: Inviting Users To Participate In Online Conversations, 2007 IUI, Jan 28-31, ACM.
7. Ren ,Y., Kraut, R., and Kiesler, S. 2007. Applying Common Identity and Bond Theory to Design of Online Communities. ACM. Syst. 15, 5 (Nov. 1993), 795-825. DOI:10.1177/0170840607076007, http://oss.sagepub.com/cgi/content/abstract/28/3/377
8. Turner, J.C. 1985 'Social categorization and the self-concept: A social cognitive theory of group behavior' in Advances in group processes: Theory and research, Vol. 2. E. J. Lawler (ed), 77-122. Greenwich, CT:JAI Press.
9. Turner, J. C., M.A.Hogg, P.J. Oakes, S.D.Reicher, and M.S. Wetherell 1987 Rediscovering the social group: A self-categorization theory. Oxford; Blackwell.
10. Tu, M. A., and D. J. Terry 2000 'Social identity and selfcategorization processes in organizational context'. *Academy of Management Review* 25/1: 121–140.
11. Cartwright, D., and A. Zander 1953 'Group cohesiveness: Introduction' in *Group dynamics: Research and theory*. D. Cartwright and A. Zander (eds). Evanston, IL: Row Peterson**.**
12. Collins, N. L., and L. C. Miller 1994 'Self-disclosure and liking: A metaanalytic review'. *Psychological Bulletin* 116/3: 457–475.
13. Blau, P.M. 1964 Exchange and power in social life. New York: John Wiley.
14. Emerson, R.M. 1972 'Exchange theory: A psychological basis for social exchange' in Sociological theories in progress (Vol. 2,pp. 38-87). J. Berger, M. Zelditch and B. Anderson (eds). Boston: Houghton Mifflin.
15. Cialdini, R. B. *Influence: Science and practice* (4th ed.). Boston: Allyn & Bacon. 2001.
16. Resnick,P and Zeckhauser,R., Trust Among Strangers in Internet Transactions: Empirical Analysis of eBay's Reputation System (2002). The Economics of the Internet and Amsterdam, Elsevier Science. Pp. 127-157.

17. Dellarocas, C. N., Fan, M. N. and Wood, C. A., Self-Interest, Reciprocity, and Participation in Online Reputation Systems (February 2004). MIT Sloan Working Papers No. 4500-04.
18. Sadlon, E., Sakamoto, Y., Dever, H. J., Nickerson, J. V. (2008). The Karma of Digg: Reciprocity in Online Social Networks In Proceedings of the 18^{th} Annual Workshop on Information Technologies and Systems.
19. Milgram,S. 1997. 'The familiar stranger: An aspect of urban anonymity' in The individual in a social world: Essays and experiments. S. Milgram (ed.), 51-53 Reading, MA: Addison-Wesley.
20. Zajonc, R. B. !986 ' Attitudinal effects of mere exposure'. Journal of Personality and Social Psychology 9/2, PT. 2: 1-2
21. Heer,J., Viegas, F.B and Wattenberg, M. 2009 'Voyagers and Voyeurs: Supporting Asynchronous Collaborative Visualization'. Proceeding of the SIGCHI Conference on Human Factors in Computing Systems, April 2007. VOL. 52, Communications of the ACM.
22. Viegas,F.B., danah boyd, Nguyen, D.H., Potter, J., and Donath,J. Digital artifacts for remembering and storytelling:PostHistory and social network fragments. In Proceedings of Hawaii International Conference on System Sciences (HICCSS) (2004), 105-111.
23. Xiong, R and Donath,J (1999). "PeopleGarden: Creating Data Portraits for Users" MIT Media Laboratory, Cambridge ACM.
24. Karahalios K.G and Viegas F.B. "Social Visualization: Exploring Text, Audio, and Video Interaction" CHI 2006, ACM.
25. Freeman, L.C., Visualizing Social Networks. Journal of Social Structure, 2000.
26. Donath,J.2002 'A Semantic Approach to Visualizing Online Conversations' (Vol. 45). Communications of the ACM.
27. Indratmo, Vassileva J and Gutwin C, 2008 'Exploring Blog Archives with Interactive Visualization', ACM (May 28-30).
28. Herring, S.C., Kouper, I., Paolillo, J.C., Scheidt, L.A., Tyworth, M., Welsch, P., Wright, E., and Yu, N. Conversations in the blogosphere: an analysis "from the bottom up." In Proc. HICSS-38, 2005
29. Webster,A and Vassileva,J ., Visible Relations in Online Communities, in Adaptive Hypermedia and Adaptive Web-Based Systems, Dublin, Ireland, Springer LNCS 4018, 223-233.
30. Chin, A. and Chignell, M. A social hypertext model for finding community in blogs. In Proc. HYPERTEXT, 11-22, 2006.
31. Jacobson, N and Bender, W. "Color as a determined communication", IBM Systems Journal, VOL 35, NOS 3&4, 1996.
32. Moreno, J.L. Who Shall Survive? Beacon, NY: Beacon House, Inc. 1953.
33. S.Wasserman and K.Faust. *Social Network Analysis*. Cambridge University Press, 1994.

A User-Centric Authentication and Privacy Control Mechanism for User Model Interoperability in Social Networking Sites

Yuan Wang and Julita Vassileva,

Department of Computer Science
University Of Saskatchewan Saskatoon SK,
Canada S7N5C9 1-306-966-4744

yuw193@mail.usask.ca jiv@cs.usask.ca

Abstract. In this paper, we present a new authentication and privacy control mechanism for personalized mashups of social networking sites. Current authentication and privacy control mechanisms lack flexibility and transparency. This mechanism can make the user model interoperation process for mashups more transparent to users. Users can have a clear understanding and control about which part of their data is being accessed by the mashup application. This mechanism is an important part of user model interoperability framework.

Keywords: Personalized mashup, social networking sites, user model interoperation, authentication, privacy

1 Introduction

One of the most important features of Web 2.0 is that it is social: users can share content with their friends and can develop social ties among each other. Social features can be combined with domain-specific applications, e.g. a music application like LastFM, to empower a community of users. Reusing existing user model data from the domain-specific application (e.g. preferences for particular groups or music genres) can minimize the effort for users, allow useful adaptations and recommendations to be provided by other applications, and thus may help bridge the gap across their presence in different communities. Many researchers in the User Modeling field have investigated how to ensure User Model Interoperability (UMI) by exchanging user model data between applications. Web-based APIs and mashups provide an easier way to implement UMI. A mashup is a web- or desktop- application that combines information and/or services from one or more external sources [1]. Social networking site mashup applications combine user social data with some domain-specific application (e.g. music player/recommender, shopping, or mapping application). At the time of writing, there are more than 50,000 facebook mashup applications. There are two mashup application modes. The first one is where the mashup application runs inside the data provider's page within a frame or gadget, such as facebook application or OpenSoical gadget. This makes it convenient for the user, allowing interaction with several applications within the same website, avoiding duplication of effort when logging in. The second mashup mode is where the mashup application runs on its own web page. In this mode the user may have to log in in more than one application, if user data is shared among them. There are no significant differences between those models from developer perspective. In both cases, the mashup application actually runs on its own server. In first mode, the social network site is simply a proxy. It displays the mashup application's page within its gadget.

2 Current Technologies

A complete UMI framework must have four parts [4]: (I) user data exposing and discovery; (II) user identification mapping (III) authentication and privacy controls, (IV) user data exchange. A personalized mashup application as light-weight mashup application also has these four parts. The following sections will explain each part briefly. This research mainly focuses on the third part: authentication and privacy controls. To clarify the terms, we use "*data provider*" to denote the application or service which publishes an open API to share user model data; we use "mashup application" to denote the application which requests external user model data and uses it to adapt its own service.

(I) *User data exposing and discovery*: During this process, the data provider publishes user data APIs and information about the semantic and syntactic meaning of the data it provides. Currently, the mashup application developer has to discover data providers manually and read their APIs documents. There are some promising techniques to automate this process: SAWSDL (Semantic annotation of WSDL) http://www.w3.org/2002/ws/sawsdl/, SA-REST (Semantic annotation of REpresentational State Transfer) [7], XRDS (extensible resource descriptor sequence) and XRDS-simple (a simple and subset version of XRDS).

(II) *User identification mapping*: In order to use external user data, the mashup application has to know the user's identity in the data provider's system. Currently, the end user has to provide this information manually in the mashup application. However, there are some universal identity management platforms available. OpenID is the most popular one [5]. Open ID is a decentralized, interoperable, extensible platform for user-centric Internet identity management. OpenID provides users with a universal internet identity which can be used for many online applications. Right now, there are dozens of OpenID providers (Google, Yahoo, Flickr, AOL and etc) and users can choose the ones they trust as their identity providers. With a universal identification management, data provider and mashup provider do not need to map user's identity across two systems.

(III) *Authentication and privacy controls*: The user data is behind the lock of username and password [2]. In order to access user model data from a data provider, the mashup application needs to authenticate itself to the data provider. Here, access means read, edit, add or delete operation on user data. Authentication has two parts: first, validating the mashup application's identity, and second, validating whether the mashup application has the right to access user data. Validating the mashup application's identity is a relatively simple task. The current solution is through API key and secret key. In order to use the data provider's APIs, the mashup application needs to register at the data provider by presenting some basic information. After the registration, the data provider assigns a pair of API key and secret key to the mashup application. The API key and secret key is like username and password for identifying the mashup application to the data provider application.

Validating whether the mashup application has the right to access user data is a more difficult task. The current solution requires sending username and password of the user in the data provider's system. There are two ways to do the authentication. First, the mashup application directly asks the user's name and password of the data provider system and does the authentication itself. This is risky from a user point of view. Alternatively, the mashup application can redirect the user's web page to the data provider. On the data provider's web page, the user is required to login to authenticate him- or herself. After the user is authenticated, the data provider will inform the user that the mashup application is trying to access his or her data and will request permission to allow the mashup application to access the user data. If the user gives permission, the data provider will "callback", i.e. transfer the user's web page back to the mashup application; the data provider will also send a session key to the mashup application (see Fig. 1).

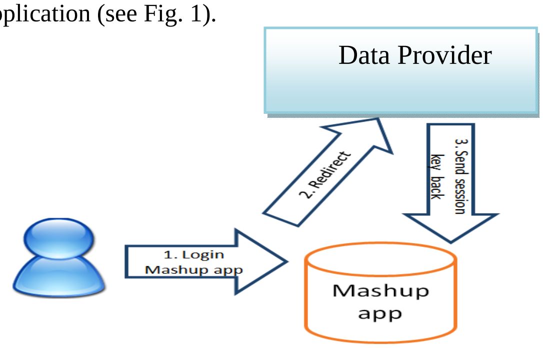


Fig.1. Safe Authentication Model

With this session key, the mashup application can access user data. This session key is used just for one user. Different data providers have their own rules about this "session" authentication. For some data provider, the session key may expire after hours and the user has to authenticate again. For other data providers, the session key may not expire. Some user data require authentication and some do not. This "session" mechanism is inconvenient for users when a mashup invokes several data providers, and needs to do many authentications to many user data providers. Before actually using the mashup application, the user has to authenticate with each data provider, and wait for the page redirecting. This authentication mode is not only inconvenient. It doesn't give users control over the user data interoperation process. Even though only data that is publicly available online is currently shared among applications, privacy concerns have been voiced and users are concerned about having little understanding or control over how data is shared. Users are unable to see which data is shared, how it is used, how long it is kept and have no control other than not adding the third party application (the mashup). The opaqueness of the user model data sharing process often makes users hesitant to use the available services.

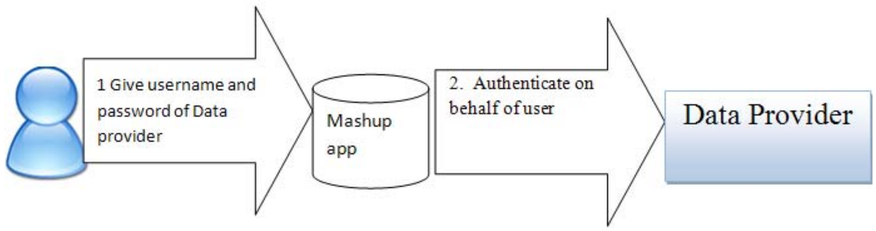


Fig.2 Risky Authentication Model

There are several mature authentication and privacy control frameworks address some of these issues, such as OAuth and Shibboleth. But there are still some limitations of these frameworks, which will be mentioned in section 3 related works.

(IV) *User data exchange*: this process is where user model interoperation really happens. Currently, Web-based API is the major technology used for this task. The most popular protocols of Web-based API are SOAP message and REST. Web-based API is reliable technology. Among the four parts of the user model interoperation, user data exchange is the most mature one.

3 Related Work

Berkovsky et al. [8] pointed out four major challenges for achieving UMI.
1. Systems are unwilling to share user models; 2. Privacy issues;
3. Technical considerations; 4. Semantic heterogeneity among applications

A lot of research has addressed the issue of semantic heterogeneity [4], [8], [9]. This research mainly focuses on the second challenge. There has been also a lot of research on privacy in user modeling [10], [11], [12]. Since the 1980ies, researchers have studied users' attitudes about internet privacy. They found out that users can be divided into three clusters [10], [12]:

1. Privacy fundamentalists, comprising approximately 17% of the entire user pool, generally express extreme concern about any use of their data and an unwillingness to disclose information, even when privacy protection mechanisms would be in place.

2. Pragmatic majority, approximately 56% of the entire user pool, are generally concerned about their privacy as well, but less than the fundamentalists. They are also far more willing to disclose personal information when they are see potential benefits and protection.

3. Privacy unconcerned, who takes 27%, of the entire group, tends to express mild concern for privacy.

In the recent decade, the number of privacy fundamentalists and privacy unconcerned is declining, and there is increase in the number of privacy pragmatic users [10]. In other words, privacy pragmatics is the majority of internet users and their number is still increasing. Therefore, most internet users care about privacy but are also interested in personalized services. As developers, we need to motivate users to disclose their data and protect their privacy. Previous research, e.g. [10], [12], reports about ways to motivate users to disclose their data. In order to motivate users to disclose their data, the application should tell users the benefits of personalization. Moreover, users want to know how their personal information is being used and to have control over this usage. Applications should be able to explain to users what facts and assumptions about them are being stored, and how these are going to be used. Users should be given ample control over the storage and usage of this data. Trust in a web site is a very important motivational factor for the disclosure of personal information. Trust is built on positive past experience, so applications should allow users to incrementally supply more information as their trust in the application increases.

Therefore, the authentication and privacy control mechanism should make the UMI process transparent to the user. This direction – to give the user control over which partial models should be made available to which applications - was suggested recently by Kay [14]. The user should be aware of the user model data required by a mashup application and the terms of use of the data. Based on this information, the user can decide whether or not to allow the mashup to access and use user data. As mentioned before, there are some frameworks that attempt to achieve that: such as OAuth and Shibboleth. OAuth is an open protocol to allow secure API authorization in a simple and standard method [2], [5]. It is a light-weight framework which has already been adopted by some social networking sites, like Twitter. But OAuth cannot let the user decide how to do the authorization. For example, when the user trusts a mashup application and feels comfortable about letting it access his or her data, the user does not want to be involved in the authorization (it is viewed as an extra burden). The Shibboleth protocol is another mature framework which ensures safe user data sharing between systems [13]. The user can define an attribute release policy to each outside system which requires user data. There are many prerequisites for using Shibboleth: the system must have secure identity management and must install the required software. Shibboleth is ideal for universities and other larger organization.

We propose a new authentication and privacy control mechanism. This mechanism can facilitate privacy control by letting users customize their privacy settings depending on each individual mashup application and their different privacy preferences. Moreover, the user can decide how to do the authorization. This mechanism does not deal with data provider discovery or semantic heterogeneity directly, but it can be integrated with other mechanisms to achieve a complete user model interoperation framework.

4 Authentication and privacy mechanism

As mentioned in the introduction, there are two mashup application modes. In both modes, mashup applications need authentication.

4.1 Application registration

When a mashup application registers at a data provider, the mashup application needs to list all the user data it will access during the service and the type of action on the data: such as read, edit, add, and remove. Besides that, the mashup application also needs to describe the terms of use of the user data, and information about who provides this mashup application. The mashup application is not able to access any user data which are not listed in the registration. The registration information is visible for the user. Therefore, the user knows what kind of data will be used by the service. When the user wants to use a new mashup application, the user data provider will show this application's registration information to the user.

4.2 Authorization

When the user invokes the mashup application for the first time, the mashup application will redirect the user to the data provider. The data provider will ask the user to login. After user login, the data provider will show the registration information about the mashup application (as shown on Fig. 4) and ask the user if he or she wants to authorize the mashup application. The user can grant the mashup application one of three levels of access. The first level is access *without user authentication*, i.e. the mashup application can access the user data it registered without user authenticating. This would be very convenient for the user since it will require no further effort for authentication; however, it gives the mashup application the right to access the user data it registered whenever it wants. The second level of access is *single authentication*. When the mashup application requests user data, the mashup application needs to redirect the user to the data provider, and the data provider will ask the user to authenticate him or her. After that, the data provider will ask the user whether he or she authorizes the mashup application to access all the user data that the mashup application has in its registration file. The user can choose the time period of authorization: for example, 1 hour, 3 hour, or 24 hours. Within that time limitation, the mashup application can access any user data in its registration file. The third level is *individual authentication*. The user can specify which user data require an individual authentication, so when a mashup application tries to access this data, it will always require user authorization.

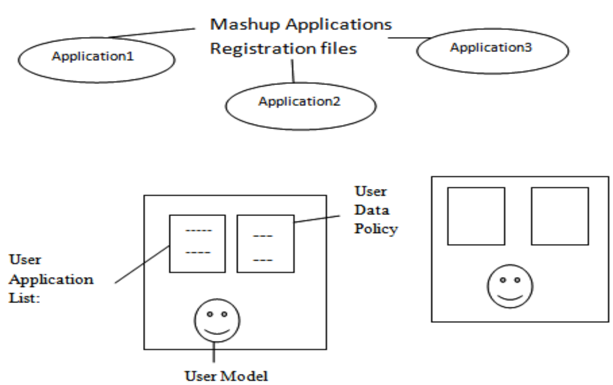


Fig. 4. The Components of the mechanism

4.3. User applications list

Each data provider hosts a list of mashup applications which have requested to receive user data for each user about whom the data provider keeps a user model. In that list, the user can see overview of the all mashup applications that the user has authorized. The user can discontinue or change the authorization at any time.

4.4 User Data Policy

To facilitate user decision making, the user can save his or her privacy settings. The User Data Policy is a file hosted at the data provider system. It has two parts: *policy about application providers* and *policy about data usage*. The *policy about application providers* contains a list of trusted application providers and a list of blocked application providers. Note that this list contains providers, not individual applications. Users can view and manipulate these lists based on external information, e.g. provider reputation services, press, etc, which may change their level of trust in particular application providers. Some providers, of course, may be unknown to the user, and not be in either of the two lists. In the *policy about data usage*, users can classify the data kept about them by the data provider into three levels: open-level, important-level, and crucial–level. The open-level data is accessible to all application providers except those in the blocked provider list. The important-level data is only open to the application providers on the "trusted" list. The crucial-level data is not to be undisclosed to any providers. The user can change both parts of the User Data Policy at any time. The purpose of the User Data Policy is to facilitate authorizing new mashup applications. When a user is authorizing a new mashup application, the data provider system can automatically compare the mashup application registration information with the User Data Policy to see whether there is a conflict. For example, the mashup application requires important-level data, but the application provider is not on the user's trusted list. If there is no conflict, the application will be authorized. Otherwise, the user data provider system will inform the user about the conflict, and the user can decide whether to change the policy (add the application provider to the trusted provider list) or not authorize the application. If the mashup application requires crucial-level data, the user has the choice of rejecting the authorization or still allowing it by changing the user data policy by moving the data to important level and adding the application provider to the trusted list.

4.5 Update application

Mashups can change their requirements for user data at any time. An updated mashup application has to update its registration information at all user data providers from which it receives user data. Data providers maintain version control on mashup application registration and user application list. The registration file of a mashup application keeps a version number for the application. In the user's application list, the mashup applications are also listed with their version numbers. When a mashup application updates its registration information, the data provider will increase the mashup application's version number. So, the version numbers for this mashup in its registration and in a user's application list will not match anymore.

4.6 Overall Workflow

Every time, when mashup application tries to access user data from the data provider, the data provider will check if the versions of the registration and user's application list match. If they do not match, the data provider will request user authorization. The mashup application will redirect user to the data provider, the data provider will show the update of the mashup registration to the user, and ask for authorization. If the user authorizes the updated mashup application, the version number on the user's application list will be updated according to the registration information. When a mashup application requests user data, first the data provider will check the mashup application's API and secret key. After that, the data provider will check whether the user data mashup application has requested is listed in the mashup application's registration. After that, the data provider will check whether the mashup application is on the user's application list. After that, the data provider will check if there is a match between the versions of the mashup application registration information and the user's application list. In the final step, the data provider will check the authorized access rights of the mashup from the user's application list.

5 Discussion

The proposed mechanism is user-centric; the user can see what kind of data is required for the mashup application and can authorize the mashup application's access to user data in a flexible way. The proposed mechanism refines the authentication process. The user can control his/her level of involvement in the authentication. If the user trusts the mashup application, he or she does not need to be involved in authentication at all. If user wants, he or she can control each step of user model interoperation. The user can chose to control only on the sensitive data's interoperation. Comparing with Shibboleth, this mechanism is light-weight; it does not require installing any software. It is ideal for small and middle-level application.

This mechanism also has some limitations. It makes the authorization process more complex. It requires more user involvement the first time when the user uses a mashup application. It also puts limitations on the mashup application development. Developers have to openly declare what kind of user data is required. Developers of applications that share user data and serve as data providers have to implement the components of the mechanism (see Fig. 4): a component that receives and updates the registration files of mashup applications, the user application list and user data policy, as well as an interface for the user to view and modify the user application list and user data policy.

The impact of mashup performance is not clear yet. If the user grants the mashup application the highest access rights, the performance should be the same as without the mechanism. But if user requires individual data authentication, the performance would be worse. Yet the user may be willing to accept the worse performance in exchange for enhanced privacy. The scope of the mechanism does not allow it to enforce how the mashup application treats user data. In the registration, the mashup application has to declare how it is going to treat the data: how long it will keep it, whether it will transfer the data to other parties or not, if it will disclose the data to other users or not. However, this mechanism cannot enforce the mashup application's compliance to its own declaration. Trust and reputation management mechanisms can be used as an orthogonal approach for ensuring that mashup application providers have an incentive to treat user data according to the registration.

Finally, social network systems and existing mashup applications on the social web store also a lot of user-contributed data that can be used later in data-mining to develop new user profile data, not explicitly represented at the moment of sharing. It is an open question how to handle the potential privacy threats arising from harvesting user-contributed data.

The data about the user's social network presents further issues. So far we were talking about exchanging user data across applications only, but these applications typically have many users. Will these users be allowed to see the user's data or not? Can users define rights for accessing data to their friends / social network that can be propagated from one application to another? Some work on sharing data in blogs addresses this issue [14].

6 Future Work

We have implemented the proposed mechanism in a mock-up social network site environment. We plan to design several scenarios involving some sensitive user data and do an evaluation of the mechanism with real users based on these scenarios. There are several hypotheses we want to test during the evaluation: First, that the user data that can be shared is shown to the user in an understandable way. Second, that the user can easily express his or her privacy control settings through the User Data Policy. Third, that the user understands from the mashup application registration file (displayed in an appropriate way) why the application needs his or her model, the benefits for user model interoperation and how the application treats the user data. Fourth, this mechanism should help to increase user participation with respect to adding new mashup applications in an experimental group that uses the framework, in comparison with a control group which use the traditional authentication and privacy mechanism. We hope to be able to test these hypotheses with a large number of users on a social network site and will use questionnaires and collect statistics about user's participation that will be analyzed to validate or refute the hypotheses. In the next stages, we will combine this mechanism with services for user model data semantic translation, service discovery, and user identity mapping mechanisms to achieve a complete user model interoperation framework.

7 Summary

Personalized mashups provide a new way to do user model interoperation. Current mashup solutions face several challenges, including insufficient authentication and privacy control. This paper proposes a user-centric mechanism to facilitate authentication and improve user privacy control. Sharing user data on the social web raises many important issues. This mechanism addresses the privacy of sharing user model data that is explicitly represented by the application.

References

1. Thang, M.D., Dimitrova, V., Djemame, K.: Personalised Mashups: Opportunities and Challenges for User Modelling. In Proceedings of User Modeling. 2007, 415-419. (2007)
2. Musser, J.: Open APIs: State of the Market presentation. Qcon San Franciso (2008) http://qconsf.com/sf2008/file?path=/qcon-sanfran 2008/slides//JohnMusser_Web_As_Platform.pdf
3. Liu, H., Mae, P.: InterestMap: Harvesting Social Network Profile for Recommendations. Proc. 10th International Conference on Intelligent User Interfaces (2005)
4. Carmagnola, F., Dimitrova, V.: An Evidence-Based Approach to handle Semantic Heterogenity in Interoperable Distributed User Model. Proc. 13th International Conference on Intelligent User Interfaces (2008)
5. Recordon, D.: "Blowing up" Social Networks by Going Open (2008) http://www.slideshare.net/daveman692/blowing-up-social-networks-by-going-open-presentation
6. A.P. Sheth, K, Gomadam, J. Lathem.: SA-REST: semantically interoperable and Easier-to-Use services and Mashup. IEEE Internet Computing, vol.11, no.6, 2007, pp.91-94 (2007)
7. Reed, D., Chasen, L., Tan, W.: OpenID identity discovery with XRI and XRDS ACM International Conference Proceeding Series, Vol.283 Proceeding of the 7th symposium on identity and trust on the Internet Pages 19-25 2008, ACM, New York,USA (2008)
8. Berkovsky, S., Kuflik, T., Ricci, F.: Mediation of user models for enhanced personalization in recommender systems. In Proceedings of User Model. User-Adapt. Interact. 245-286. (2008)
9. Heckmann, D., Schwartz, T., Brandherm, B., Kroner, A.: Decentralized User Modeling with UserML and GUMO. *Proceedings 10th International Conference on User Modeling*, (2005) LNAI 3538 Springer, Berlin-Heidelberg, pp.428-432
10. Kobsa, A.: Privacy-Enhanced Personalization. Communication of ACM, vol.55, pp24-33 ACM, New York, NY, USA (2007)
11. Anwar, M.,Greer,J.: Role and Relationship-Based Identity Management for Private yet Accoutable E-Learning, Trust Management II, vol.262, pp.343-358 Springer ,Boston (2008)
12. Cranor,L.F., Reagle,J., Ackerman, M.S.: Beyond Concern: Understanding net users' attitudes about online privacy. AT&T labs-research technical report TR 99.4.3, http://www.research.att.com/library/trs/trs/99/99.4/. (1999)
13. Macquarie E-Learning Center of Excellence (MELCOE), Macquarie University: Shibboleth , October-November 2008 , AAF Shibboleth Rollout Workshop. http://www.federation.org.au/workshop/AAF%20Shibboleth%20Rollout%20Workshop%202008.1.pdf
14. Kay,J.: Lifelong Learner Modeling for Lifelong Personalized Pervasive Learning. IEEE Transactions on Learning Technologies Volume 1, Issue: 4 (2008)
15. Indratmo., Vassileva,J.:A Usability Study of an Access Control System for Group Blogs. Proc. ACM International Conference on Weblogs and Social Media. Boulder, CO (2007)

Experiences from Implementing Collaborative Filtering in a Web 2.0 Application

Wolfgang Woerndl, Johannes Helminger, Vivian Prinz

TU Muenchen, Chair for Applied Informatics – Cooperative Systems
Boltzmannstr. 3, 85748 Garching, Germany
{woerndl, helminger, prinzv}@in.tum.de

Abstract. The goal of this paper is to report our experiences from integrating item-based collaborative filtering into the Web 2.0 site linkfun.net. We discuss the necessary steps to implement the selected Slope One algorithm in our real world application. It was necessary to conduct performance optimization to allow for recommendations without any delays in page generation on our site. Firstly, we significantly reduced the data model by including only items similarities for pairs of items where both items been rated by at least k users. Secondly, we precomputed recommended items for users. By analyzing the empirical results, we found out that user activity increased on the site after introducing the recommender. In addition, users rated recommended videos higher on average than others which indicates that the recommender allowed users to find preferred videos more effectively.

Keywords: recommender systems, collaborative filtering, slope one, performance optimization

1 Introduction

Web 2.0 applications such as MySpace, YouTube or FlickR have gained much interest in industry and academia in recent years. Users can easily upload content such as videos, photos or links in order to share them with others. However, most current sites lack structured intelligence and finding meaningful information can be difficult [1]. Recommender systems and collaborative filtering are techniques to deal with this problem as they filter information items according to a user's needs and taste.

In this project, we are investigating the integration of a collaborative recommendation system in the real world Web 2.0 site *www.linkfun.net* (Fig. 1)[1]. This site allows users to share links to funny content such as videos. While other sites like YouTube feature all kind of videos, linkfun.net focuses on humorous content. In addition, linkfun.net does not host the videos itself, but users provide links to content on various other sites. The overall goal of the work presented in this paper was to provide good recommendations to users and thus increase user interest in the site and expanding the community and value of linkfun.net. Notable requirements included the

[1] The user interface of linkfun.net is currently in German only

integration of the recommendation without decreasing the performance of the site, extra hardware needs or additional effort for the users.

Fig. 1. Screenshot www.linkfun.net

The rest of this paper is organized as follows. The next section gives some background on recommender systems and discusses which type of system appears well suited for our scenario in principle. In section 3, we explain the necessary steps to integrate the filtering functionality in linkfun.net. In section 4, we present the empirical results from the analysis of log files. Finally, we conclude the paper with a short summary and an outlook.

2 Recommender Systems and Collaborative Filtering

The basic idea of recommender systems is to recommend products like books and CDs and other items such as restaurants or videos to an active user. To do so, the system computes the chance that a user likes an item. This is based on information about the user the items and possibly other data such as contextual information. Characteristics of recommender algorithms include the quality of recommendations, storage and runtime complexities, anonymity and extensibility of the model.

2.1 Individual Recommender Systems

In general, we distinguish between individual and collaborative recommender systems. Individual recommenders determine fitting items based on the profile of the active user. Thereby, the system matches explicitly entered or implicitly observed user preferences and interests with items meta data. Hence, this type of recommender system is often called content-based recommender system. One way of implementing this kind of recommender is to use a rule system. However, the content-based approach is not well suited for Web 2.0 content because additional information about the items and/or users is required. Moreover, an individual recommender does not fit the social and collaborative nature of Web 2.0 applications.

2.2 Collaborative Recommender Systems

The second category of recommender systems are based on collaborative filtering (CF). CF utilizes the ratings of other users for items, for example a rating on a scale of 1 to 5. The vector of all ratings of a user for various items is called a user's rating vector. CF seems appropriate for Web 2.0 because it needs no information about items and implements the "word of mouth" idea that is also prevalent in Web 2.0. Users like to express their opinions on content and basic rating schemes already exist in some sites.

We differentiate between two variants of CF, user- and item-based collaborative filtering. The recommendation process of user-based CF basically consists of two steps. First, neighborhood creation: Determine a set of k users that have rated similarly to the active user in the past. Second, recommendation of new items for the active user. For neighborhood creation, the active user's rating vector is compared to the vectors of all other users. To do so, different metrics have been proposed in the literature, for example Euclidean distance, cosine similarity or Pearson-Spearman correlation [2]. Thus, user-based CF analyzes the available raw data, namely the user-item matrix of ratings. In the second step, the algorithm selects items, which the active user has not rated yet, but which have been rated positively in the neighborhood of the active user. User-based CF has proven very useful and accurate in applications such as Web shops. However, this type of collaborative recommender has several drawbacks. First of all, there is the new user problem. User-based CF cannot generate a suggestive recommendation if the active user has not rated any items. In a Web 2.0 site such as linkfun.net, new or occasional users may represent a high share of the user base. A second relevant problem of user-based CF is that the approach is computationally costly. The approach operates on the raw data of ratings, which have to be analyzed each time a prediction is computed.

2.3 Item-based Collaborative Filtering

The alternative approach is item-based CF. Item-based CF does not consider the similarity of users, but of items [3]. Thus, the user-item matrix is not analyzed line by line, but column by column. One significant difference to user-based CF is the independence from who the active user is: the item similarities can be precomputed to build an item-item matrix. The item-item matrix is the model of the algorithm. Therefore, this type of recommendation algorithm is also called model-based collaborative filtering. One element $S_{i;j}$ of the item-item matrix expresses the similarity between items i and j, determined from the users' ratings. The ratings vector of the active user is then used to recommend items that are similar to items the active user has rated positively in the past.

It is important to note that item-based CF has little in common with individual, content-based filtering. This is because the users' ratings are solely used for computing the item similarity. Meta data of items is irrelevant. Item-based CF has an advantage over user-based CF with regard to the complexity of the computation. The item-item matrix can be calculated as an intermediate result, independently from the active user. The main advantage of item-based CF is performance, because generating recommendations from the model instead of the raw data of ratings is much more efficient.

Slope One [4] is an example of an item-based CF algorithm. The main idea of the approach is to use differentials of ratings and store them in the item-item matrix. Slope One uses predictors of the form $f(x) = x + b$, which precompute the average difference between the ratings of two items for users who rated both items [4]. Consequently, the model can be updated on the fly, without the need to recalculate the model when a rating is made. In addition, it is not demanding as much information from new users. One rating is enough to be able to generate recommendation for a user. Finally, [5] describes a straightforward implementation in PHP using a SQL database. Linkfun.net was also implemented in PHP and SQL, so we decided to base our recommender on Slope One. More details on Slope One and our implementation are given in the next section.

3 Data Model, Recommendations and Optimization

In this third section of the paper we explain the design decisions when integrating Slope One into linkfun.net and implementing the collaborative filtering method.

3.1 Data Model

The initial situation in linkfun.net was that users were able to give ratings on a scale from 1 to 5 with 5 being the best grade. However, the application did only save the aggregated average value for each item. Information about the individual ratings of users was not kept. Slope One and all other CF algorithms do need the detailed user-item matrix though.

Hence, the existing ratings had to be discarded and a new data model had to be designed. This new model consists of three database tables to store the necessary information:

- Table `rating` to store the user ratings for items, with one rating corresponding to one row in this table.

- Table `dev` for the average deviations of ratings for an item-item pair [5]. This table implements the item-item matrix or, in other words, the model of Slope One.
- Table `rating_recom` to log when a video was recommended to a particular user and optionally how the user rated it after the recommendation.

Every time a user *u* rates an item *i*, our system updates `rating` and calculates the impact on `dev`. To do so, the algorithm first determines all items u has rated, computes their rating differentials to *i* and updates the affected entries in `dev`.

3.2 Generating Recommendations

The model can then be used to generate and display the top five recommended items – primarily videos – when a user accesses linkfun.net. Slope One distinguishes between non-personalized and personalized recommendations [5]. Non-personalized recommendations are not based on collaborative filtering in the strict sense. After a user rates just one item *i*, the algorithm searches for items, which have the highest rating differential to *i*, i.e. were rated best in comparison to *i*.

To predict the rating for a particular item *i* for an active user *u*, the algorithm first selects the set of items *rs*, which were rated by *u* and also by at least one other user. In the second step, the differentials between the ratings of *i* and the items in *rs* are determined using the item-item matrix, respectively our database table `dev`. Finally, the differentials are summed up and divided by the number of items in *rs*. This results in a predicted rating for *i*. Note that the predicted rating is possibly higher than the highest grade, for instance 5 on a 1-5 scale. This is because Slope One works on rating differences [4]. However, it is only important to compare the predicted values of items to each other to be able to generate a ranked list of items for the personalized recommendations.

3.3 Optimization

The number of elements in the item-item matrix – i.e. the table `dev` – soon grew to over 3.4 million entries with about 2000 items (Table 1). This is due to the fact that with more ratings of users, more and more items receive at least one rating and more and more similarities between item pairs can be calculated. This led to delays in handling ratings and computing recommendations. It hurt the overall user experience on linkfun.net because page generation was noticeably slowed. This necessitated performance optimization.

We implemented two solutions to improve performance:

1. Reducing the data model and the number of entries of the table `dev`
2. Precomputing recommended items for users

Table 1. Reducing the data model

	Number of entries in `dev`	Available videos for recommendation
No threshold	3404049	1961
k=2	2436997	1885
k=3	1397523	1759
k=4	*605561*	*1555*
k=5	198889	1278
k=10	43	19

The basic idea to minimize the size of the item-item matrix is to only compute and store item similarities for pairs of items where both items been rated by at least *k* users [5]. The obvious drawback is that some items, which have received few ratings, are not represented in the item-item matrix anymore. Thus, these items are no longer available for recommendation. Table 1 shows the number of entries in `dev` and the number of available videos for certain values of the threshold k of necessary ratings. At that point of time, a reasonable value for *k* was *k=4*. This value reduces the numbers of entries in `dev` significantly from 3404049 to 605561 while keeping about 80% of items available for recommendations.

Despite this reduction, generating recommendations when a user hits the corresponding link took still too much time. Hence, the recommendation had to be precomputed. To do so, we created a new database table `precomp_recom` that stores five recommended items for every user. This table is updated according to the following schedule:

- Once per day, the recommended items for all users are recalculated. This procedure takes about 15 minutes. It is performed during the night when less users access the site.
- The recommendations are updated every 5 minutes for users that have rated items since the last update.

The second condition ensures that recommendations are updated promptly and regularly for active users and reflect their latest ratings. In any case, the model, i.e. the item-item matrix as basis for the recommendations, may be slightly out of date. Yet this fact has to be accepted to allow for instant recommendations by precomputing them.

In general, CF algorithms may suffer from cold start problems with new users or new items. For example, new items cannot be recommended until they receive at least one rating. This was not a problem in our case. Most of our items were rated within hours and thus were potentially considered for recommendations reasonably soon. As far as new users are concerned, newly registered users to linkfun.net were shown a message that they need to rate at least one video to obtain recommendations.

4 Empirical Results

In this section, we analyze the log files to evaluate the effect of the CF function on linkfun.net. We were in particular interested in two questions:

- User activity: what were the effects of the recommender system on user activity?
- Quality of recommendations: were recommended videos rated higher on average than other items?

4.1 User Activity

At time of research, there were 490 registered users of linkfun.net, although some of the registered users frequented the site rather seldom. 150 users rated at least one item and 50 users actively used the recommendation function. Overall, there were 10500 ratings and 100000 times a video was played by users. About 60% of the video playbacks occurred after using the recommendation function.

Table 2. User activity with regard to rating items

Total number of users	**490**	**100%**
Users with more than 1 rating	150	30,6%
Users with more than 5 ratings	87	17,8%
Users with more than 10 ratings	67	13,7%
Users with more than 20 ratings	54	11,0%
Users with more than 50 ratings	30	6,1%
Users with more than 200 ratings	12	2,4%

Table 2 shows the distribution of the number of user ratings. The table shows, that a rather small user base (“early adopters”) contributed to most of the ratings. Twelve users have accounted for about 8600 ratings or roughly 75% of all ratings.

For overall user activity we looked at the visitor logs after the recommendation function was introduced. Fig. 2 illustrates the sessions and page visits per month over the course of the research period. While the number of sessions increased only gradually, we noticed a far bigger increase in page impressions. This means that the average time users spent on the site grew considerably. Although we are not able to measure the exact impact of the recommender on site activity, the overall goal of increasing user activity was met.

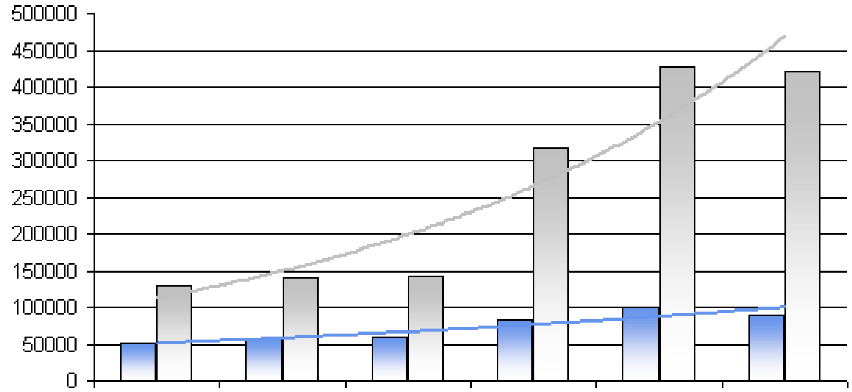


Fig. 2. Sessions (left bars) and page visits (right bars) per month

4.2 Quality of Recommendations

As far as the quality of recommendations is concerned, we investigated the ratings the users gave to the videos. Overall, the videos were rated high on average with many videos receiving the best grade. This may be due to the fact that linkfun.net is a specialized community where users only provide links to funny content that may cater to a similar taste. We noticed a trend that this high rating average increased even further after the introduction of the recommender function (Fig. 3). Our assumption is that the rating and recommendation scheme allowed users to find preferred videos more effectively.

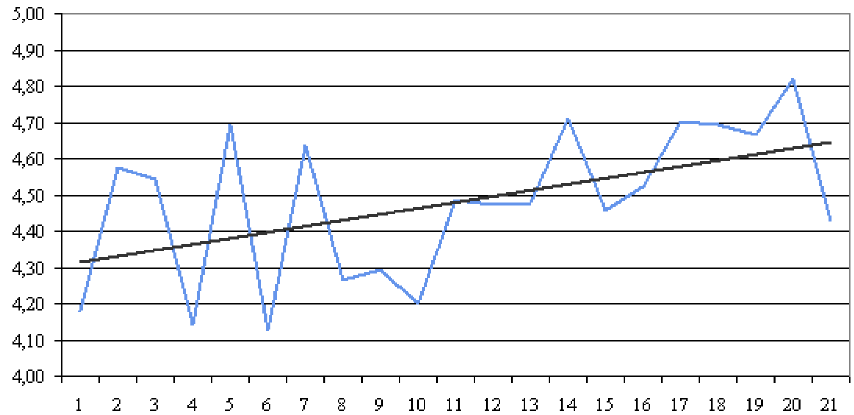
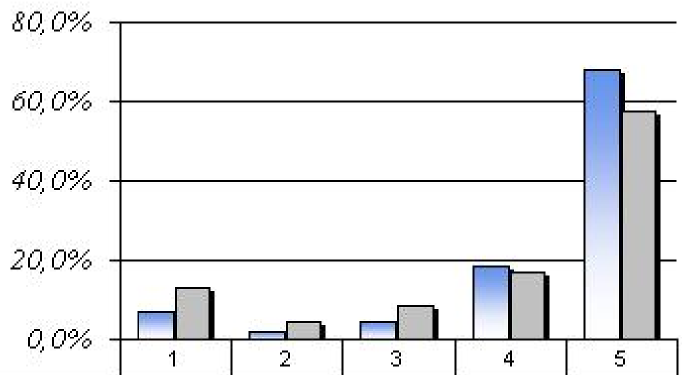


Fig. 3. Trend of average ratings per week

To evaluate the quality of recommendations in more detail, we compared the ratings of videos that were recommended to a user with the ratings of videos that were not in the list of recommendations. The latter category consists of videos accessed from the homepage of the site or from a “newest videos” section. We found out that recommended videos received higher grades: the average rating was 4.4 in comparison to 4.0 for non-recommended videos.



	1	2	3	4	5
Recommended	7,0%	2,1%	4,4%	18,7%	67,9%
Non-recommended	12,9%	4,5%	8,5%	16,8%	57,3%

Fig. 4. Distribution of ratings: recommended videos (left bars) vs. non-recommended videos (right bars)

Fig. 4 shows the distribution of ratings for the videos in more detail. For example, 67,9% of recommended videos were rated with the top grade (“5”), while the percentage of top-graded non-recommended videos is significantly lower (57,3%).

5 Conclusion

In this paper, we have described our experiences from integrating the item-based collaborative filtering algorithm Slope One into the Web 2.0 site linkfun.net. We explained the necessary steps including the data model. After we have done some performance optimization, the recommender function ran smoothly on the site, without any delays in user experience or additional hardware requirements. The performance optimization included reducing the data model of item similarities and precomputing recommended items for users. Overall, the Slope One algorithm proved to be very practicable and fitting for the examined site. By evaluating our implementation, we found out that user activity increased on the site after introducing the recommender. In addition, users rated recommended videos higher on average than others.

As far as related work is concerned, there are several applications which also use SlopeOne in practice. For example, InDiscover (http://www.indiscover.net) is a site for promoting independent musician and recommending new music to interested customers. Their system uses RACOFI which is a framework for rule-based collaborative filtering partly based on Slope One [6]. However, there is no published information about performance optimization or empirical results. There is plenty of research in improving recommender systems, mostly with a focus on prediction quality [7], but there are few reports on experiences from applying recommender algorithms in practical Web 2.0 applications. Leimstoll and Stormer discuss in [8] how collaborative filtering can be integrated in online shops in principle. Their proposal is somewhat similar to our approach, although no experiences from real world applications are reported.

One goal of our planned future activities is to integrate implicit ratings that can be observed from user behavior. So far, all recommendations are based on explicit ratings users have made. We are currently investigating methods to measure the exact time a user spends with a video. When a user is watching a video until completion, one can assume that she liked the video, which relates to a good rating. If the user cancels the playback soon after the start, we would assign a low rating. Subsequently, we want to measure whether recommendation based on these implicit ratings derived from user observation performs as well as the explicit user ratings.

References

1. Lin, K.-J.: Building Web 2.0. IEEE Computer, vol. 40, no. 5, pp. 101-102 (2007)
2. Adomavicius, G.; Tuzhilin, A.: Towards the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. IEEE Transactions on Knowledge, and Data Engineering, vol. 17, no. 6 (2005)
3. Sarwar, B.; Karypis, G.; Konstan, J.; Riedl, J.: Item-based Collaborative Filtering Recommendation Algorithms. Proc. 10th International Conference on World Wide Web (WWW10), Hong Kong, China (2001)
4. Lemire, D., MacLachlan, A.: Slope One Predictors for Online Rating-based Collaborative Filtering. SIAM Data Mining (2005)
5. Lemire, D., McGrath, S.: Implementing a Rating-Based Item-to-Item Recommender System in PHP/SQL. Available at: http://www.daniel-lemire.com/fr/documents/publications/webpaper.pdf (2005)
6. Anderson, M., Ball, M., Boley, H., Greene, S., Howse, N., Lemire, D., McGrath, S.: RACOFI: Rule-Applying Collaborative Filtering Systems. IEEE/WIC COLA'03, Halifax, Canada, (2003)
7. Pu, P., Bridge, D., Mobasher, B., Ricci, F.: Proceedings of the 2008 ACM Conference on Recommender Systems (RecSys ’08). ACM (2008)
8. Leimstoll, U., Stormer, H.: Collaborative Recommender Systems for Online Shops. AMCIS 2007, Keystone, CO (2007)

A Collaborative System Based on Reputation for Wide-Scale Public Participation

Ana Fernández[1] and Jens Hardings[1]

[1] Departamento de Ciencia de la Computación, Pontificia Universidad Católica de Chile
Vicuña Mackenna 4860 (143), Macul, Santiago, Chile
{asfernan, jhp}@ing.puc.cl

Abstract. The aim of this paper is to use a reputation system to promote trust among participants of an online social network. By the use of past behaviour and ratings from other users, this paper presents a novel metric to compute the reputation of peers. Also a prototype and deployment results are included.

Keywords: Reputation system, public participation, e-democracy.

1 Introduction

This article proposes the use of a reputation system to promote trust among participants and the system construction over an online social network.

Although reputation systems are being used in several fields, it has not yet been implemented in politics and citizen participation. We have not found other attempts of reputation systems for the world of politics in order to promote participation and e-democracy between peers.

Reference [1] defines e-democracy as the sum of acts realized by individuals or groups in order to influence the way the political system operates. Due to the Internet, citizens can easily access political content and such an increased access to political information should extend governmental transparency and thus democracy.

In reference [2], a proposal over the Internet where players have to cope with uncertainty from quality of products and trustworthiness of participants is presented. The method to address this predicament is to use feedback ratings about past behaviour to make recommendations about who to trust.

In reference [3] a proposal for the use of reputation systems in Communities of Practice (CoPs) was presented in order to assist users in creating relationships for honest and useful participation, based on trust, for the benefit of the entire community. Indeed [3] presents a simple reputation calculation based only in the median of past reputations.

In [4] we have used reputation systems in a Mobile Ad hoc Network (MANET) which is a low complexity system. But in this paper we offer a generalisation of the use of reputation systems to a more complex framework represented by the world of politics with the aim to promote participation and e-democracy between peers.

The paper is organized as follows: section 2 presents reputation systems concepts and issues; section 3 presents the proposed reputation model with its respective reputation metric; section 4 presents the deployed system; and finally, section 5 discusses our conclusions.

2 Reputation Systems

Many interactions in real world are based on rumours or on friends' experiences. As a result of this, future interactions can be influenced by past interactions. We call this the reputation of a user. Keeping that in mind we can build a system that collects, processes and distributes information about the quality of interactions. Referring to [5], we call such system a "reputation system". Reputation systems are well suited for stimulating social control within online communities. The idea is to let parties rate each other and use those ratings to derive a reputation score, which can assist other parties in deciding whether or not to transact with that party in the future [6].

Reputation systems need models in order to calculate the reputation of its users, that is, a way to obtain a qualification for each individual, using information stored in the system. Many reputation models have been proposed for online environments systems throughout the past years, but there is not an accepted common model yet.

3 Proposed Reputation Model

In our model we identified several factors that influence on the reputation of a user in the system which will be described as follows.

Whenever a user participates in the system he should be rewarded. A good way to measure the participation is by the relative contribution factor which will be the amount of actions executed by a user over the amount of total actions. We will denote C_i^P as the relative contribution factor for participation which has been divided in m areas, where m represents the amount of participation dimensions measured by the system, and its values will satisfy $0 \leq C_i^P \leq 1$ for $i = \{1,...,m\}$. Each contribution should have different importance in the system, for such reason we will identify β_i as the importance weight of C_i^P which values will go between 0 and 1. We then define the participating reputation R_P of user a as:

$$R_P(a) = \sum_{i=1}^{m} \beta_i C_i^P(a) \quad \textbf{(1)}$$

Certain users have the ability to generate participation in others and such ability should be rewarded by the system. In a similar way to participating reputation, C_i^L represents the contribution factor for leadership which will be sorted out in n different areas, where n represents the amount of leadership dimensions and its values will satisfy $0 \le C_i^L \le 1$ for $i = \{1,...,n\}$. We will define δ_i as the weight of C_i^L in the system which values will go between 0 and 1. The leadership reputation R_L of user a will then be defined as:

$$R_L(a) = \sum_{i=1}^{n} \delta_i C_i^L(a). \quad \textbf{(2)}$$

Users in the system can be qualified by others for a performed activity. Agent a will be rated and given a qualification $q \in Q$ where $Q = \{1,0\}$ which represent a positive or negative qualification respectively. Q_a represents the time-sorted list of qualifications of user a assigned by other users where $Q_a[1]$ is the oldest rate and $Q_a[h]$ is the most recent. Each user in the system will have an ordered list used to store his last h qualifications. When a new qualification $h+1$ arrives, the oldest one comes out of the list like a FIFO array.

Agents will behave more probably like they did in their most recent transactions. Therefore we chose a metric called BlurredSquared [7] which computes a weighted sum of all ratings. The older a rating is, the less it influences the current reputation. In our particular case the reputation will only be calculated with the last h qualifications.

The peer reputation R_Q of user a will then be defined as:

$$R_Q(a) = \sum_{j=1}^{h} \frac{Q_a[j]}{(h - j + 1)^2}. \quad \textbf{(3)}$$

The chosen model is based in the one described in [8]. The essential distinction between that metric and ours is that this novel metric considers qualifications from other nodes assigning more importance to the most recent ones.

We will define F as a function that determines the level of recent activity of a certain node. Let $T(a)$ be the residential time of user a in the system and let k be a discount factor between 0 and 1 that will be chosen in order to decrease the level of participation when the time spent in the system is higher and increase it when it is shorter. The level of recent activity for user a will be:

$$F(a) = R_P(a) \cdot R_L(a) \cdot k^{T(a)}. \quad \textbf{(4)}$$

Our model computes the global reputation or trust of a user based on two factors: past qualifications and level of recent activity. Trust for user a will be calculated as:

$$Trust(a) = \frac{R_Q(a)^{1+F(a)} - 1}{R_Q(a) - 1}. \quad \textbf{(5)}$$

4 Deployed System

The proposed system was implemented in the Alumni Center of the Faculty of Engineering of Universidad Católica de Chile using the well-known social network Facebook. Such implementation offers a participation platform for students as it permits them to express their concerns and ideas and allows others to vote or comment about them. The previously described model was applied in order to determine the improvement of trust among peers.

Figure 1 shows the evolution of trust for several users. Initially all users begin with the same trust value. Their behaviour in the system and the qualifications assigned by others determines the progress of their trust. User 1 has an increasing participation and leadership reputation as well as a good reputation among other peers; therefore his level of trust increases significantly over time. User 2 presents a decreasing participating reputation but an incremental leadership and a high reputation. Finally User 3 has a poor participation in the system and is not well qualified by others for that reason it presents a decreasing trust value as time passes by.

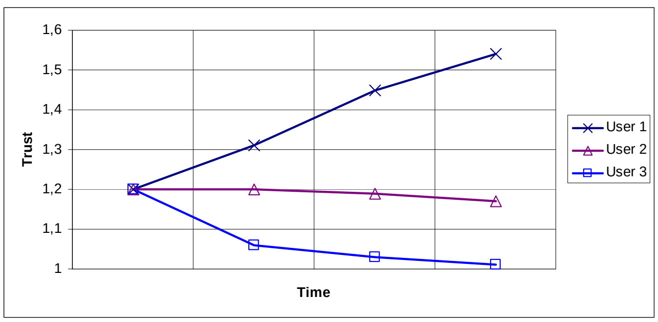


Fig. 1. Evolution of trust for different users.

5 Conclusions

This paper reflects the behaviour of a user in the system and the quality of his contribution in his trust value. As a direct consequence arises the fact that users with high level of trust are comparatively better users and therefore will eventually come with good ideas that could be used as part of future policies in the world of politics.

From simulation we can conclude that good behaviour in the past and the use of ratings from other participants is a high-quality metric in a social network.

A trust-based system built over a well-known social network brings a great opportunity to participate for all interested users as well as an opportunity to identify high-quality users whom may become in the leaders for tomorrow.

References

1. Hagen, M.: Elektronische Demokratie: Comput-ernetzwerke und politische Theorie in den USA, Lit-Verlagsgsellschaft, Hamburg (1997).
2. Zhang. W., Liu, L., Zhu, Y.: A Computational Trust Model for C2C Auctions. Proceedings of ICSSSM '05. (2005).
3. Cruz, C. C. P., Gouvêa, M. T. A., Motta, C. L. R., Santoro, F. M.: Towards Reputation Systems Applied to Communities of Practice. IEEE. 74-79 (2007).
4. Fernández, A., Hardings J.: A Protocol to Increase the Security in a MANET Based on Trust and Reputation. To be presented at Tenth International Symposium on Communication Theory and Applications (ISCTA ’09), 13th - 17th July, 2009, Ambleside, UK.
5. Zhou, R., Hwang, K.: Powertrust: A Robust and Scalable Reputation System for Trusted Peer-to-Peer Computing. IEEE Trans. Parallel Distrib. Syst. 460--473 (2007).
6. Josang, A., Haller, J.: Dirichlet Reputation Systems. The Second International Conference on Availability, Reliability and Security. 112--119 (2007).
7. Schlosser, A., Voss, M.: Comparing and Evaluating Metrics for Reputation Systems by Simulation. Proceedings of the IEEE Workshop on Reputation in Agent Societies.(2004).
8. Ren, Y., Boukerche, A.: An Efficient Trust-Based Reputation Protocol for Wireless and Mobile Ad Hoc Networks: Proof and Correctness. Proceedings of the IEEE GLOBECOM, (2008).

An Approach towards the Exploration of User Preference Adaptation for Cross Device, Cross Context Video Content Recommenders in Web 2.0 Environments

Kevin Mercer

Ergonomics Safety Research Institute, Loughborough University,
Leicestershire, LE11 3TU, United Kingdom
K.C.Mercer@lboro.ac.uk

Abstract. Video content recommenders are becoming wide spread in the age of ubiquitous access to the internet and web 2.0. But how does the context of video consumption effect content preference? This paper argues for a greater understanding of the impact of viewing context upon preference adaptation in the new age of multi platform, mobile video entertainment. This paper advocates an approach which investigates preference adaptation from the perspective of users self moderating content choices in response to perceptions of the current viewing context. The author suggests ethnographic study of naturalistic content consumption behaviours as a possible methodology to uncover insight into this area, which could inform design requirements for future video recommenders operating in cross context environments.

Key words: Video, Recommenders, User, Experience, Context.

1 Introduction

Video content recommenders have been with us for some time. Popularised by TiVo[1] on the set top box and MovieLens[2]and Netflix[3] on the internet we have seen video content recommender engines, (as well as peer recommendations) propagate onto many of the worlds most popular web 2.0 video, movie and TV web sites, see [1], [2], [3]. This has offered benefits to both commercial content providers as well as end users by enabling the promotion, discovery and enjoyment of long tail [4], video content.

Web 2.0 applications are at the forefront of an increasing trend towards video content aggregation and personalised recommendations. The technology has established itself at a time when mainstream ubiquitous access to high speed

[1] TiVo:http://www.Tivo.com
[2] MovieLens movie recommendations:http://www.movielens.org/
[3] Netflix:http://www.netflix.com/

internet has exploded. This has fueled an entertainment revolution with access to video services and long tail content over a multiplicity of mobile devices in addition to more traditional content delivery routes such as broadcast television. The next step in the enhancement of these systems (which is well underway) is the joining up of services across devices and networks in order to enable a consistent brand message and user experience. There are many examples of evidence for this trend in the marketplace with content providers and service operators alike offering singular branded video content delivery propositions across all of their television, internet and mobile services, [5], [6].

2 The Challenge

A user utilising a video content service across a range of devices and viewing contexts may have expectations regarding the types of content they wish to consume within each of those situations. We can imagine a member of a family when home alone, taking the opportunity to watch one piece of preferred content over another purely because they know it is of interest only to themselves. They do so under the rational that the other content whilst of equal interest is also enjoyed by the family as a whole. Therefore further opportunities to watch that item are more likely to present themselves at some other time with the family group. Another example could be a user choosing not to watch a movie to their preference on a train commute. The reasons being that the time available, mobile device screen size, and context of sitting on a train surrounded by strangers would (from their perspective) spoil the experience in contrast to watching the same content at home on the settee in front of their wide screen television.

As operators and broadcasters look to unify video content services across devices and environments, several fundamental theoretical questions are raised in relation to the contexts of video consumption which impact upon content selection decisions and therefore the role of recommenders. In order to provide a good experience to the user, a video recommender acting as part of such a service would need to take consideration of the nuances of context. This raises a number of issues surrounding not only if recommender outputs can be filtered to provide the best utility to a user within a given viewing context, but also if the construction of a single user model is valid for a system collecting information from many different viewing situations.

We must therefore ask, do the contextual factors which surround use in different socio-technical environments influence the video content selections users make at any given time? If this is the case, how can video content selections collected from within a specific context be usefully applied within a user model with the intention of providing recommendations across a landscape of ever changing contexts of use? Finally, even with an efficient recommender system in place how can recommendation selection and presentation be optimised to cope with the conditions imposed on a viewing experience by those same contextual factors? These questions need to be answered to ensure the success of future video content recommenders operating across devices within ubiquitous mobile environments.

3 Unraveling Perceived Context

Addressing research problems in this area requires a consideration of what context actually is. This short overview is not the place to delve deeply into the literature in relation to definitions of context, however many researchers agree that there still remains considerable confusion surrounding the notion of what context is [7], and many competing view points [8], [9], [10]. In terms of content recommenders the author advocates an approach which supports the following two viewpoints. Firstly Winograd [11], argues context is only that information which is useful to convey and act upon. Therefore developing a system to understand more than is needed in order to support the user activities is a source of wasted time, money and added complexity. Secondly is the argument of Bellotti and Edwards [12], that systems should not seek to act on behalf of the user, but should instead support a users actions and defer to them in efficient and non-obtrusive manners. This second point is pertinent to content recommenders if we approach viewing preference selections within given situations as self moderated action. The author defines this as selections made by a user in response to their own subjective perceptions of the current context and related predictions for the viewing experience to follow. The author would argue that these factors lead users to moderate absolute content preferences when choosing things to watch. This is a subtlety different approach to considering content selections purely in terms of video content preferences made within a specific context. The important factors for a recommendation system to focus upon now become the users own perception of the viewing context rather than any notion of context as a set of technical, geographical or temporal constraints.

This as a useful way in which to consider context in future video recommenders as historic restrictions on access to content due to the constraints of broadcast schedules and device connectivity are being rapidly eroded through technological and commercial advances. This approach addresses a world where we could view any video content anywhere. Choices are made in response to users own perceptions of the viewing context and predictions for the experience to follow, which in turn are based on past viewing experiences in others contexts perceived as similar.

Approaching the problem of context from the perspective of user perceptions has many precedents in the literature. As example the concepts of situatedness [13], and re-place-ing space [14], demonstrate that higher level notions of perceived context can provide a general approach to the identification of relevant aspects of a situation through which video consumption experiences may be characterised. A focus for contextual investigations following this research approach should therefore be to attempt to identify differences in viewing situations by the same inter-contextual cues [15], perceived by the users when they build their own mental models of the current context.

The hypothesis under consideration is that inter-contextual cues manifest within a consumption situation are used by the user in the formulation of their own perceptions of context, upon which they then base decisions to mediate their own content selection and consumption behaviours.

4 Next Steps and About this Research

The research direction advocated in this paper relates to user research activities currently underway aimed at identifying those aspects of context which influence video consumption behaviours and content selection. An ethnographic study is currently being conducted which is following a range of individual users through a two week period. An array of qualitative and quantitative data collection methods are being employed in an attempt to capture a picture of each users video content selection behaviours across the range of sources they consume video content from. In parallel the study will describe the setting (physical, social and technical) within which they consume. The goal is to attempt to identify self moderated patterns of content preference adaption and the important factors within each viewing situation which may signify the inter-contextual cues (upon perception of which) the user has responded by adapting their content choices.

This study is being carried out as part of a PhD project investigating the wider issue of user experience optimisation for future video content recommenders. The overall goal of the research is to investigate the possibility of a framework for a video content personalisation and presentation system which can operate across devices and consumption contexts whilst providing the best possible experiences for users.

References

1. Google Video, http://www.video.google.com
2. Veoh, http://www.veoh.com
3. Hollywood Video, http://www.hollywoodvideo.com
4. Anderson, C.: The Long Tail. Wired Magazine. 10, pp. 171–177. (2004)
5. British Sky Broadcasting, http://packages.sky.com/
6. BBC iPlayer, http://iplayerhelp.external.bbc.co.uk/help/about_iplayer/
7. Dourish, P.: What we talk about when we talk about context. Personal and Ubiquitous Computing. 8(1), pp. 19–30 (2004)
8. Jameson, A.: Modeling both the context and the user. Personal and Ubiquitous Computing. 5(1), pp. 29–33 (2001)
9. Chen, G., Kotz, D.: A survey of Context-Aware Mobile Computing Research. Dartmouth College, Hanover, New Hampshire. (2000)
10. Greenberg, S.: Context as a Dynamic Construct. Human-Computer Interaction. 16(2, 3, and 4), pp. 257-268 (2001)
11. Winograd, T.: Architectures for Context. Human-Computer Interaction. 16(2, 3, and 4), pp. 401-419 (2001)
12. Bellotti, V., Edwards, K.: Intelligibility and accountability. Human-Computer Interaction. 16(2, 3, and 4), pp. 193-212 (2001)
13. Rohlfing, K.J., Rehm, M., Goecke, K.U.: Situatedness: The Interplay between Context and Situation. Journal of Cognition and Culture. 3(2), pp. 132-156 (2003)
14. Harrison, S., Dourish, P.: Re-place-ing space: The Roles of Place and Space in Collaborative Systems. In: Proceedings of the Conference on Computer Supported Cooperative Work (CSCS 96), ACM Press, pp. 67-76 (1996)
15. Clancey, W.J.: Situated action: A neuropsychological interpretation response to Vera and Simon. Cognitive Science, 17(1), pp. 18-116 (1993)

MyHealthEducator: Personalization in the Age of Health 2.0

Luis Fernandez-Luque

Northern Research Institute, Tromso, Norway,
luis.luque@norut.no

Abstract. Most Europeans use the Internet for searching health information [1] and many of them use the Web 2.0 to access health information and services, share knowledge and socialize. There is an emerging trend towards the developing of personalized Health 2.0 applications which could dramatically change how the health consumers use the Web. This paper provides an overview of personalization in the Health 2.0 domain and it presents the ongoing project MyHealthEducator, which is an early example of personalization in the Age of Health 2.0. MyHealthEducator aims to study the feasibility of using Recommender Technologies for delivering personalized and adaptive recommendations of web health information based on the user's Personal Health Records and content from their community (e.g. user's comments).
Key words: eHealth, Personalization, Health 2.0, Health Education

1 Introduction

Personalization is not new in eHealth, especially in health education[2, 3]. It has been traditionally based on explicit feedback (e.g. questionnaires) for delivering tailored educational resources aiming at modifying a health behavior (e.g. stop smoking). In general, these stand-alone systems are static and designed for a specific disease, taking into account a closed set of parameters and resources controlled by healthcare professionals. This approach is not aimed at the current context of the Web 2.0, where many different types of health resources are appearing. For example, health consumers are creating content (e.g. blogs, v-logs, comments) and socializing through Social Networks (e.g. Facebook, Tudiabetes.com). They are also managing their health records by using web-based Personal Health Records (PHRs) such as Google Health.

The Web 2.0 provides many opportunities for personalized health applications, especially due to the increased availability of information about the users. For example, approximately half of teenagers' profiles in MySpace contain private health information (e.g. drug abuse, sexuality, etc.) [4]. This type of information is being used in the project Riskbot [5] for delivering personalized health promotion messages. Tags [6] and ratings [7] have been also used in personalized health education. In addition, there are already personalized applications based on the data available in Google's and Microsoft's PHRs. Bourgeois et al. [8] used Indivo PHR [9] for delivering tailored messages about influenza vaccination.

2 MyHealthEducator

MyHealthEducator approach consists of a service for recommending personalized health information adapted to the changing needs of the patients and not designed for a specific disease. Its main characteristic is the adaptability to the changes, both in the educational resources and in the user's data. We are aiming to achieve this adaptability with Semantic Modeling techniques to create dynamic models of the users and educational resources. The knowledge about the health user's status, preferences, and demographic information will be modeled as the changing user's context (e.g. diagnosed diseases) and gathered mainly from their PHRs. MyHealthEducator, figure 1, comprises 3 main components: 1) the User-models Repository which contains the information about the users 2) the Health-Repository with the metadata about the educational resources and 3) the Recommender Engine. The system is integrated with external components, such as the PHRs, repositories of health educational resources and the user interface. The user interface will vary depending on the platform where it is integrated. Currently, it is being designed to be integrated as a web-based gadget in our telemedicine platform MyHealthService [10].

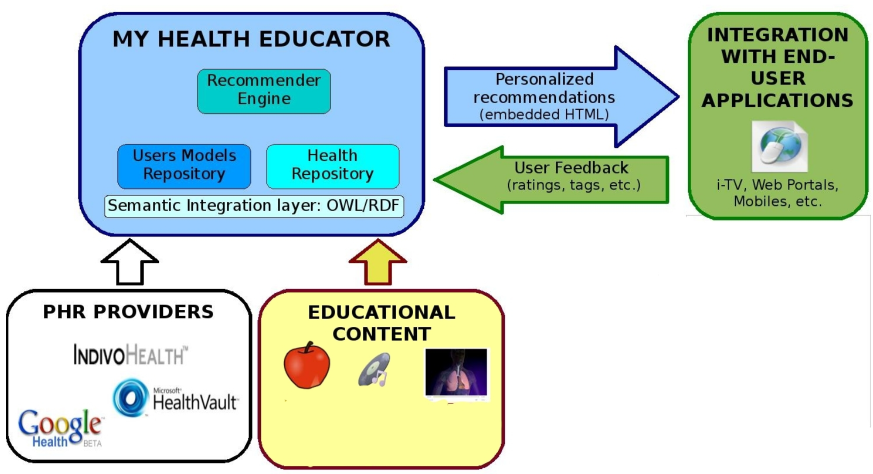


Fig. 1. Overview of MyHealthEducator

2.1 User-models Repository and Health Repository

The **User-models Repository** contains the information about the users. The health information will be gathered from the PHRs. After the users grant access to their PHRs, the system can access the user's data using the PHR's secure APIs. The health information will be modeled as context, which could vary and be different between the users. The non-health related information about the users is also modeled and stored in a Personal Record containing information such as the user's preferences, which will be gathered using implicit feedback (e.g. user's interaction with the system) and explicit feedback (e.g. favorited content provider). The models will be based on Semantic Technologies, such as Semantic Networks or Concept Profiles. Instead of extracting list keywords to build the user models the system captures linked concepts and terms, decreasing the polysemy problem. We are looking into the usage of the Unified Medical Language System (UMLS) Semantic Networks, which are widely used in the health domain and have been ported to OWL.

The **Health Repository** contains the metadata about the web-based educational resources. Similar to the User-models the available information about the resources (e.g. descriptions, comments) will be analyzed to extract relevant keywords and concepts in order to build a semantic model of the resources. The information created by the community of users (e.g. ratings, comments) will be also used to enrich the resource's model. One of the main challenges to address will be the diversity of the users' vocabulary and the use of acronyms.

2.2 Recommender Engine

The recommender engine will be a hybrid Recommender System based on: 1) the analysis of the semantic structure of the models about the users and the educational resources, 2) collaborative techniques. A pre-filtered list of educational resources is generated by analyzing the semantic similarity between the users and resources models. Finally, the list is sorted using collaborative techniques.

2.3 Status and future work

MyHealthEducator is currently under development based on our previous studies about the Patient Generated Content, such as educational resources [11] and comments [12]. The first prototype, which is expected by the end of 2009, will be a recommender system of health videos from YouTube based on the analysis of the User Generated Content and the PHRs. The evaluation of this prototype will be focused on the evaluation of different recommendation algorithms based on the analysis of data collected from the system usage and users' feedback (e.g. surveys).

3 Conclusions

The increased availability of structured and un-structured data about health consumers and content has opened a new conduit for research opportunities towards the development of personalized Health 2.0 applications, where PHRs are becoming platforms with ecosystems of personalized health applications. The impact of these applications can ultimately lead to a paradigm shift of patient-centered healthcare systems. Many challenges are also appearing; including new ethical dilemmas related to web-mining sensitive information or technical questions regarding to the interoperability and integration. Some of these challenges are being addressed in the ongoing project MyHealthEducator. This project will increase the knowledge about the usage of Web Technologies for health personalization.

4 Acknowledgements

I would like to thank MyHealthService team, especially Randi Karlsen and Lars K. Vognild. This project belongs to the Tromso Telemedicicine Laboratory (co-funded by the Research Council of Norway).

References

1. Andreassen, H.K., Maria, Chronaki, C.E., Dumitru, R.C., Pudule, I., Santana, S., Voss, H., Wynn, R.: European citizens' use of e-health services: A study of seven countries. BMC Public Health **7** (April 2007) 53+
2. Cawsey, A., Grasso, F., Paris, C.: Adaptive information for consumers of healthcare. (2007) 465–484
3. Lustria, M., Cortese, J., Noar, S., Glueckauf, R.: Computer-tailored health interventions delivered over the web: Review and analysis of key components. Patient Education and Counseling **74**(2) (February 2009) 156–173
4. Moreno, M.A., Parks, M.R., Zimmerman, F.J., Brito, T.E., Christakis, D.A.: Display of health risk behaviors on myspace by adolescents: Prevalence and associations. Archives of pediatrics & adolescent medicine **163**(1) (January 2009) 27–34
5. Jason Bonander: Personally tailored health information: a health 2.0 approach. In: Proceedings of Medicine 2.0., Medicine 2.0. 2008, Toronto, Canada (September 2008)
6. Khan, S.A., Kukafka, R., Cohall, A.: A tag based recommendation engine to suggest information resources in an online community for health promotion. AMIA .Annual Symposium proceedings (2008)
7. Witteman H, Chignell M, K.M.: A recommender system for prostate cancer websites. In: AMIA Annu Symp Proc. (6 November 2008)
8. Bourgeois, T.F., Simons, W.W., Olson, K., Brownstein, S.J., Mandl, D.K.: Evaluation of influenza prevention in the workplace using a personally controlled health record: Randomized controlled trial. J Med Internet Res **10**(1) (Mar 2008) e5
9. Mandl, K.D., Simons, W.W., Crawford, W.C.R., Abbett, J.M.: Indivo: a personally controlled health record for health information exchange and communication. BMC Medical Informatics and Decision Making **7** (September 2007) 25+
10. Burkow, T.M., Vognild, L.K., Krogstad, T., Borch, N., Ostengen, G., Bratvold, A., Risberg, M.J.: An easy to use and affordable home-based personal eHealth system for chronic disease management based on free open source software. Stud Health Technol Inform **136** (2008) 83–8
11. Fernandez-Luque, L.: Study of the epatient as a generator of content. In: Proceedings of Medicine 2.0., Medicine 2.0. 2008, Toronto, Canada (September 2008)
12. Fernandez-Luque, L.: An analysis of personal medical information disclosed in youtube videos created by patients with multiple sclerosis. In: MIE 2009 (accepted), Sarajevo, Bosnia (September 2009)

New Tagging Paradigms for Content Recommendation in Web 2.0 Portals

Andreas Nauerz[1], Fedor Bakalov[2], Martin Welsch[1], and Birgitta König-Ries[2]

[1]IBM Research and Development, 71032 Böblingen, Germany,
{andreas.nauerz | martin.welsch}@de.ibm.com
[2]Friedrich-Schiller University of Jena, 07743 Jena, Germany,
{fedor.bakalov | birgitta.koenig-ries}@uni-jena.de

Abstract. With this paper we tie in with what we presented during last year's workshop [5] where we illustrated how to analyze users' tagging and rating behavior to construct user- and context models that can be used to perform adaptations and to issue recommendations in order to create more user-tailored Web Portals. This time we want to present more sophisticated tagging paradigms and their influence on users collaboration behavior and the construction of user- and context-models. The concepts presented are currently been prototypically implemented within IBM's WebSphere Portal and can be presented in a live demo at the workshop.

1 Introduction

In recent years Enterprise Information Portals have gained importance in many companies. As a single point of access they integrate various applications and processes into one homogeneous user interface. Today, typical Portals are comprised of a huge amount of content. They are no longer exclusively maintained by an IT department, instead, Web 2.0 techniques are used increasingly, allowing user generated content to be added. These systems grow quickly and in a more uncoordinated way as different users possess different knowledge and expertise and obey to different mental models. The continuous growth makes access to really relevant information difficult. Users need to find task- and role-specific information quickly, but face information overload and often feel *lost in hyperspace*. Thus, users often miss out on resources that are potentially relevant to their tasks, simply because they never come across them. On the one hand, users obtain too much information that is not relevant to their current task, on the other hand, it becomes cumbersome to find the right information and they do not obtain all the information that would be relevant.

The recent popularity of collaboration techniques on the Internet, particularly tagging and rating, provides new means for both semantically describing Portal content as well as for reasoning about users' interests, preferences and contexts. It can add valuable meta information and even lightweight semantics to web resources.

In our previous work [5] we proposed a framework which allowed arbitrary annotators, e.g. human users or analysis components (for automated tagging), to annotate any of these resources. Analysis of the tagging behavior allowed us to model interests and preferences of users as well as semantic relations between resources, and thus to perform reasonable recommendations and adaptations.

In this paper we will present tagging paradigms like alien tagging, reputation-based tagging, quantitative tagging, anti tagging, tag expiry, contextual tagging, and describe how these can be used to refine our models and to perform even more valuable adaptations or to issue more valuable recommendations.

2 Related Work

Using collaborative metrics to get to know what is of relevance to users or entire user groups has been done before. Especially collaborative ranking, i.e. ranking which takes into consideration entire communitys' interests, has recently become more important. Access patterns are used to assess the importance of single web pages [1]. Improved versions of the original PageRank [6] and HITS [3] algorithms have been developed (cp. FolkRank [2], CollaborativeRank [4]).

Other work focuses on the personalized recommendation of content based on its relatedness to certain tag terms. [7] proposes a modified version of the HITS algorithm to determine experts and high-quality documents related to a given tag.

3 Concepts

3.1 Alien tagging

As said before Web 2.0 communities can be rather heterogeneous. The expertise of users contributing (and consuming) content can vary a lot. What might be obvious for one user might be completely unknown to others. *Alien tagging* allows more experienced users to tag content for less experienced ones. In our prototypical implementation tag widgets allow power users to apply tags to resources on behalf of other users (or even user groups). Next time one of the users for which alien tags have been applied logs-in, he or she is notified about the availability of these and can inspect the underlying resources. The same way we used "normal" tags in our previous work [5] to refine user models that describe users interests and preferences we can use these alien tags, too. In real environments *alien tagging* could be used e.g. by managers pretagging content for their new hires, by team- or technical leads to point their team members to relevant content which they otherwise might have missed. Thus *alien tagging* opens another opportunity to prevent users from missing out content by issuing recommendations provided by "alien" users.

3.2 Reputation-based tagging

In our previous solutions we always assumed that the weight (i.e. the importance) of tags only depends on the frequency of their occurrence. I.e. a tag applied more often with respect to a certain scope was regarded of higher importance than a tag applied less often. In our new prototype we additionally assume that the weight of a tag can depend on the reputation (or expertise) of a user. I.e. that tags applied by more experienced users have higher weights, and thus higher influence on what content the community is presented (or recommended) with, than tags from less experienced users. This way we can point users to more relevant content as we assume experts to know better what the community should focus on. E.g., in development team we assume the tagging behavior of the team- or technical lead of higher importance. With *reputation-based tagging* we also ensure that ”incorrect or less suited” tags perceive lower weights (influence). E.g., a newbie might apply a more ”incorrect/less suited” tag as he just misunderstands (due to his insufficient knowledge) what he is looking at. The way we determine users' expertise has already been described in [5].

3.3 Quantitative tagging

Previously we also assumed that tags can only have ”positive character”. I.e. that we assumed that a resource can be tagged with a term to describe that the resource has something to do with this term, but also assumed that a resource cannot be tagged with a term to describe that the resource has nothing to do with it. In addition to that aspect we did not provide means for single users to express that a certain tag is of less relevancy for them. *Quantitative tagging* provides a solution to both problems: in our prototypical implementation a plus- and a minus sign is presented besides each tag being displayed. In addition, when applying a tag, a not-sign is presented. Clicking the not-sign when applying a tag allows users to express that a resource has nothing to do with the term applied, a helpful feature for more fine-granular categorization of resources: e.g., users could tag some resources with the term *Web 2.0* and a few of them with ”not” *scientific*. This helps users to quickly find all Web 2.0 related resources and to quickly distinguish between the scientific and non scientific ones among them. Clicking the plus- and minus-signs when working with tags allows single users to express that they are less interested in a tag (or a certain tag associated to a certain resource) or can additionally express that a tag is of less relevancy for the entire community. Thus, these mechanisms allow for further refinement of our user models.

Anti tagging *Anti tagging* describes an enhancement to *quantitative tagging* (cp. 3.3). Here we automatically increase or decrease tags' relevancy for the entire community by analyzing tags semantics (cp. [5]). One option we have evaluated is to take into consideration antonyms. E.g., when a resource is tagged with ”good” and ”bad” we regard it as not tagged at all with either of these two terms as they annihilate each other. Antonyms can e.g. be found using the antonym thesaurus [1]. As *anti tagging* is not trivial to be realized as most examples are much more complicated and less obvious than the one just provided we have not yet incorporated it in our prototype.

3.4 Tag expiry

In our previous work we also assumed that tags can be applied once and stay alive until they are manually deleted again. This let to tag-space littering as most users never deleted tags anymore even if they became obsolete. The fact that tags do not remain valid forever occurs in Portals that provide dynamic content very often. This resulted in having a lot of tags assigned to resources that did not describe the resource adequately nor express the resources relevancy to the community appropriately anymore. In our prototype *tag expiry* allows users to specify a chronological validity for tags when assigning them to a resource. Taggers can give tags a start date, an end date or a time frame in between they live. We also allow tags that are assigned a "lifetime" to become more (or less) important as time passes by. E.g. if there is a page in the Portal system providing information about the Olympic Games 2012, this page might become more and more interesting to users as we get nearer to the year 2012 and less interesting after 2012. Thus users can specify that the tag should not be available before 2011, vanish after 2013 and become more important from 2011 till 2012 and less important from 2012 till 2013. Thus, *tag expiry* is yet another mechanism to help the community to focus on what is currently really relevant. Moreover, *tag expiry* allows us to neglect "invalid" tags from being considered when doing content adaptation or recommendation.

3.5 Tagging tags and meta-tagging

Previously we have also worked on solutions to solve major problems of tagging systems: most of these problems discussed dealt with synonyms (multiple tags having the same meaning) and polysemies (a single tag having different meanings). Current tag engines often try to overcome these issues by applying stemming and normalization algorithms which most often only solve problems resulting from morphological variations. Semantical variations can most often not be detected to be a synonym e.g. In our latest prototype we allow the community to resolve the resulting tag-space littering. In our tag-clouds we allow users to drag and drop tags on each other to consolidate them. In addition to that we allow users to create meta-tags (or meta-tag bags as we call them) under which other tags can be organized. Users could create private meta-tag bags only they can see or community meta-tag bags all users part of the community can see. That way users could e.g. create a meta-tag bag "sports" drag all sports related tags into that bag; users could also create a meta-tag bag "favorite-stuff" and just drag what he/she likes most into it.

[1] http://www.synonym.com/synonyms/

3.6 Other concepts

We are also allowing for tag sharing among subcommunities. Most current tagging systems allow to either create public or private tags but do not allow for a granularity in between. Our prototype allows to share tags with a dedicated set of other users. We also allow for contextual tagging where we can associate tags a certain context (for our context modeling approaches refer to [5]) to prevent irrelevant tags (irrelevant in a certain context) to appear. The latter helps focusing on currently relevant content again.

4 Conclusion and Future Work

In this paper we have presented tagging paradigms which we are using to refine our user- and context modeling approaches presented in our previous work [5] in order to perform content adaptation and recommendation. The concepts described have already been prototypically implemented and can be presented at the workshop. We have not yet performed in-depth evaluation on these early ideas described in this short paper but are looking forward to discuss them and receive initial feedback. Of course, especially the usefulness of each single concept has still to be evaluated.

For the future we plan to merge our Web 2.0 collaborative tagging approaches with Semantic Web ideas heading towards the Web 3.0.

References

1. J. Caverlee, Ling Liu, and W.B. Rouse. Link-based ranking of the web with source-centric collaboration. *colcom*, 0:11, 2006.
2. Andreas Hotho, Robert Jäschke, Christoph Schmitz, and Gerd Stumme. Folkrank : A ranking algorithm for folksonomies. In *LWA*, pages 111–114, 2006.
3. Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. In *SODA*, pages 668–677, 1998.
4. Amir Michail. CollaborativeRank: Motivating People to Give Helpful and Timely Ranking Suggestions. http://collabrank.web.cse.unsw.edu.au/collabrank.pdf, April 2005. (work in progress).
5. Andreas Nauerz, Stefan Pietschmann, and Rene Pietzsch. Social recommendation and adaptation in web portals. In *Proceedings of the International Workshop on Adaptation for the Social Web*, Hannover, Germany, 2008.
6. Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project, 1998.
7. Harris Wu, Mohammad Zubair, and Kurt Maly. Harvesting social knowledge from folksonomies. In *Proc. of the 17th Conf. on Hypertext and hypermedia*, pages 111–114, New York, NY, USA, 2006. ACM Press.

Personal Navigation in Semantic Wikis

Diego Torres[1] Alicia Díaz[1] Hala Skaf-Molli[2] and Pascal Molli[2]

[1] LIFIA, Facultad de Informática,
Universidad Nacional de La Plata, Argentina
{diego.torres, alicia.diaz}@lifia.info.unlp.edu.ar
[2] LORIA – INRIA Nancy-Grand Est
Nancy Université, France
{skaf, molli}@loria.fr

Abstract. In this paper, we propose a personal navigation approach to Semantic Wikis. In semantic wikis, wikis pages are annotated with semantic data to facilitate research and navigation. The navigation is collaborative designed and shared by every user. However, individuals involved in a collaborative knowledge building activity need to customize the navigation according to her personal needs. In order to overcome this, we extend semantic wikis with personal annotations facilities to support personal navigation. This approach differs from other adaptive navigation approaches, because of the personalization is carried out by the user herself. We have implemented and validated these ideas on the top of a P2P semantic wiki.

1 Introduction

One of the most characteristic features of Social Web or Web 2.0 systems is users change their participation style. In these systems, users become contributors of contents, being more proactive to edit,they produce and share resources on the Web. In this new approach, computer-supporter collaborative systems (CSCW) have taken a notorious position, specially, those that support a collaborative knowledge building activity [1]. Semantic Wikis [2–4], which are one of the most successful semantic web applications, are well positioned, since they enable users to develop a shared knowledge repository in cooperation.

Semantic wikis differ from traditional wikis. They enable users to add semantic annotations to the wiki pages. These semantic annotations allow us to structure the content of the Wiki and to promote a more suitable navigation. Semantic annotations can be attached according an existing ontology (like in IkeWiki [2]) or by categorizing and adding semantic relationships like in Semantic MediaWiki [3] or Swooki [4]. In the most popular semantic wikis [2–4] the navigation is determined by categories pages and semantic relationships. The resulting navigation is shared among all the users.

However, the individuals involved in a collaborative knowledge building activity, need supports for personal activities [5], even, when this activity is navigation. Therefore, individuals need to be able to adapt wiki navigation to personal needs.

At the moment, personal semantic wikis like SemperWiki [6] supports personal navigation; however, the wiki is dedicated only to one user. On the other hand, in the most of semantic wikis [2–4] the navigational hyperspace is the result of a social process and is shared by every user. Currently, there is no available semantic wiki that combines both personal and shared navigation.

In order to overcome this problem, we propose to extend semantic wikis with personal annotations to achieveing personal navigation. We believe that the fact of adding personal navigation in a semantic wiki enables to provide customized navigation according to user needs, to achieve concept-based navigation [7], to combine personal and shared navigation and consequently, to enrich and augment the shared navigation hyperspace. This approach differs from other adaptive navigation approaches [7], because of the personalization is carried out by the user herself.

In this paper, we present a new approach to support personal navigation in semantic wikis. The Section 2 introduces navigation features in semantic wikis. The Section 3 explains how personal semantic annotations help user to organize personal navigation. The Section 4 presents implementations. The last Section concludes the paper and introduces some future works.

2 Navigation in Semantic Wikis

In most of the existing semantic wikis [2–4], semantic annotations are used to categorize a wiki page and to define semantic relationships. These two kinds of semantic annotations propose two different alternatives to express navigation:

- Semantic relationships can represent a typed link between wiki pages. These links can be embedded in the wiki content. Therefore, navigation through a semantic relationship is clearly richer than those in traditional wikis.
- The other alternative is the navigation by means of categories. Each category has its own wiki page. Category pages are useful to browse semantic wikis, because each category page is an access point to wiki pages that were categorized by the underlined concept of the current category. Category pages implement what Brusilovsky calls concept-based navigation [7].

In the most semantic wikis, the navigation is shared among the users. The navigation space is the product of a collaborative activity, where every user is able to categorize wiki pages or to add semantic relationships. However, they do not provide any private and personal navigation. In the next section, we introduce how we have extended semantic wikis under the concept of personal semantic annotation and we explain how these annotations help users to organize personal navigation.

3 Personal Semantic Annotations for Personal Navigation

We have developed P-Swooki [5] as an extension of Swooki, a P2P semantic wiki [4]. P-Swooki supports both personal and shared navigation. Both kinds of navigation are expressed in a differentiated way. The shared navigation is product of the collaborative knowledge building as in any semantic wiki. The personal navigation is defined by means of personal semantic annotations.

In P-Swooki, for shared navigation, we follow the same approach as in SMW where *shared semantic annotations* are embedded in the wiki text by using a suitable syntax. In this approach, the shared semantic annotations are unique and accessible to everyone. For personal navigation, we propose *Personal Semantic Annotations*.

Personal semantic annotations look like *tags* as it is shown in the figure 1, however they are semantically richer. They can be a *category* or an *individual*. *Categories* define a family of elements, whereas *Individuals* denote elements that fall at least in one category. *Personal semantic annotations* are associated to the wiki page and they are only accessed by the owner user. Every wiki page could be tagged with several personal annotations

Adding *Personal semantic annotations* improves semantic wiki navigation, because:

- Personal semantic annotations enable customized navigation. As personal annotations are handled in a private space, they are only accessible by the owner user. Therefore, users are able to adapt the navigation according to their needs.
- Personal semantic annotations also improve concept-based navigation. The users can define new categories by means of personal category annotations and therefore, they are able to described personal concept-based navigation.
- The personal navigational hyperspace complements the shared one. Consequently, the shared navigational hyperspace is enriched and augmented by the personal navigational hyperspace.

4 Implementation

We implemented P-Swooki as an extension to Swooki. We choose a P2P approach because it is easy to manage the propagation of both kinds of annotations.

In P-Swooki, there are two repositories: a shared and a personal one. Every user works in one peer and has her own copy of shared data. The shared repository is identical for all users thanks to the synchronization algorithm [4]. The personal repository is not replicated to the rest of the peers and users keep personal annotations private.

The figure 1 shows the P-Swooki GUI. On the right, there is a widget which enables to add personal semantic annotations. Such widget also enables us to browse annotations. Additionally, P-Swooki provides a kind of category page as it can be shown on figure 2.

We have conducted some usage studies of P-Swooki. The results of this evaluation showed us the needed of personal navigation features as detailed in [5].

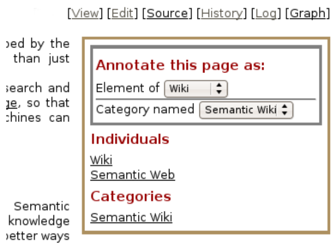


Fig. 1. Personal Annotation Box

Fig. 2. Personal Annotation Navigation Page

5 Conclusions and further work

In this paper, we have proposed an approach to extend semantic wikis navigational capabilities with personal ones. Personal navigation is carried out by the use of personal semantic annotations. By means of personal semantic annotations user can customize navigation according to their needs and to achieve personal concept-based navigation.

Currently, we are working on a transformation algorithm that allows transforming personal semantic navigation into shared one and vice-versa. This will allow enriching both personal and shared navigation.

References

1. Stahl, G., ed.: Group cognition: Computer support for building collaborative knowledge. Cambridge, MA: MIT Press (2006)
2. Schaffert, S.: IkeWiki: A Semantic Wiki for Collaborative Knowledge Management. 1st International Workshop on Semantic Technologies in Collaborative Applications (STICA06), Manchester, UK (2006)
3. Völkel, M., Krötzsch, M., Vrandecic, D., Haller, H., Studer, R.: Semantic wikipedia. Journal of Web Semantics **5**(4) (2007)
4. Rahhal, C., Skaf-Molli, H., Molli, P.: Swooki: A peer-to-peer semantic wiki. In: The 3rd Workshop: 'The Wiki Way of Semantics'-SemWiki, co-located with the 5th Annual European Semantic Web Conference (ESWC), Tenerife, Spain. (June 2008)
5. Torres, D., Skaf-Molli, H., Diaz, A., Molli, P.: Personal and shared knowledge building in p2p semantic wikis. Research Report 6783, INRIA (12 2008)
6. Oren, E., Breslin, J.G., Decker, S.: How semantics make better wikis. In: Proceedings of the International World-Wide Web Conference, New York, NY, USA, ACM Press (2006) 1071–1072
7. Brusilovsky, P.: Adaptive navigation support. In Brusilovsky, P., Kobsa, A., Nejdl, W., eds.: The Adaptive Web. Volume 4321 of Lecture Notes in Computer Science. Springer-Verlag, Berlin, Germany (May 2007) 263–290

AP WEB 2.0
http://ailab.dimi.uniud.it/en/events/2009/ap-web20/